
A FRAMEWORK FOR MONITORING
BACKUPS AND THEIR PROPERTIES
FOR A VAST NUMBER OF
HETEROGENEOUS SYSTEMS IN A
BUSINESS CLOUD COMPUTING
ENVIRONMENT

ANNE WENDT

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF COMPUTER AND INFORMATION SCIENCES

2015

SCHOOL OF COMPUTER AND MATHEMATICAL SCIENCES

EMBARGO NOTICE

This thesis contains confidential data and information that are of strategic importance to SAP. It may only be made available to examiners and authorised members of the board of examiners. Any whole or partial publication or duplication of this thesis is strictly prohibited. An inspection of this work by third parties requires the expressed written permission of the author and SAP. This thesis is subject to an embargo until 31st January 2017 (see Appendix B).

TABLE OF CONTENTS

Embargo Notice.....	II
List of Figures.....	VII
List of Tables.....	VIII
List of Acronyms.....	IX
Attestation of Authorship.....	XI
Acknowledgements.....	XII
Ethical Approval.....	XII
Abstract.....	13
Chapter 1 Introduction.....	14
1.1 Cloud Computing.....	14
1.2 Research Partner SAP.....	16
1.3 Research Objective.....	19
1.4 Thesis Structure.....	22
Chapter 2 Literature Review.....	24
2.1 Backups and Backup Technologies.....	25
2.1.1 From tapes to snapshots.....	25
2.1.2 Strategies for “good” backups.....	26
2.1.3 Backup and data storage as cloud-based service.....	27
2.2 Heterogeneity.....	28
2.2.1 Heterogeneity from a technical perspective.....	28
2.2.2 Heterogeneity from a business perspective.....	31
2.3 Providing Cloud Services.....	32
2.3.1 Virtualisation.....	32
2.3.2 Multitenancy.....	34
2.3.3 Monitoring cloud performance.....	35
2.3.4 Creating private clouds.....	35

2.4	Cloud Security	36
2.4.1	Industry standards and frameworks	36
2.4.2	Building secure systems	38
2.5	The Research Question	40
2.6	Chapter Summary	42
Chapter 3	Research Methodology	43
3.1	Theoretical Foundations.....	43
3.1.1	The Design Science framework	43
3.1.2	The software development lifecycle	46
3.1.3	Requirements engineering.....	47
3.2	A Pre-Study	50
3.2.1	Project description	50
3.2.2	Project findings.....	52
3.3	Design of the Research.....	54
3.4	Chapter Summary	57
Chapter 4	Findings	58
4.1	Environment Description	59
4.2	Interview Findings.....	60
4.2.1	Participant summary	60
4.2.2	Data analysis.....	64
4.2.3	Set of requirements	68
4.2.4	Interdependencies between requirements	75
4.2.5	Other findings.....	79
4.3	Chapter Summary	80
Chapter 5	Software Design	81
5.1	Constraints.....	81
5.2	Design Considerations	83
5.3	Software Overview	84

5.3.1	Database schema	84
5.3.2	Background data collection.....	87
5.3.3	User interface	91
5.4	Chapter Summary	96
Chapter 6	Discussion	97
6.1	Software Artefact.....	97
6.1.1	Compliance with user requirements	98
6.1.2	Assessment of the overall functionality	102
6.2	Research Validity	103
6.3	The Framework	104
6.3.1	Framework description.....	105
6.3.2	Analysis of the framework	108
6.4	Chapter Summary	110
Chapter 7	Conclusion.....	111
References	114
Appendices	120
Appendix A	PGR1 Form	120
Appendix B	Embargo Approval.....	126
Appendix C	Ethical Approval.....	127
Appendix D	Participant Information Sheet.....	128
Appendix E	Participant Consent Form	131
Appendix F	Invitation Letter	132
Appendix G	Interview Notes.....	133
Participant 1	133
Participant 2	134
Participant 3	135
Participant 4	136
Participant 5	137

Participant 6	138
Participant 7	138
Participant 8	139
Appendix H Mapping Interview Notes into Requirements	141
Participant 1	141
Participant 2	142
Participant 3	143
Participant 4	144
Participant 5	145
Participant 6	146
Participant 7	146
Participant 8	148
Appendix I Notes taken during the operation of the tool	150
Appendix J Source Code	151
Bkp_rep_daily_data_collector.pl	151
Monitoring.pm	159
Appendix K Fold-out List of Requirements.....	172

LIST OF FIGURES

All figures were created by the researcher, unless otherwise specified.

Figure 1-1: The three cloud service models	14
Figure 1-2: The four cloud deployment models	15
Figure 1-3: The five areas of SAP's cloud operations	21
Figure 2-1: Research areas related to the study.....	24
Figure 2-2: Architectures of federated databases.....	29
Figure 2-3: Virtualised infrastructure	33
Figure 2-4: Process to create a secure system	39
Figure 3-1: Design Science research guidelines: essential parts of an artefact	44
Figure 3-2: Design Science research methodology.....	45
Figure 3-3: The software development lifecycle	46
Figure 3-4: Six essential skills for requirements engineering	49
Figure 3-5: First screenshot of the tool developed in the pre-study.....	51
Figure 3-6: Current screenshot of the tool developed in the pre-study.....	53
Figure 3-7: Methodology framework	55
Figure 4-1: Requirements identification in the context of the methodology	58
Figure 4-2: Requirement connections in terms of functionality.....	76
Figure 4-3: Requirement connections in terms of software component interaction	78
Figure 5-1: Software design in the context of the methodology	81
Figure 5-2: Database schema.....	85
Figure 5-3: Flowchart of the data collection script	88
Figure 5-4: Flowchart of the error handling.....	90
Figure 5-5: Data selection screen of the software tool.....	92
Figure 5-6: Result display of the software tool	94
Figure 6-1: Demonstration and evaluation in the context of the methodology	97
Figure 6-2: Categorisation of requirements implementation	98
Figure 6-3: The proposed framework for the development of a backup monitoring tool	105

LIST OF TABLES

All tables were created by the researcher, unless otherwise specified.

Table 1-1: Categorisation of SAP's cloud products	18
Table 2-1: Categorisation of cloud standards.....	36
Table 4-1: Participant summary	62
Table 5-1: List of mapping system states into coloured dots	96

LIST OF ACRONYMS

Acronym	Meaning
AICPA	American Institute of Certified Public Accountants
BaaS	Backup as a Service
BOD	Business Object Document
ByD	Business byDesign
COD	Customer Relationship Management on Demand
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
FDBMS	Federated database management system
FIN	Financials on Demand
GMP	Global Management Portal
HANA	H igh Performance A nalytic A ppliance
HEC	HANA Enterprise Cloud
IaaS	Infrastructure as a Service
IAASB	International Auditing and Assurance Standards Board
IFAC	International Federation of Accountants
ISAE	International Standard on Assurance Engagements
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
JPaaS	Java Platform as a Service
NIST	National Institute for Standards and Technology
PaaS	Platform as a Service
PAY	Payroll on Demand
SaaS	Software as a Service
SAS	Statement on Auditing Standards

SCM	Supply Chain Management
SISM	SAP in-house system manager
SME	Small and medium enterprises
SOC	Service Organisation Controls
SOX	Sarbanes Oxley Act
SPC	Service Provider Cockpit
SSAE	Statement on Standards for Attestation Engagement
TEM	Travel and Expense Management on Demand
TIC	Technical Infrastructure Controller
VLAB	Verification lab
VM	Virtual Machine

ATTESTATION OF AUTHORSHIP

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly stated in the acknowledgements); nor does it contain material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Anne Wendt

ACKNOWLEDGEMENTS

I would like to use this opportunity to thank the people who have made this work possible.

First, I would like to express my sincerest gratitude to my primary supervisor, Krassie Petrova. Your guidance and support were invaluable for completing this research. Thank you for understanding my shortcomings and untangling my spaghetti ☺

Many thanks go also to my second supervisor, Professor Stephen MacDonell. Your comments on my work were most helpful and gave me some peace of mind that I was not too far astray.

I am deeply grateful to the German Academic Exchange Service for granting me a scholarship so I could pursue my Master's studies. Without their financial support I would have had no means of going to New Zealand, and even less be able to study here.

During my work for SAP I had the pleasure to meet many great people to whom I am truly thankful: Heinrich, Michael, and Paul, my guides, mentors, and friends. Angus and Dieter, who broke fresh grounds and made the impossible possible. Jaffer Ali, whose knowledge is just incredible. Christian and Falk, for their patience and help. And of course all the other colleagues who supported me in my time with SAP. You are awesome!

Special thanks go to my participants for freeing space in their busy schedules to support me in my research. I hope the outcome of my work meets your expectations.

Furthermore, I would like to thank Diana Kassabova for her excellent proofreading that was not only speedy but also taught me a better way of expressing myself.

Last but certainly not least, heartfelt thanks go to my parents, my sister, and my boyfriend. Thank you so much for believing in me and making this journey possible.

ETHICAL APPROVAL

Ethical approval for this research was obtained from the Auckland University of Technology Ethics Committee (AUTEK) on 24th February 2014, AUTEK application number 13/315. The approval can be found in Appendix C.

ABSTRACT

With the increasing popularity of cloud computing technologies, the expectations towards cloud providers increase not only in terms of functionality, cost, and service delivery speed, but also for the security of the customers' data that is stored remotely. Cloud providers can be certified for their compliance with established industry standards for data security; however, there is minimal research published on how to design internal processes to achieve this goal. The objective of this thesis is to address this gap by providing a framework that can be used by cloud providers to create a process for monitoring customer data backups. The area of backup and restore is an important part of data security, as it ensures data integrity and availability. In order to address the research objective, the Design Science methodology is combined with the software development lifecycle so that a specific software-based instantiation of the framework can be designed, implemented and evaluated.

The first outcome of the research is a set of 36 requirements for the software tool that are collected by interviewing business experts who work for SAP, a cloud provider and the partner of this research. It was found that the most important aspect of monitoring backups is to create a flexible input option for adding new system types. An intermediate outcome of the research is a method of discovering interconnections between requirements by using post-it notes that are placed on a whiteboard so that they can be positioned in relation to each other by drawing arrows and lines between them. The second major outcome is the software tool that is developed based on the findings from the requirements and is assessed for utility by a sample of users. The aspect of flexibility is addressed by basing the data collection on a search query that can be configured according to the users' specifications. By generalising the workings of the software tool, a framework is developed as the final outcome that is independent from the underlying technical base. It consists of four stages (system analysis, solution design and development, process operation, and communication) that should enable an implementing company to create a reliable backup monitoring process.

CHAPTER 1 INTRODUCTION

This chapter introduces the reader to the application area of this study, which is cloud computing, and to the company that served as a research partner, SAP. Furthermore, it states the rationale and research objectives and describes the structure of the thesis.

1.1 CLOUD COMPUTING

Cloud computing is a way of dynamically delivering computing power, storage and applications to customers and is based on their current demands (Hill, Hirsch, Lake, & Moshiri, 2013). The customers access a defined set of resources remotely via the Internet, rather than having them on their premises. According to the definition of cloud computing by the National Institute of Standards and Technology (NIST), there are three service models which define the scope of the service provided (Mell & Grance, 2011). Figure 1-1 shows the relationship between these models.

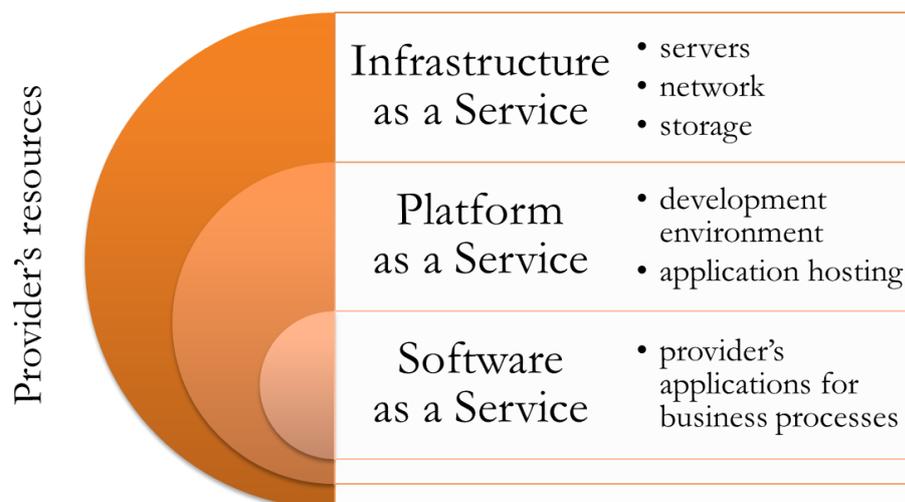


Figure 1-1: The three cloud service models¹

Infrastructure as a Service (IaaS) provides the greatest access to the provider's resources. Customers may run their own operating systems and applications, as if the virtual environment were an extension to their own hardware (Hill et al., 2013).

Platform as a Service (PaaS) restricts the customers to a certain development environment which is set up by the provider. It mainly functions as virtual hosting space for customers' own applications (Hill et al., 2013).

¹ Here and throughout the thesis all figures and tables included were created by the researcher.

Software as a Service (SaaS) gives customers the least freedom for configurations, but the greatest convenience. All applications and their maintenance, as well as the whole underlying hardware and infrastructure, are managed by the provider. Customers can access the purchased services via various thin clients, such as browsers, tablets or mobile phones (Hill et al., 2013).

Apart from this categorisation, NIST has defined four **deployment models** that specify the intended user group of the cloud service (Mell & Grance, 2011). Figure 1-2 depicts those four models.

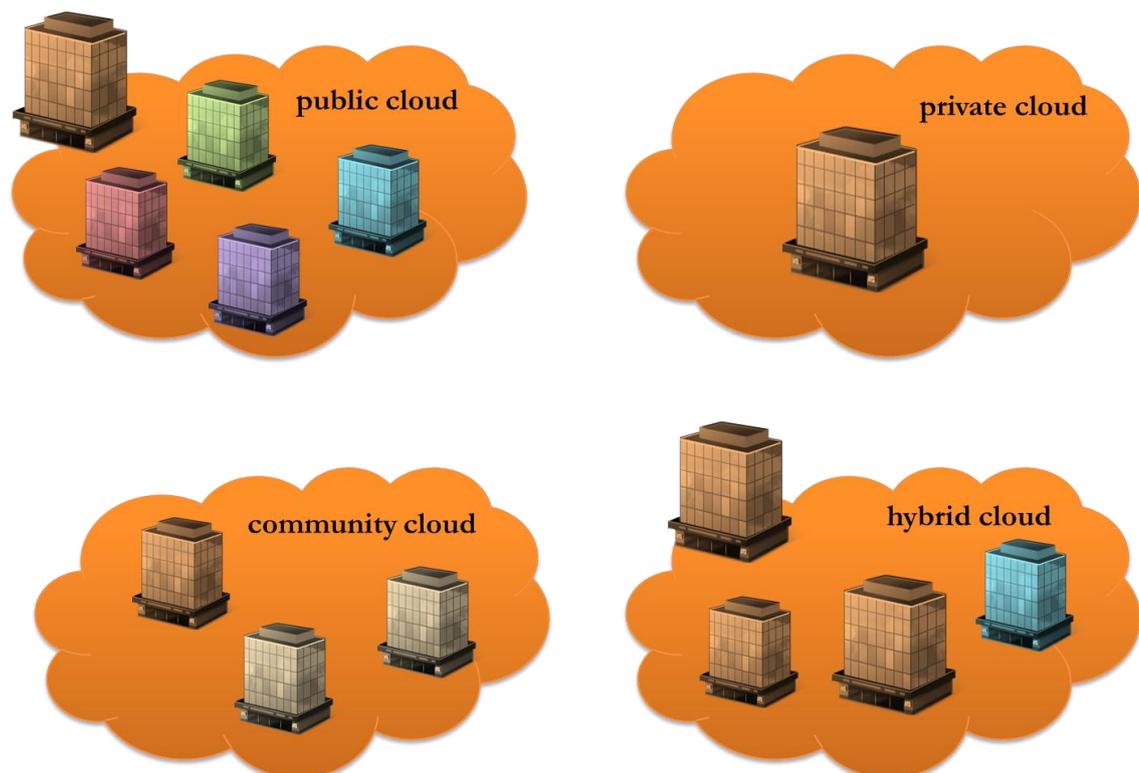


Figure 1-2: The four cloud deployment models

In the **public** cloud model, one provider offers the hardware and software in their cloud environment to several customers. As multiple tenants are sharing the infrastructure, this model is also called multi-tenancy model.

A **private** cloud, on the other hand, is only used by one entity, which is usually, but not always, also hosting the cloud in their own datacentre at the same time. This model can be chosen by big companies that have enough resources to set up a private cloud environment on their premises; however, it is also used as a remote option by smaller companies that are very concerned about holding their data separate. Even if these companies decide to utilise cloud services by a third provider, the private cloud model ensures that their data are held separately from competitors' data and the public.

The third model, **community** cloud, is similar to a private cloud, but in this model the infrastructure is also shared with partners and other associated parties. A mixture of some or all of those three models is referred to as **hybrid** cloud (Hill et al., 2013; Mell & Grance, 2011).

By its nature, cloud computing involves entrusting one's data with another company. Therefore, data security and privacy play a great role for every party involved. Combining the service models with the deployment models creates different scenarios that implicate different risks (Hill et al., 2013). In order to support customers in managing these risks, several policies and best practises have been established or are being created by reputable organisations, such as the already mentioned NIST (Hogan, Liu, Sokol, & Tong, 2011), the International Organization for Standardization ISO (International Organization for Standardization, n.d.-b), or the International Auditing and Assurance Standards Board IAASB, which is part of the International Federation of Accountants IFAC (International Auditing and Assurance Standards Board, 2010).

There are many advantages of using cloud computing services, especially for small and medium sized companies that do not have the expertise and resources to set up a big IT infrastructure (Sahandi, Alkhalil, & Opara-Martins, 2012). A recent survey conducted by Spiceworks (2013) amongst 500 IT professionals working in small and medium size enterprises found that the main reasons for companies to adopt cloud services are reduced maintenance effort and cost, location-independent accessibility, and easy scalability. According to the survey, almost half of the respondents said they had already adopted cloud or planned to do so within the following twelve months. As a result, this creates a huge market for cloud providers like Google, Amazon, Salesforce, Microsoft, or SAP, which is estimated to reach US\$19.5 billion (NZ\$25 billion) by 2016 (451 Research LLC, 2013).

1.2 RESEARCH PARTNER SAP

SAP is one of the biggest business software vendors in the world (Pricewaterhouse Coopers, 2013) and was kind enough to support this research project. The company was founded in 1972 by five German computer scientists, who, as opposed to other business software at that time, had the idea to develop applications that could process data in real-time (SAP AG, n.d.-c). Today, around 66,500 SAP employees in 130 countries work on supporting more than 253,500 customers in 25 different industries, gaining a revenue of €16.82 billion (NZ\$26.82 billion) in 2013 (SAP AG, n.d.-f).

Over the past forty years, their portfolio has greatly expanded. For a long time their main focus used to be on on-premises solutions, a concept in which all applications are purchased by the customer and then installed in their own datacentre. Nowadays, SAP's portfolio also includes: mobile solutions for smartphones and tablets, which are connected to the customer's backend applications and mostly serve as a lightweight user interface for system access "on the go"; an in-memory database called HANA (**H**igh **P**erformance **A**nalytic **A**pliance) that can be used for fast analysis of huge amounts of data; and various cloud applications, which are mostly marketed at small and medium enterprises (SME) that want to use an established set of software but still save effort and money in setting it up and maintaining it (SAP AG, n.d.-d). As the topic of this study is cloud computing, the following paragraphs focus on SAP's cloud products.

SAP's cloud applications have been partly their own development, and partly added to the portfolio through acquisitions of specialised smaller cloud companies. For example, SAP purchased *Ariba* for their collaborative procurement, *Fieldglass* for their vendor management and *SuccessFactors* for their human capital management solutions (SAP AG, n.d.-a; SAP News, 2014b). The two biggest product groups that have been created by SAP themselves are called *Business byDesign* (ByD) and *HANA Enterprise Cloud* (HEC). These applications include several sub-products that are offered to the customers separately, but share a general task set and system layout in the database.

Table 1-1 contains an overview of SAP's cloud applications and how they can be categorised in the models identified in the previous section. All business-related applications, such as ByD or the SuccessFactors products, as well as the *Ariba Business Network* and SAP's *Social Collaboration*, are SaaS products offered in the public cloud, with ByD also available as private edition. As defined above, a public cloud means that all customers share the same resources, whereas in a private cloud a part of the provider's resources are allocated to only one customer in order to increase their data security. Nonetheless, all SaaS applications are hosted in SAP's own datacentres or by certified third parties. This is not always the case for SAP's PaaS solutions, such as *HANA Cloud Platform*, *HANA One* or the *ByD Cloud Application studio*. These applications allow the user to develop their own HANA or ByD software, and for example HANA One can be hosted by Amazon Web Services (SAP AG, n.d.-e). Another PaaS solution is *SAP NetWeaver Neo* which is also known as JPaaS (Java PaaS). This platform can be used to develop applications that complement and extend existing on-premises solutions (Missbach, Stelzel, Gardiner, Anderson, & Tempes, 2013, pp. 134-135). SAP do not offer IaaS solutions, or community and hybrid deployment models.

Table 1-1: Categorisation of SAP's cloud products

model	SaaS	PaaS	IaaS
Public	All business-related applications (e.g., ByD, SuccessFactors products, Fieldglass) Ariba Business Network Social Collaboration	HANA Cloud Platform, HANA One ByD Cloud Applications Studio SAP NetWeaver Neo (JPaaS)	Not offered
Private	Most of SAP's own business applications are offered as a private model (e.g., ByD and ByD-like products)	(HEC)	

The reader will have noticed that the above mentioned HEC is written in brackets in Table 1-1. This is due to its special status which does not allow for customary categorisation. This product can be described as a basis for other SAP applications that a customer wants to run in the cloud with an underlying HANA database for faster transaction speed. For example, the *SAP Business Warehouse*, a data warehousing solution that can deal with huge amounts of data, or the *Business Suite*, which incorporates SAP's most popular business software solutions such as *Enterprise Resource Planning (ERP)*, *Customer Relationship Management (CRM)*, or *Supply Chain Management (SCM)*, can run on this platform (SAP AG, 2013). This deployment concept is comparable to a private PaaS model that has a pre-defined range of applications that can be installed on it (SAP AG, n.d.-b).

Due to their historical focus on business software, SAP do not offer pure IaaS solutions, since that would be comparable to hardware delivery. Although there is a product called *HANA Infrastructure Services*, which is part of the HANA Cloud Platform and basically provides the customer with a HANA database, this solution still allows only HANA-based applications to be developed on it, which categorises it as a PaaS model (SAP AG, n.d.-g).

Furthermore, community and hybrid deployment options are not offered by SAP as such. However, there are several possibilities for customers to combine different solutions, even across platforms; an example is when a customer decides to keep financial and human resources data in their own datacentre using on-premise software, but consumes cloud solutions for less sensitive data or where easy scalability is needed. The different applications can be connected with each other so that it is possible to exchange data with the customer's existing on-premises IT infrastructure.

1.3 RESEARCH OBJECTIVE

The original idea for this project grew within the business environment of SAP, as they identified a need for comprehensive software that would allow them to efficiently monitor their customers' data backups. They want to be able to guarantee absolute data safety and security for their customers, including infallible backup and recovery of data. This is also how they get certified by accredited auditors like *KPMG*, *Price Waterhouse Coopers*, and *Grant Thornton* according to the industry standards published by ISO and IAASB which are mentioned in section 1.1. SAP already have automated tools in place that support them in most of the operational tasks that come up in their day-to-day business regarding cloud services; however, further automation is necessary in order to cope with the increasing number of systems. As it is explained in section 1.2, SAP's cloud landscape is very wide and heterogeneous due to the development of new applications and several acquisitions of other companies. This creates a challenge for SAP as their current tools are not suitable anymore in terms of comprehensiveness and efficiency. What is more, most of the acquired companies had their own software in place to monitor data backups (while others had these processes outsourced altogether), which further complicates a unification of procedures and the creation of a single central management office.

In order to integrate all different applications into one solution, a strategic project called *One Delivery* was launched in 2013 by SAP's upper management that will affect all cloud products and in all functional areas. However, due partly to the immense differences between those areas, this project will not be completed before 2016 (also confirmed by participant 7 in the interviews). What complicates this process even further is that SAP enter unknown territory with this project. Even though comparable cloud providers like IBM or Cisco also expanded their portfolio by acquiring other vendors and their applications as SAP did (Cisco Inc., n.d.; IBM Inc., n.d.), no company could be found that would have ever completed a similar project as a consequence of their growth and published their findings afterwards. Therefore, the findings and insights of this study will be fundamentally new and pioneering in this area, so the expected contribution to the body of knowledge is unique.

The main reason for SAP to consolidate their cloud application management is their responsibility towards their customers in terms of data security. In the area of backup of ByD-like systems, for example, they guarantee that firstly, at least 98% of their systems have a successful backup every day, and secondly, no system backup fails for three or more consecutive days. In order to prove that they meet these criteria, they are certified

semiannually by external auditors that are specialised in compliance and risk management, such as KPMG. This certification is achieved by passing an audit in which SAP's cloud compliance department present their analysis of possible risks, a detailed plan about which controls are in place to mitigate those risks, and the monitoring methods for risks and associated controls. These mechanisms are then analysed by the auditors for their efficacy and thoroughness, and tested with the help of spot checks. However, this process is as divergent as the various, independent cloud compliance departments throughout SAP, so that each department has their own audit process at present. As part of the One Delivery strategy, they will be consolidated into one team, creating one central hotspot responsible for cloud compliance, security, and risk management, and eventually there will be only one audit that covers all of SAP's cloud applications (as also confirmed by participant 7).

The practical application of this research is to support the implementation of the One Delivery strategy in exactly this aspect. Therefore, the main objective of the study is to develop and implement an automated solution that can deal with all varieties of technical entities and tasks that arise while managing their cloud infrastructure. This study scrutinises a particular aspect of that, namely the area of backup monitoring. For that, a framework will be developed which addresses the issues raised above. The framework will be implemented in the SAP environment in order to evaluate its usefulness and efficacy.

The process of backup and recovery is one of five major areas of SAP's cloud security operations as depicted in Figure 1-3. The other four areas are physical security, network security, support of customers' compliance, and confidentiality and integrity (SAP Cloud Compliance, 2013). However, in these areas, company-wide processes are already in place and tools are set up that can be easily combined and that are also compatible with most of the newly added cloud applications, so that relatively little effort is needed to update those areas. Therefore, the topic of backup was chosen for this study because it is a crucial point of operations, and, additionally, still requires much manual effort to provide evidence for audits. It was for the latter reason that a small-scale automated tool, focusing only on backups of ByD systems, had been implemented by the researcher as part of her job with SAP in late 2012 to support the monitoring staff. As that software tool provided valuable practical insights on how to overcome potential challenges, it is described in Chapter 3 as a pre-study that informed the research in this study.

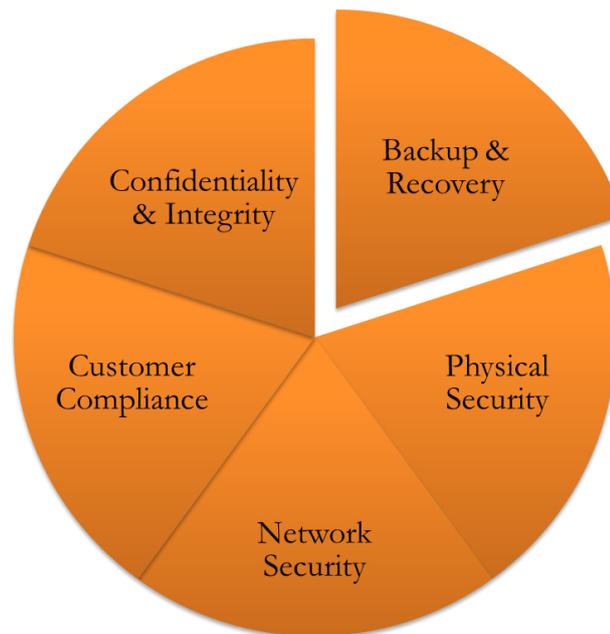


Figure 1-3: The five areas of SAP's cloud operations

The research is embedded into a widespread collection of different cloud management tools. SAP have 16 datacentres around the world (SAP News, 2014a), most of which were part of the acquisitions they made, and quite a few that are yet to be opened. The software tool that is currently used to manage all systems in the “original” SAP datacentres located in St. Leon-Rot (Baden-Württemberg, Germany) and Newtown Square (Pennsylvania, USA) is called Global Management Portal, or GMP, and was developed internally. It allows the technical employees to set up new systems, customer tenants, or filers, and also to define and monitor the respective backups. At the moment, this system only covers ByD-like systems properly, although attempts have been made to also include SuccessFactors systems. However, due to their different technical structure, they could not be fully incorporated yet. There is another system in place called Technical Infrastructure Controller, or TIC, which is technically a replication of GMP covering the HEC systems. However, the employees managing the HEC systems also use – and more frequently than the TIC – a system called SAP In-House System Manager (SISM) to operate their system landscape. In addition, there are other tools used by the companies that have been acquired by SAP. This creates a very heterogeneous landscape of management systems and it has not been agreed yet which of them will be used for all applications in the future. Consequently, the technical outcome of this study will have to be universally deployable, and possibly adaptable to many different environments.

This universality, on the other hand, will also make the study outcomes useful to other users, albeit mostly limited to other cloud providers. As the sector of cloud computing for business usage is growing, vendor numbers will increase and existing ones will expand,

creating a demand for efficient and secure system management tools. Moreover, customers' increasing exigency in terms of security is strengthened and manifested with standards that are currently being developed to be suitable for the cloud environment in particular, such as ISO 27017 – Code of practice for information security controls based on ISO/IEC 27002 for cloud services (International Organization for Standardization, n.d.-d). This could impose challenges on smaller providers, as they may not have the resources to address these topics. Even if such companies just employ parts of the framework that is developed as a result of this study, they shall still be useful for them. Another objective of this research is, therefore, to create a framework that is independent from the actual implementation in the SAP environment. The implementation is rather seen as a practical application to evaluate the framework's efficacy. This will ensure that the result of this research project is also useful for a wide range of other cloud providers.

To sum up, this study aims at providing three outcomes. The first outcome will be a set of requirements that represent business experts' suggestions about monitoring customer data backups according to acknowledged and established security standards up to a level that makes it possible to create unassailable evidence for audits. Subsequently, the second outcome will be a software tool which is implemented in the SAP environment based on the requirements. As SAP are the research partner of this project, it should also create a direct value for their daily business by integrating with existing solutions and automating tasks that are at present still done manually. The third outcome will be a framework which is developed by generalising the workings of the software tool and drawing conclusions from evaluating the software tool for its usefulness in terms of supporting the backup monitoring process. In order to account for the main two challenges of the study, the three outcomes shall be able to deal with large amounts of heterogeneous data originating from different sources and being handled by different departments.

1.4 THESIS STRUCTURE

The thesis is organised as follows: Chapter 2 provides an in-depth literature review of available resources in the application area. As only a limited number of publications on the thesis topic could be found, related and bordering literature is consulted in order to outline the subject. This includes the issues of how data are backed up in traditional systems, ways of handling heterogeneity in other areas of computer science, technical and procedural guidelines on providing cloud services, and an overview of the most important cloud security standards that can be used as further guides. The chapter concludes by identifying and expounding the research gap that this study aims to address.

Chapter 3 explains the design of the research. The approach that was used in this study is called Design Science. This methodology is described in relation to the software development lifecycle, which was utilised to actually implement the automated tool. In addition, a former project that had been realised by the researcher in cooperation with SAP prior to this study is presented. The project involved the creation of a similar tool, but with a less comprehensive scope, as it was only created to support operational job-related tasks performed by the researcher. The findings and conclusions of this small-scale project, however, provide useful insights for the bigger project, so that it can be seen as a pre-study to this research. Therefore, the findings of this smaller project are stated, and incorporated in the methodology. Furthermore, the approach for gathering the requirements for the developed software tool is described. Several methods are outlined, with the conclusion that a combination of interviewing business experts and performing the backup monitoring task should be used. This chapter also explains ethical issues and the employed analysis technique.

The findings of the research are presented in Chapters 4 and 5. Chapter 4 focuses on the interview findings, whereas Chapter 5 describes the software tool that was implemented. First, the environment in which the software tool was developed is described in order to explain the technical terminology and the implementation restrictions. Second, the interview findings are stated and analysed, with the outcome being a list of requirements that the software tool has to fulfil. Analysing the requirements further also reveals interdependencies between them, and enables prioritisation. Based on the list of requirements, Chapter 5 shows how the software tool was designed in order to address the issues raised. It begins by describing the technical constraints and, based on that, states considerations for the design of the software tool. After that, the backend logic, the underlying database schema, and selected aspects of the user interface are explained.

Chapter 6 provides a discussion of the research outcome. It evaluates the developed software tool in terms of how it met the participants' requirements, and how its general functionality was assessed by the users. The second section analyses the design of the research methodology, indicating how representative the research is for the problem area. The third section evaluates the framework that was developed based on a generalisation of the software tool presented in Chapter 5. It discusses how well this framework addresses the research objectives, and if it answers the research question in an appropriate manner.

Chapter 7 summarises the study and shows the research contribution. It also presents the limitations that affected the research, and indicates directions for future research.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews relevant literature in order to show how the project is informed by and could add to existing research. Following the description of the background and the problem identification in the introduction, there are several aspects that need to be considered, namely backup technology, heterogeneity or diversity of systems and technical landscapes, procedures and guidelines for providing cloud services, and cloud security standards. These four areas are to be seen under the overarching paradigm of cloud computing, as it is shown in Figure 2-1. The box in the middle of the graphic, which is formed as an intersection of the four areas, indicates the thesis topic.

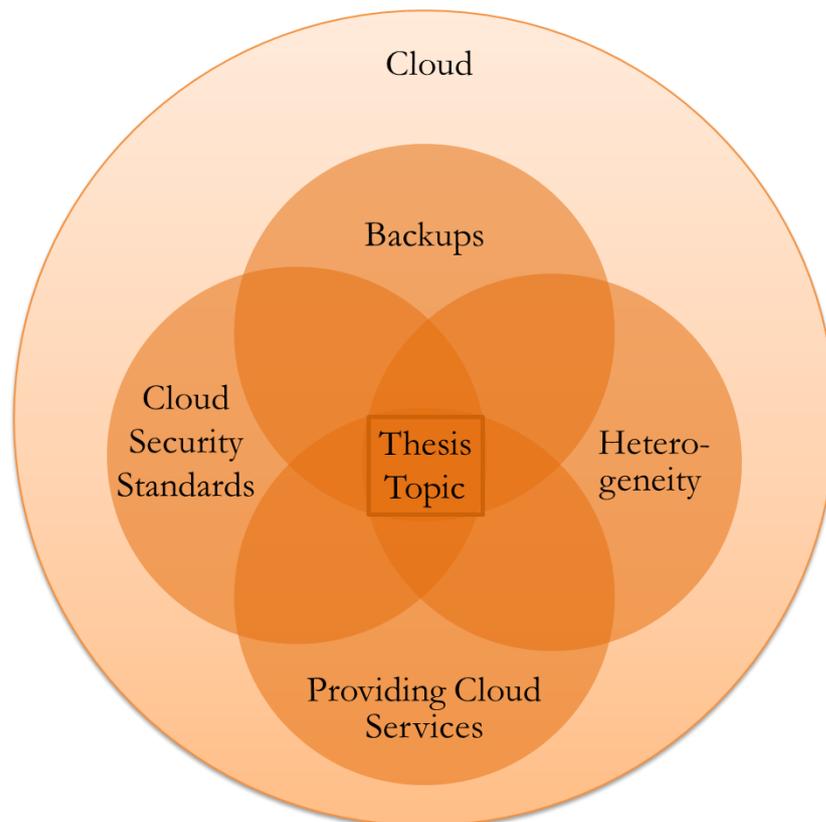


Figure 2-1: Research areas related to the study

This chapter dedicates one section to each of those four areas, starting with the topic of Backups and going clockwise on Figure 2-1 to Cloud Security Standards. The last section draws a conclusion from what was found in the consulted literature, highlighting the topics that are still underrepresented. As mentioned in the previous chapter, these topics are addressed by the research presented in this thesis.

2.1 BACKUPS AND BACKUP TECHNOLOGIES

Based on the combination of several definitions found in literature, this study uses the term backup to describe a “copy of a defined set of data” (Little & Chapa, 2003, p. 3) “taken at a particular point in time” (Nelson, 2011, p. 2) that is created at regular intervals and stored independently from the original data (Credo General Reference, 2006; Little & Chapa, 2003; Scriba, 2009). This section describes the historical evolution of backups, concluding with the current development towards cloud-based data storage. Details of important stages and milestones are explained together with their relevant contributions.

2.1.1 FROM TAPES TO SNAPSHOTS

A helpful resource for understanding the timely technical progress and changes of philosophies and methods is the relevant literature of those times, as it conveys the authors’ world views within the constraints of the environment and circumstances under investigation. This way, some very interesting insights were gained from a journal article by Rangachari (1992) on backups in enterprises. He defines the criteria that a good backup process should meet in order to be heterogeneous and scalable. This includes factors like easy addition of new systems, no limitation of storage space, centralised administration, and reasonable timeframes. However, more than twenty years ago, the term “reasonable timeframe” must be seen in the context of using magnetic tapes and being restricted to backups running over night in allocated timeslots, so that the author advises parallel writing and continuous data streaming to the tape drive to ensure finishing the process in a timely manner.

Using tapes as a storage medium decelerated the backup mainly due to the slow writing speed. However, tapes are still used for permanent storage, albeit with decreasing popularity. Nelson (2011, p. 49) shows advantages and disadvantages of this technology: Even though tapes are inexpensive, well-trying, easy to extend, and portable, they lack in robustness (especially if exposed to magnetic fields), reliability, scalability in terms of performance, and they are not compatible with modern network-based architectures as they can only be accessed by one server at a time. It is for these reasons that tapes started being replaced by disks during the last decade and that nowadays they are typically used as secondary medium for long-term storage (Nelson, 2011).

As globalisation proceeded, it became increasingly vital for businesses to provide constant availability of their services, thus, creating the need to overcome the burden of allocated backup windows in which all systems had to be shut down in order to create a consistent

backup. A technology called *snapshot* became popular during the last decade and is now standard in most systems. For example, an advertisement by Vinca Corp. (1995) still describes it as a novelty, whereas Nelson (2011, p. 2) already defines backups as “snapshot copies of data”, thus using both words as a synonym. Snapshots are point-in-time copies of data (M. Adams, 2001; Scriba, 2009). M. Adams (2001) and Nelson (2011) describe the snapshot process as follows: When a backup is triggered by a scheduler, all applications are temporarily quiesced after all pending transactions are completed. This is called “cold backup” (Scriba, 2009). Then, the relevant data are copied and first stored locally in order to reduce the time that is required for the backup to a minimum, so that the application can quickly resume. Afterwards, a mapping of the data and their respective address is created, and data are written to other storage media, creating the actual backup.

The main advantage of this technology is the radical reduction of offline time required to create backups, and also the easy recovery process (M. Adams, 2001; Hunter, 2004). Moreover, only actual changes are stored, making snapshots very efficient (Hunter, 2004; Nelson, 2011). However, Scriba (2009) also notes that snapshots of running applications (so called “hot snapshots”) can be inconsistent if only persistent information from disk but not the transient information from the kernel is included.

2.1.2 STRATEGIES FOR “GOOD” BACKUPS

With recent advancements in technology the perspective on backup is changing. Three fundamentally different approaches in terms of the extent to which source data are preserved in the backup are full, differential, and incremental backups. Although being used under different terminology in some of the literature (as in Little & Chapa, 2003, for example), the commonly agreed definitions (Credo General Reference, 2006; Nelson, 2011; Preston, 2007) are that a full backup is a copy of every file in the scope of the backup, a differential backup is a copy of all files that have been changed since the last full backup, and an incremental backup is a copy of all files that have been changed since the last backup of any kind. Scriba (2009) transfers this differentiation between full and partial backups to snapshots, calling them physical and logical, respectively.

Regardless of how it is performed technically, the underlying backup and recovery process must produce an accurate outcome relevant to the company’s needs. This can be achieved through proper planning and a backup strategy that considers the company’s rationale for making a backup; data scope; time, location, legal, and resource constraints for backups; and that identifies and mitigates all possible risks (Little & Chapa, 2003; Preston, 2007). It

is equally important to monitor the process, not only to immediately correct errors, but also to plan ahead for possible improvements and necessary changes (Little & Chapa, 2003; Nelson, 2011).

Unfortunately, in most companies only little attention is paid to the backup process until a recovery is required (Preston, 2007). This is due to the effort that is needed to create and maintain a robust backup process, which is seen as disproportional to its usefulness. Hence, it is often overlooked that the actual loss of data caused by insufficient backups can involve severe financial and non-financial risks, such as the loss of customers, orders, billing data, employee engagement (no need to work hard if their hard work is lost anyway), and trust in the IT department, the latter potentially causing employees to do their own backups, decentralising the process and consuming resources (Preston, 2007).

2.1.3 BACKUP AND DATA STORAGE AS CLOUD-BASED SERVICE

Since backups remained a critical point even after advances such as snapshot technology or the increasing availability of backup management tools (Preston, 2007), a way of conveniently outsourcing the process via a network was developed. The advantages of combining snapshot technology with networks were already mentioned by M. Adams (2001) and Hunter (2004), albeit restricted to local networks within the company. In addition, Little and Chapa (2003) see “virtual backups” as a clear trend in the future of backup technology.

Extending this architecture to make use of the Internet was not really an option at that time, as bandwidth, connection speed, and reliability of the Internet were fairly poor. Nelson (2011) sees issues with this option, as **Backup-as-a-Service** (BaaS) did depend on a stable connection and sufficient bandwidth in order to be competitive with traditional approaches in terms of trustworthiness and reliability. Furthermore, all data need to be encrypted as they are sent via the Internet (Heitmann, 2007), which can significantly increase the time that is needed to store and retrieve the backup data (Nelson, 2011).

However, utilising Internet-based services for doing backups also has some very innovative advantages. From a monetary and resource-oriented perspective, the same level of security can be achieved without having to invest large sums into technical infrastructure and personnel beforehand (Heitmann, 2007), so that costs are more easily predictable on a monthly basis. In addition, no special backup expertise is needed in the company anymore, as it can be relied on the provider to employ experts (Boomer, 2012). The latter factor also means that support is readily available, and that the services offered by a specialised

company should be more secure than what a “normal” company could provide (O'Bannon, 2012). To address the limitations of bandwidth, the amount of data that actually have to be transferred can be kept to a minimum by using compression techniques and establishing an efficient combination of full and differential or incremental backups (Boomer, 2012). Another advantage is the geographical independence of the backup to the original data, which creates an additional protection against server crashes or natural disasters (Boomer, 2012; O'Bannon, 2012). The remote storage also allows data to be retrieved from any location, which is especially important for location independent employees such as salesmen (Boomer, 2012).

As companies entrust their vital data with a provider, it is essential that they consider certain issues first. One of these issues is data security. This includes data encryption, access control, and compliance with relevant policies such as *SAS70* (Statement on Auditing Standards) and *SOX* (Sarbanes Oxley Act) that have to be met in accounting processes (section 2.4 presents these standards in detail). In terms of functionality, the provider should offer cross-platform support, easy scalability, and guaranteed availability of their systems (Heitmann, 2007). Furthermore, it is important that providers are trustworthy in terms of quality and economic stability, in order to avoid data being lost due to mishandling or insolvency (Boomer, 2012).

Backups have always played a vital role in data management, and techniques have been developed to make them more reliable and less resource-consuming at the same time. It is important to understand these developments in order to create even better methods in the future.

2.2 HETEROGENEITY

This section looks at the concept of heterogeneity, or diversity of systems, from two different aspects, namely technology and business. While the first focuses on integrating different databases, infrastructures, and systems to retrieve information, the latter looks at organisational issues that can arise when two companies merge, and how to overcome these issues.

2.2.1 HETEROGENEITY FROM A TECHNICAL PERSPECTIVE

The term heterogeneity has various meanings depending on the context in which it is used, but, in general parlance, describes the combination of dissimilar artefacts in a specified setting to achieve a shared goal. One definition found in (Nicolescu, O'Connor, & Piguet,

2012, p. 1) is especially relevant for the aspect of heterogeneity that this study looks at: “data heterogeneity, in computing, refers to a mixing of data from two or more sources, often in two or more formats”. As explained in Chapter 1, one of the goals of this research is to create a solution that can monitor backup states across several diverse system landscapes, meaning that system and backup data have to be extracted from various sources in various formats. Therefore, using this definition when elucidating the problem is beneficial to creating a relevant solution.

A related definition can be found in the area of *federated database management systems* (FDBMS). In order to understand the underlying concept, some papers from the early beginnings of federated databases were consulted. A detailed description of the architecture and a comprehensive guide on how to set up an FDBMS can be found in (Sheth & Larson, 1990). The authors define two different layouts of federated databases, which they call loosely and tightly coupled. As shown in Figure 2-2, the loosely coupled databases are all interconnected, which means that in order to process a user’s query, schemas of the connected databases are directly translated into the database where the query was created. In contrast to that, the tightly coupled architecture defines a central administrator that creates and controls the connections between system schemas. That way, a user’s query is not handled by their database, but is forwarded to the administrator which takes care of retrieving the necessary data from other databases.

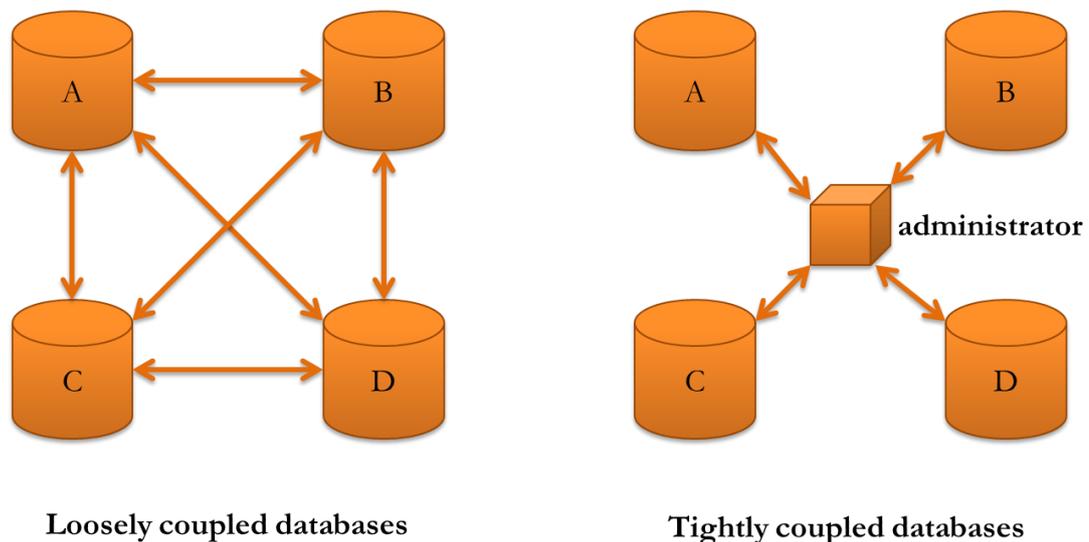


Figure 2-2: Architectures of federated databases

In a paper by Kamel and Kamel (1992), the authors elaborate on three design aspects of a database infrastructure employing tightly coupled databases, namely transaction management, system architecture, and schema integration. In their proposed methodology, databases are also tied together by common schemas. In contrast to Sheth and Larson

(1990), who had users access all data through their own database, here users access only the common schema in the central server. That way, all users send their queries through a central access point which would be in the position of the administrator in Figure 2-2.

The main advantage of FDBMS (compared with conventional distributed database management systems) is that the databases remain autonomous to the greatest possible extent, which is achieved by only adding a logical layer to each of them that creates the federation (Sheth & Larson, 1990). Before that, database management systems were connected by either replacing the old ones with a new system, or manually by the user, who would have to run queries to different databases in different systems using possibly different languages, and then create a meaningful result of that (Sheth & Larson, 1990).

Several concepts have been developed that work in a similar way with federated databases. For example, Li and Su (2001) propose to solve the issues of virtual applications in company mergers by employing *business object documents* (BOD) as their common entity to model all underlying schemas. BODs were developed by the Open Application Group and describe meaningful business entities (*Business Objects*) in a standardised way, which enables uniform communication between different databases throughout a company. As a second example, Srinivasan, Ngu, and Gedeon (2000) discuss the role of meta-level information to abstract from database objects, which is comparable to FDBMS due to the shared application domain of the connected databases. Consequently, they argue that those shared conceptual features can be used to create a common schema in systems where applications serve similar purposes. A third and very recent example is presented by Scott, Boardman, Reed, and Cox (2014), focusing on the application domain of research in materials engineering. In this area, many datasets are available as the results of many experiments, but due to their heterogeneity in size and format it is difficult for researchers to analyse them and extract the information they need. The authors propose a solution known as *Heterogeneous Data Centre*, which serves as a repository for researchers to share and access datasets. The strategy to deal with heterogeneity involves metadata, tagging, and a holistic search function that operates across all available data.

In all aforementioned examples the authors utilise an approach that is related to the federated database approach. Therefore, the work on federated databases will be most useful in designing a good solution to the research problem that this thesis focuses on. Not only is the context of retrieving information from different data sources similar to the one of this research, but also the suggested architectural and conceptual methodologies and their implementations show how this kind of heterogeneity can be addressed.

2.2.2 HETEROGENEITY FROM A BUSINESS PERSPECTIVE

As the integration and consolidation issues that SAP are facing also arose from several national and international acquisitions of other companies, conflicts at the organisational layer have to be taken into consideration. It is not just different technical infrastructures and specifications that could clash, but also company culture and personal viewpoints. Therefore, this section takes a brief look at how this problem is addressed in literature.

Maloney and Zellmer-Bruhn (2006) elaborate thoroughly on the topic. On one hand, they see many benefits of heterogeneous globally distributed teams, such as insight into local markets, broad professional and cultural knowledge, increased creativity in problem solving, and time-shared customer service. On the other hand, there are also several challenges faced by those teams, such as power struggles, lack of trust, interpersonal dislike, delayed communication (due to its limitation to emails or telephone), and cultural discrepancies, which can be caused by prejudices, stereotyping and a lack of will to understand each other's perspectives. These advantages and disadvantages arise from different dimensions of heterogeneity, which can be categorised as demographic (age, gender, country, culture, ethnicity), functional (work expertise, knowledge background), or hierarchical (company culture, power, administrative privileges). In their conclusion, the authors warn that the benefits of heterogeneous teams can be annihilated by attempts that aim at alleviating the described challenges, if the nature of the differences and their consequences are not understood.

Furthermore, Maloney and Zellmer-Bruhn (2006) distinguish between deliberate and collateral heterogeneity; the first one being intended by the manager creating the team and usually focused on the functional dimension, while the latter is more incidental and can usually be found in the demographic and hierarchical dimensions. While managers usually focus on creating deliberate heterogeneity, it is equally important to address collateral heterogeneity. The authors see problems in effectively doing so, as conventional strategies usually also diminish the benefits. Therefore, they propose several approaches around the concepts of self-verification and social integration to overcome issues caused by collateral heterogeneity without affecting the intended advantages. These include emphasising on shared goals and characteristics of team members to avoid sub-groups, recognising and acknowledging unique qualities and how they can be put to good use for the whole group, seeing team members as individuals rather than as representatives of their countries and cultures, and agreeing on team rules such as reliable and timely communication, sharing information with all team members, and being clear about personal matters that affect the

team, e.g., public holidays. These findings can be of great help when designing a new team structure for SAP's One Delivery project, and will be considered in this study.

Other work dealing with organisational heterogeneity focuses mostly on financial aspects. For example, it is noted that companies usually pay more for mergers and acquisitions than they can recoup from possible synergies, yet they still consider it as an option, so researchers examine reasons for this behaviour and try to analyse the outcomes with new methodologies (Bjorvatn, 2004; Shimizu, Hitt, Vaidyanath, & Pisano, 2004). Although this research is certainly relevant for managers when making strategic decisions, it does not apply to this case, as the acquisitions have been completed, and the challenges that lie ahead are more focused on integration. However, it has to be kept in mind that there are budget restrictions which will influence the overall evaluation of the One Delivery strategy.

2.3 PROVIDING CLOUD SERVICES

Since the One Delivery strategy, and the backup process as a part of it, are aimed at delivering services to cloud customers, another aspect on which this literature review needs to expand on is how to provide cloud services. While there are many guides and books available for companies that want to utilise cloud services, for example by Chang, Abu-Amara, and Sanford (2010), only few authors give a comprehensive overview in that way for the other involved party, the cloud providers. From what was found, most researchers seem to focus on certain aspects of the topic rather than using a general approach. This section attempts to bring together those that are most relevant for this thesis. Therefore, it is organised to first describe the concepts of virtualisation, multitenancy, and monitoring cloud systems, and then it gives an overview of the literature that takes a more comprehensive approach.

2.3.1 VIRTUALISATION

According to Raj (2012), virtualisation is one of the most important concepts for the technical architecture of cloud computing. In a virtualised environment, the physical (i.e., hardware), and computational (i.e., software) layers are decoupled (Jin et al., 2010; Marinescu, 2013), creating a “logical abstraction of physical assets” (Bauer & Adams, 2012, p. 16). That way, resources are partitioned and shared amongst users, leading to their increased utilisation, thus reducing costs and effort to manage them (Jin et al., 2010).

The typical layout of a virtualised infrastructure is depicted in Figure 2-3 and can be described as follows: The environment in which everything is located is a datacentre

containing a pool of memory and computing resources, networking appliances and storage systems (Bauer & Adams, 2012; Jin et al., 2010; Marinescu, 2013). All entities are managed with the help of a Virtual Infrastructure Manager. Data are stored on servers, which are often referred to as “physical nodes”. Each of these nodes contains several Virtual Machines (VM), which encapsulate web services and applications, and, depending on the type of VM, also operating systems (Bauer & Adams, 2012; Marinescu, 2013). For the user of such a VM, this creates the impression that they are accessing an actual computer, when, in fact, it is only an image of one (Bauer & Adams, 2012).

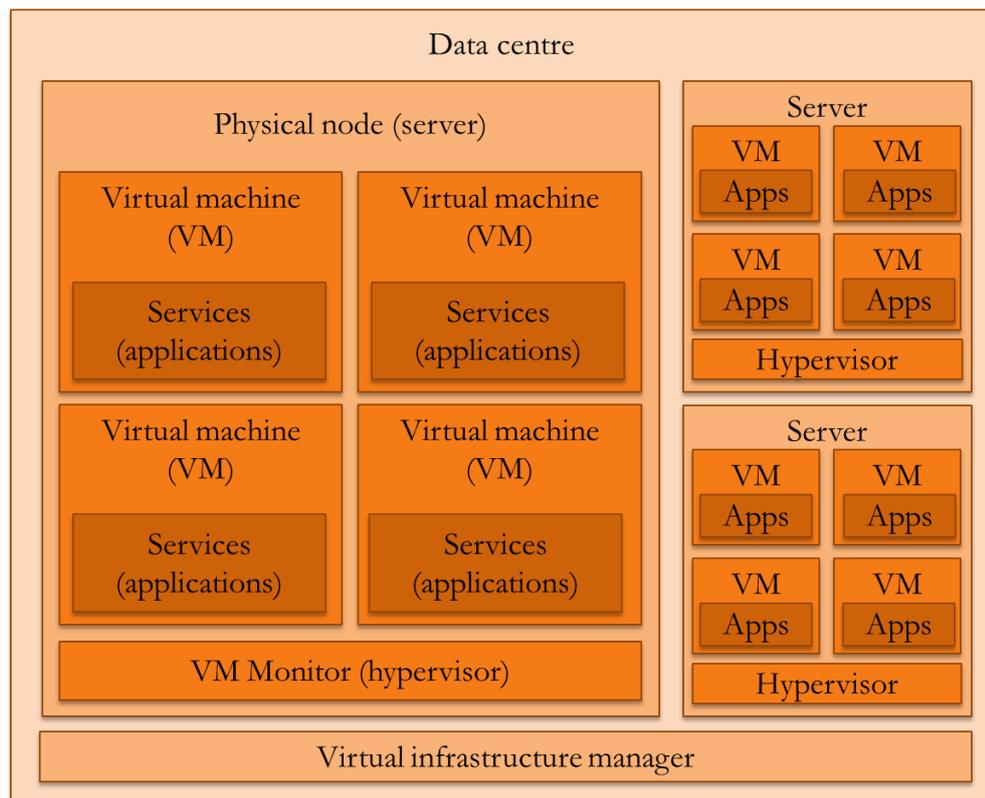


Figure 2-3: Virtualised infrastructure

The VMs running on one server are managed by a VM monitor, which is also called *hypervisor*. Depending on the type of VM, the hypervisor can either run directly on the hardware or on top of the operating system if it is shared by all VMs. In *full virtualisation*, the VMs are completely separated from each other and from their server, using different operating systems and, thus, needing a hypervisor that supports multiple operating systems (Bauer & Adams, 2012; Marinescu, 2013). A special form of this type is *hardware assisted virtualisation*, in which the hardware can interact with the VMs for selected operations, thus avoiding a processing delay caused by the hypervisor as medium (Bauer & Adams, 2012). The next type, rating the extent to which the VMs are connected to the hypervisor, is called *para-virtualisation*. In this, the operating systems on the VMs communicate with the hypervisor by implementing its interface. As a result, the VMs are not completely

independent of each other and, thus, are less easily interchangeable; however, para-virtualisation also increases the overall performance of the system since communication is more efficient (Bauer & Adams, 2012; Marinescu, 2013). The closest link between VM and hypervisor exists when *operating system virtualisation* is utilised. In this architecture the hypervisor runs on top of an operating system which has to be used by all VMs. Therefore, this type creates a stronger dependency, while at the same time further reducing operational resource costs (Bauer & Adams, 2012).

Employing the concept of virtualisation has several clear benefits: Firstly, the encapsulation and isolation of VMs means that the deployment of applications is flexible and fast. VMs can be moved from one server to another during runtime with little effort. This simplifies the maintenance of the physical components and alleviates the effects of hardware failure, creating a highly reliable and available system (Bauer & Adams, 2012; Jin et al., 2010). Secondly, the decoupling between infrastructure and applications supports scalability and workload management, as VMs can be added easily, without the need to simultaneously increase the number of servers (Jin et al., 2010). For this there are two options available, namely vertical growth, which describes an increase of resources for the VM to use, and horizontal growth, which describes the addition of new VMs to existing hardware (Bauer & Adams, 2012). Thirdly, the encapsulation of a client’s applications and data in a VM enables the client to configure “their” VM without affecting other VMs. This also increases their data security, provided that a reliable access control system is in place (Jin et al., 2010). Furthermore, encapsulation fosters the swift creation of system copies, which can then be used to set up a clone of the VM for another client, or as a backup copy, commonly known as snapshot (see section 2.1.1 for a description). Thus, the concept of virtualisation can help set up a secure backup and recovery process (Bauer & Adams, 2012).

2.3.2 MULTITENANCY

Multitenancy is a concept that is closely related to virtualisation. It is employed by cloud infrastructures and involves several tenants (users or clients) sharing the same database in which their data are held physically together but virtually apart from each other (Jiménez-Domingo, Lagares-Lemos, & Gómez-Berbís, 2011). There are three ways of setting up a database to serve multiple tenants. First, each tenant can have their own database with their own schema (thus, this form strictly speaking is not a type of database multitenancy). Second, tenants can share a database but still use different data schemas; and third, tenants can share both the database and a common schema. Jiménez-Domingo et al. (2011) recommend the third option due to its easy maintenance and low cost.

2.3.3 MONITORING CLOUD PERFORMANCE

Monitoring is an important factor in operating a cloud infrastructure, and is crucial to retaining its benefits (Katsaros, Kübert, Gallizo, & Wang, 2011). For example, factors like access control, resource usage, and the compliance with service level agreements should be monitored to avoid intruders, system overload, or dissatisfied customers, respectively. However, there seems to be a lack of tools available to support administrators, according to Katsaros et al. (2011). The authors analyse monitoring requirements from an architectural point of view. At infrastructure level, it is important to maintain an overview of physical resources to retain availability and scalability, whereas at application and service level it is important to monitor access control, applicable policies, and applications' performance; the latter is measured by response time, rather than by CPU usage, as virtualisation makes detailed technical monitoring at higher levels significantly complex (Katsaros et al., 2011). The authors recommend a layered monitoring infrastructure, which consists of three parts: gaining data from hardware, storing it, and then analysing and evaluating it.

2.3.4 CREATING PRIVATE CLOUDS

By studying how companies set up private clouds, insights can be gained which can be transferred to running cloud environments in general, and which can then be used to provide cloud services to a customer base beyond the company's boundaries. For example, Raj (2012) describes how to design and operate a cloud infrastructure from the viewpoint of enterprise architecture, and gives suggestions and instructions on many processes that have to be performed by the company running it. Although the author's discussion is mainly aimed at organisations that want to set up a private cloud which aligns with their current enterprise architecture, the recommendations can also be used by cloud providers who offer their services to external customers if the customers' architecture is known or standardised. A more detailed technical specification is given by Marinescu (2013), who describes how to properly set up the servers, develop applications that can run in the cloud, and manage and maintain the infrastructure. Their recommendation can be applied very well beyond a private cloud infrastructure, as this information is the foundation of offering cloud services. As another example, Bauer and Adams (2012) focus on making cloud systems as reliable and available as "traditional" on-premises systems. Based on a detailed risk analysis of the components of the cloud infrastructure, they make several suggestions and recommendations on how to alleviate and mitigate those risks by designing a robust architecture. The works reviewed above can be a good starting point for companies that want to offer cloud services commercially. In order to create a system that can be used by

business customers, these services have to be accompanied by legal and compliance policies, which are described in the next section.

2.4 CLOUD SECURITY

This section elaborates on security aspects in cloud systems. It is divided into two sub-sections; the first one gives an overview of commonly used most recent standards for which a company can be certified, while the second one discusses how a secure cloud system can be built.

2.4.1 INDUSTRY STANDARDS AND FRAMEWORKS

The popularity of cloud computing has led to an increased interest in how cloud systems can be made as secure as traditional IT systems, and several standards that intend to assess cloud service providers' security measures have emerged. The term *standard* is defined by the ISO as “a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose.” (International Organization for Standardization, n.d.-a) This section reviews the most recent standards as summarised in Table 2-1, based on the extant literature. The standards included are categorised as general and cloud-specific on one side, and as certifiable and not certifiable on the other side. In this context, “certifiable” refers to the existence of a certification process for the specific standard. The table does not include standards that are neither cloud-specific nor certifiable, as these are deemed not relevant to the topic.

Table 2-1: Categorisation of cloud standards

area covered	not certifiable	certifiable
general	(not relevant)	SAS 70, ISAE 3402, SSAE 16 ISO 27001 and ISO 27002 SOC 1
cloud-specific	NIST standards ISO 17788 and ISO 17789 ISO 19086-1	SOC 2 and SOC 3 ISO 27017 and ISO 27018

The upper right cell of the table shows standards that are not cloud-specific and for which companies can receive a certification if they are compliant. Due to the lack of cloud-specific regulations, in the early days of cloud computing companies had to employ standards created for financial or data privacy purposes, such as *SAS 70* (Statement on Auditing Standards), *ISAE 3402* (International Standard on Assurance Engagements), or *SSAE 16* (Statement on Standards for Attestation Engagement) (Gaskin, 2009, 2010; International Auditing and Assurance Standards Board, 2010). Standards that focus on general security of information systems were also used, for example *ISO 27001* in combination with *ISO 27002* (International Organization for Standardization, n.d.-b). Due to their reputation gained in other industries, these standards are still widely used to assess the security of cloud systems (see Waschke, 2012, for example). However, they were not specifically tailored at this task and, thus, disregard aspects like remoteness and virtualisation (Beckers, Côté, Faßbender, Heisel, & Hofbauer, 2013; Durbano, Rustvold, Saylor, & Studarus, 2010; Ristov, Gusev, & Kostoska, 2012). Attempts have been made to address those issues and improve existing standards, e.g. *ISO 27001*. For example, Beckers et al. (2013) propose a method to support providers in creating an information security management system by extending the existing *ISO 27001* with cloud-specific measures, based on an analysis of the standard's shortcomings.

In order to support providers more specifically in the area of cloud computing, roadmaps and guidelines on how to create secure systems were published, for example by *NIST* (Badger, Grance, Patt-Corner, & Voas, 2012; Hogan et al., 2011; Jansen & Grance, 2011). In addition, the ISO is currently developing a terminology (*ISO 17788*), and reference architecture (*ISO 17789*) for cloud systems, and also a reference framework for service level agreements (*ISO 19086-1*) (International Organization for Standardization, n.d.-c, n.d.-e, n.d.-f). These standards can be found in the lower left cell of Table 2-1. The drawback here is that they are not a sufficient assurance for cloud customers, as there are no certifications available that prove a provider's compliance.

One solution to the problem of non-certifiability is the SOC framework (Service Organisation Controls), which was developed by the AICPA (American Institute of Certified Public Accountants) (Singleton, 2011). It has three parts that are aimed at different target groups: *SOC 1* builds upon *SSAE 16* and, thus, mostly covers financial reporting controls (making *SOC 1* a more general standard); *SOC 2* and *SOC 3* assess the three security pillars confidentiality, integrity, and availability, with the distinction that a *SOC 2* report is designed to be read by experts whereas *SOC 3* is for the general public

(Singleton, 2011). As *SOC 2* and *SOC 3* cover cloud-specific topics, they can be found in the lower right cell of Table 2-1.

In addition to this, ISO are also currently developing standards that are cloud-specific and certifiable (shown in the lower right cell of Table 2-1), which will be filed under the numbers *ISO 27017* and *ISO 27018*. The latter is currently under publication and addresses the protection of personal information in public clouds (International Organization for Standardization, n.d.-g), a topic that is of increasing interest to the general public. In the context of this research, however, the most relevant standard is *ISO 27017*, which is currently in the draft stage (International Organization for Standardization, n.d.-d). It is based on *ISO 27002* and will define a cloud-specific control framework that can be used to implement an efficient security system. However, the standard is still at an early stage with specific details under discussion.

2.4.2 BUILDING SECURE SYSTEMS

This section explains how companies can be made compliant with the standards mentioned earlier, and then be assessed to prove their compliance. In the literature reviewed, significant attention has been paid to helping cloud customers who want to ensure that data they put in the cloud are safe (for example, see Marinescu, 2013; Onwubiko, 2010; Parthasarathy, 2013). The cloud provider's perspective seems to have attracted less attention, however, it usually agrees widely on important features to create a secure system. More specifically, most researchers note that due to its special architecture, cloud computing implicates many risks, such as unauthorised (physical and virtual) access to data, unknown storage location, data ownership issues, confidentiality breaches, network outage, or hacking attacks motivated by the large number of possible victims (Chaput & Ringwood, 2010; Dölitzscher, Reich, Knahl, & Clarke, 2013; Marinescu, 2013; Onwubiko, 2010).

In order to mitigate the risks, providers should implement standards such as the ones introduced in the section 2.4.1 (Beckers et al., 2013; Chaput & Ringwood, 2010). Several authors present tools (Chaput & Ringwood, 2010) or methods (Beckers et al., 2013) that can further support those implementations. In essence, they follow the process shown in Figure 2-4, which is explained in the following paragraphs.

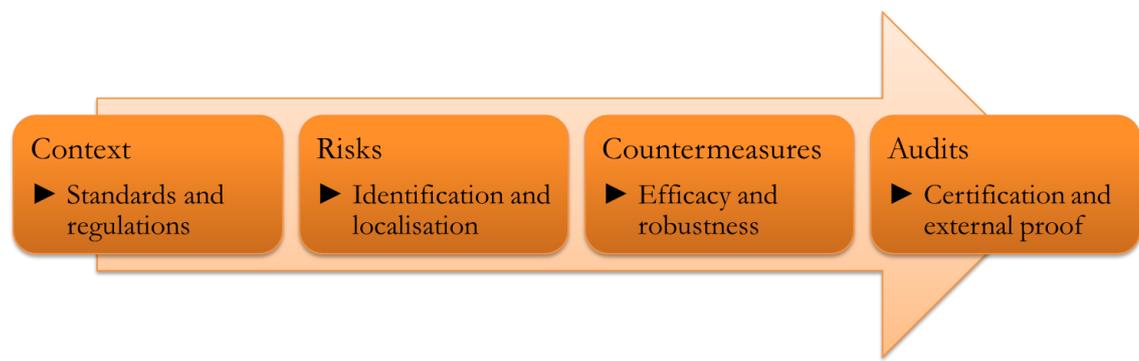


Figure 2-4: Process to create a secure system

First, the context of the cloud system has to be understood by all involved parties, including local and national laws (see Sotto, Treacy, and McLellan (2010) on US and European requirements), industry-specific regulations, and the sensitivity of processed data (Chaput & Ringwood, 2010).

Second, based on the context analysis, risks can be identified in several areas of the cloud infrastructure. Durbano et al. (2010) do this with the help of the Information Technology Infrastructure Library (ITIL) framework. This includes security relevant topics such as asset management of virtualised hardware, change management for hardware and patch management for software alterations, capacity management for scalability, backup and incident management, logging and monitoring, and communication management for safeguarded network usage, to name just a few.

Third, countermeasures have to be put in place, such as strong authorisation mechanisms in combination with data encryption and the definition and use of confidentiality levels (Chaput & Ringwood, 2010; Durbano et al., 2010; Onwubiko, 2010). Furthermore, all entities in the environment should have a globally unique identifier via which the entity's virtual and physical location can be retrieved (Durbano et al., 2010). This precaution in combination with a thorough event logging and analysis process and comprehensive monitoring tools across the whole infrastructure can detect and thwart suspicious activity from external and internal sources effectively (Dölitzscher et al., 2013; Durbano et al., 2010). Many of these countermeasures and their proper implementation are described in the standards mentioned above and would enable a company to successfully harden their system in terms of data integrity, availability, and confidentiality.

As the last step in the process, an external certification authority should be invited to analyse and evaluate the provider's risk management process and the efficacy of the proposed countermeasures in an *audit* (Dölitzscher et al., 2013). Audits should be held on a regular basis to reflect the frequently occurring changes of infrastructure and instances, and

they should take cloud-specific characteristics into account (Dölitzscher et al., 2013; Durbano et al., 2010; Onwubiko, 2010). The inspecting agency reviews the provider's compliance with a pre-defined set of regulations, and then compares their findings with the specifications of the standard for which the provider is seeking certification (Dölitzscher et al., 2013). This procedure is used for several reasons. Firstly, customers often lack the skills to evaluate a provider's countermeasures, whereas professional testers can evaluate the provider's countermeasures thoroughly (Dölitzscher et al., 2013; Halpert, 2011). Secondly, the professionally gained certificate creates trust among customers as they can expect a certain level of security to be in place (Beckers et al., 2013; Chaput & Ringwood, 2010; Onwubiko, 2010). And thirdly, the number of customers is usually too high to conduct individual inspections as this requires effort, resources, and time from both sides (Halpert, 2011); however, the audit results are freely available to all customers (Onwubiko, 2010).

The audit process and its implications are a central theme within this thesis, as they are SAP's main motivation to request a tool that simplifies the collection of audit evidence for the backup process. For their cloud infrastructure, SAP hold certifications for ensuring highest availability of a datacentre, quality management and improvement of operational processes, energy efficiency in datacentres, and also ISAE 3402, SSAE 16, and ISO 27001 (SAP Cloud Compliance, 2013). For the latter three, an audit is conducted every six months. SAP's datacentres have a robust architecture, multiple fall-back mechanisms for different types of disasters, and an enhanced access control process. Even though backups are transferred to a geographically distant datacentre, customer data stay in the same jurisdiction by having several of those datacentres in major markets like Germany and the USA (SAP Cloud Compliance, 2013). For these reasons, SAP can advertise their cloud security procedures as being of very high quality.

2.5 THE RESEARCH QUESTION

As stated in Chapter 1, the research objective of this thesis is to design and evaluate a solution that can help cloud vendors monitor the backup process of their customer systems in order to provide evidence for audits. In the literature review it was found that work has been done in four related areas and the thesis can build on the outcome from those works.

First, the literature on backup is very valuable for understanding the technical foundations of the problem that shall be addressed. This means that the problem itself can be better understood.

As a second aspect, heterogeneity plays a vital role for SAP due to the large number of cloud applications which they offer. Again, the literature helps to understand the problem to a greater depth, and also provides solutions for issues that emerge due to the differences.

For the aspect of providing cloud services, it could be noted that most studies seem to be targeting operators of private clouds; however, it was also found that the works reviewed can be of help in commercial environments as well.

Finally, the underlying architecture has to be secure, which is achieved by complying with established standards. This compliance can be certified in an audit. This is very important to SAP, as they can only sell their cloud applications if customers agree to entrust their data with SAP's datacentres.

As their customer base grows, SAP need to automate their audit process as much as possible if they want to maintain high quality of their services and do so in an efficient and cost-effective manner. This is something that is currently lacking in their backup monitoring process. Addressing the research question stated above needs to draw on all four aspects of the topic discussed previously. Such a combined approach was not found in the existing literature. The research question that this thesis aims to answer can, therefore, be formulated as follows:

How can we monitor backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment?

The research question shows the broader context of the project (business cloud computing), its major challenges (vast number of heterogeneous systems), and addresses a need SAP still see in their process chain, which is backup monitoring. Out of this, two sub-questions can be developed as follows:

What information is needed for the audits? In order to create an outcome that is relevant to the problem, the requirements need to be fully understood. Thus, this sub-question addresses the issue of what features to include and to exclude.

How will the outcome be evaluated? Since the outcome of the research has never been created in this form before, there are no established methods of measuring its success rate. Thus, a significant part of the research has to focus on how well the outcome answers the main research question.

2.6 CHAPTER SUMMARY

The literature review addressed four main aspects that can be associated with the research objective. It provided a short overview of the historical development of backup technology, followed by an analysis of how to address heterogeneity caused by technical and non-technical features. In order to understand the provider's perspective, section 2.3 presented the main concepts of cloud architecture. The next section introduced several important standards and explained how to build a secure cloud system. Out of this, the research questions were developed in the last section.

CHAPTER 3 RESEARCH METHODOLOGY

This chapter discusses and justifies an approach to answering the research questions formulated in Chapter 2. It begins by introducing the theoretical foundations of the methodology that was found to be most suitable to address the problem, followed by a presentation of a pre-study that was carried out prior to commencing this study. Based on the findings of the first two sections, the methodology used in this research project is described and discussed in the last section of this chapter.

3.1 THEORETICAL FOUNDATIONS

This section presents an overview of the theoretical foundations that were considered when designing the methodology of this research. As the creation of a software artefact was involved, three relevant ideas were considered, namely the Design Science Framework as the overarching methodology, the software development lifecycle as a more specific process of creating software, and a collection of requirements engineering methods which could inform the first two.

3.1.1 THE DESIGN SCIENCE FRAMEWORK

The term *Design Science* was introduced by Hevner, March, Park, and Ram (2004), who described it as a methodology based on engineering in which problems in information systems research are understood and solved by designing innovative artefacts. As opposed to routine building of systems, the outcome of the Design Science process “addresses important unsolved problems in unique or innovative ways or solved problems in more effective or efficient ways” (Hevner et al., 2004, p. 81) and contributes significantly to the body of knowledge by addressing problems that involve a certain level of complexity, flexibility and instability of surrounding factors, and dependence on human creativity and social behaviour.

In order to fulfil these goals, both the artefact and the process of its creation must evolve. Hevner et al. (2004) distinguish between four types of artefacts, namely constructs (i.e., definitions), models (i.e., representations of the real world which make use of constructs), methods or procedures, and instantiations (i.e., the implementation of a construct, model, or method in a system). According to the authors, the process of building the artefact is as important as the artefact itself, its typical parts being creation (i.e., design, verbalisation, and potentially development), and evaluation (e.g., formal proof, quantitative comparison,

qualitative analysis). In this, instantiations are most helpful when evaluating the artefact, as they provide the opportunity to apply them directly to the problem and assess whether the solution actually works.

As stated by Hevner et al. (2004) and depicted in Figure 3-1 with clockwise arranged icons, an artefact needs to include six essential parts. First, the purpose has to be identified and clearly formulated, and it has to be shown that the problem to be addressed is of scientific relevance. Second, an evaluation method has to be provided, assessing the advantages and shortcomings of the artefact in terms of utility, quality, and efficacy. Third, the artefact has to be innovative, i.e., it has to present something new or improved. Fourth, since scientific norms should be met, the artefact has to be well-defined, well-presented and consistent. This relates to the fifth point, according to which academic rigour has to be shown by employing a methodology that can be justified to find the best solution. Last, it is important to effectively communicate about the artefact to make it known to both the research community and a possible management audience (Hevner et al., 2004).

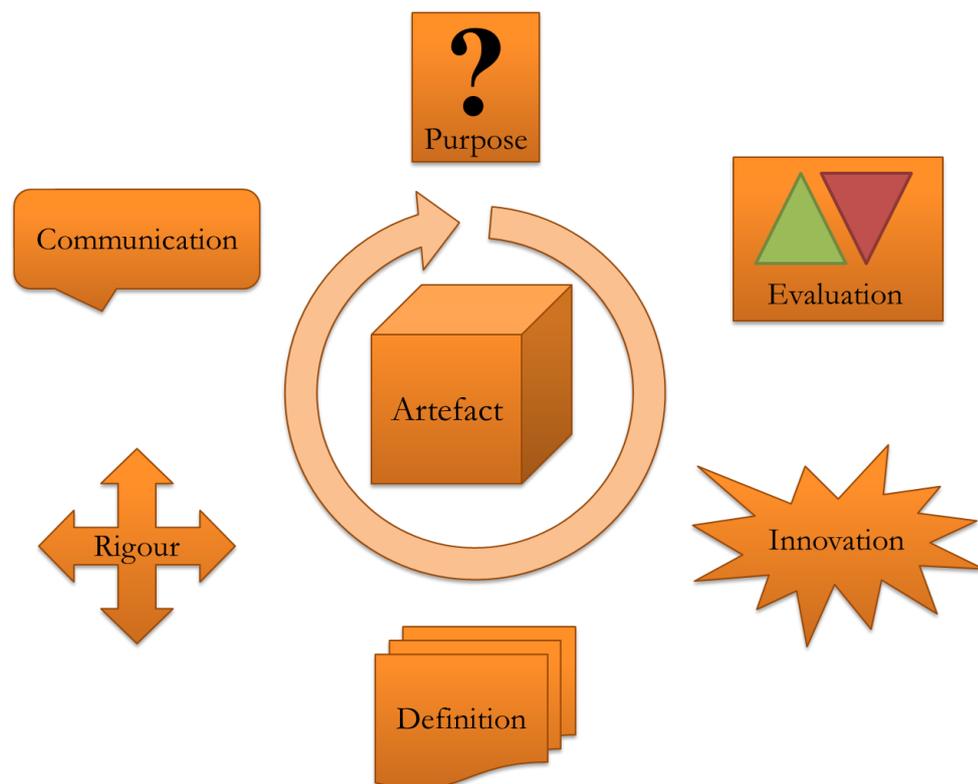


Figure 3-1: Design Science research guidelines: essential parts of an artefact

Peppers, Tuunanen, Rothenberger, and Chatterjee (2008) presented a methodology describing the steps of the Design Science process that meets these criteria. Their approach has been acknowledged by researchers in the field (Hevner & Chatterjee, 2010) and consists of six activities that can be executed successively. A flow chart of the methodology is shown in Figure 3-2. The first step is the problem specification, in which the aim is to identify the issues that shall be addressed, and to show the importance of the topic. In the second step, the research objectives are defined. These provide the foundation for the future evaluation. In the third step, the artefact is designed and developed, following three

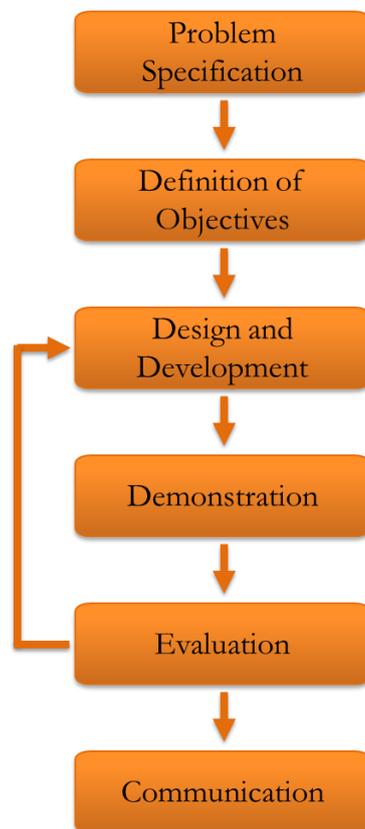


Figure 3-2: Design Science research methodology

of the earlier presented criteria, namely innovation, formal definition, and academic rigour. The fourth step is a demonstration of the artefact's ability to solve the identified problem. The result of the demonstration can be viewed as the first component of the evaluation, which is the fifth and, in terms of academic expectations, probably the most important step.

Hevner et al. (2004) suggest several approaches to how the artefact can be evaluated, including observational, analytical, experimental, testing, and descriptive methods. While it is most important that an artefact meets the goals stated in the second step, it can also be evaluated against more general objectives, such as its functionality, completeness, consistency, accuracy, performance, reliability, usability, and its fit with the organisation in which it shall be employed (Hevner et al., 2004). If the evaluation shows that the artefact does not fulfil the set objectives, the researchers have to go back to the design and development stage and eliminate the identified flaws (Hevner & Chatterjee, 2010; Peppers et al., 2008). The final step of the Design Science methodology is communication. This usually includes a scientific publication that shows the contribution to the body of knowledge (Peppers et al., 2008). Hevner et al. (2004) also see the necessity to communicate results effectively to a management audience, because managers are the ones who will eventually make decisions about deploying a certain artefact in their organisation. Such publications should include a less detailed technical description of the artefact, as their focus will be on the resources required for the artefact to be implemented (Hevner et al., 2004).

3.1.2 THE SOFTWARE DEVELOPMENT LIFECYCLE

Ever since computer programming came into existence, a need arose for some form of guidelines that would make it efficient and effective. Originally adapted from a US military control system, the first model that was academically defined was the waterfall model, as first described by Royce (1970) and later named by Benington (1983). This model defines the stages of the software development process as they can still be found in current guidelines (see Braude, 2004; or Pham, 2007, for example). It is depicted in Figure 3-3. At the beginning, a thorough requirements analysis is performed, leading to an operational plan and a list of specifications to be observed (Benington, 1983; Pham, 2007). This is followed by the design phase, in which the programmer tries to find the best possible architecture and programme structure to meet the requirements (Pham, 2007). The next step is the implementation, or coding, of the software. If the design phase was done well, the programmer can now focus on producing easily understandable and maintainable code with a detailed documentation (Pham, 2007).

After, or usually in parts during the implementation phase the software is also tested (Pham, 2007). For this, the tester can take two different approaches: in *white-box testing*, the code structure is visible so that individual elements can be tested, whereas *black-box testing* compares the desired output with the received one, without actually looking at how this output is obtained (Braude, 2004). Testing also depends on the scope of the code that is inspected, reaching from unit level (i.e., methods or classes) to intermediate level (i.e., collections of classes) up to system tests including the whole application (Braude, 2004). The final stage of the development process is the operations or maintenance mode, which is preceded by a “shakedown” (Benington, 1983), or simply the installation of the software (Pham, 2007). However, after this stage the cycle is not finished. Since about one fifth of all errors are not discovered before operation (Pham, 2007), they will impact the requirements analysis and development process of subsequent systems, which is symbolised by the arrow leading back to the first stage.

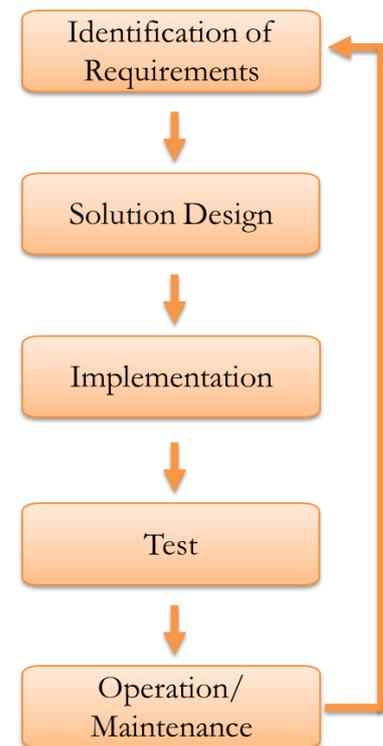


Figure 3-3: The software development lifecycle

Building the base for all other models, the waterfall model was constantly enhanced, for example by adding evolutionary stages into the steps, as first proposed by Gilb (1985), or by adding iterative testing cycles to detect errors at an early stage, as described by Tsai, Stobart, Parrington, and Thompson (1997).

There exist numerous other variations, and summaries and comparisons of those models focusing on different aspects are widely available. For example, Ruparelia (2010) gives a historical overview of main models such as the waterfall, incremental, spiral, agile, or V-model. It can be noticed that a common aim of all models is to decrease the implementation time, leading to higher flexibility and faster deployment of solutions. A very popular approach in this area is agile software development (Stober & Hansmann, 2010).

With this many models and methods available, it can be difficult for developers to choose which one suits their project best. Guntamukkala, Wen, and Tarn (2006) found that this decision also highly depends on the project itself. They conducted interviews with experts in order to assess the feasibility of certain models in specific situations. For this, they categorised popular models into the three categories heavyweight, middleweight, and lightweight, based on three features that a model could have or not (development sub-cycles, early prototype, and rapid feedback during design phase). These features would reflect a model's flexibility inasmuch as the more of them were included, the "lighter" a model would be (Guntamukkala et al., 2006). The researchers found that a project's initial situation correlates with the desired or required flexibility of the chosen model. Especially in projects that were characterised by constantly changing requirements and environments, and in which scope, time constraints, system architecture, and risks were not well understood, lightweight models were preferred, because they allow constant changes on most layers during the development process (Guntamukkala et al., 2006).

3.1.3 REQUIREMENTS ENGINEERING

This section takes a more detailed look at the requirements engineering process, which is the first step of the software development lifecycle described in the previous section and depicted in Figure 3-3. Gathering and thoroughly understanding requirements is paramount for the success of any project, because a problem can only be solved if its characteristics are known and well understood. Furthermore, requirements engineering is the foundation of all following processes, which means that mistakes are carried over into successive phases where they could cause further disruptions or even put the successful

completion of the whole project at risk (Ralph, 2013). Based on what was found in literature, in this thesis the term “requirement” is defined as an *attribute or constraint for a software system that is requested by one or more users of such a system and that will help said users to solve a problem or achieve an objective* (Kotonya & Sommerville, 1998, p. 6; Leffingwell & Widrig, 2003, p. 15; Wiegers, 1999, p. 6; Young, 2003, pp. 1-2).

As described in detail by Leffingwell and Widrig (2003, pp. 377-381), there are six essential skills necessary when managing requirements, which are shown in Figure 3-4. These six skills provide a useful outline for the requirements engineering process. First, the problem that shall be solved has to be analysed and defined with its boundaries and constraints. Second, users’ needs have to be understood, which is achieved by employing various methods for requirements elicitation, such as interviews, workshops, or storyboarding. As a third skill, a system has to be designed in a way that addresses the identified requirements, for example with the help of the use case model developed by Rumbaugh (1994). Depending on the characteristics of the project, in this step it may be more fruitful to use task descriptions instead of the more formal use case method (Lauesen & Kuhail, 2011); however, use cases can be more easily transformed into actual source code (Leffingwell & Widrig, 2003, pp. 291-293) or into test cases (Leffingwell & Widrig, 2003, pp. 305-309).

The fourth skill focuses on the overall management of the requirements process by giving advice on how to keep the project scope by effectively utilising prioritisation and negotiation. The fifth skill involves refining the system definition, which is based on the outcomes of the testing phase. As the last skill, the researchers recommend to create traceability of requirements back to decisions and sources, and the setup of an effective change management process. This is closely related to the overall management of requirements.

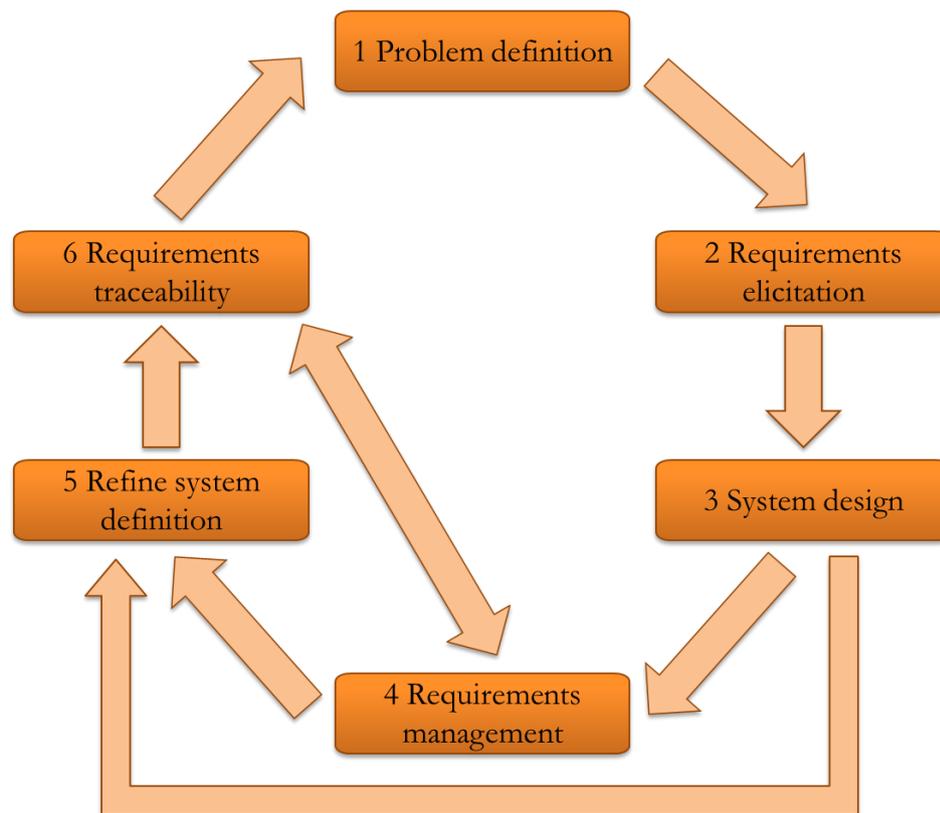


Figure 3-4: Six essential skills for requirements engineering

Within the context of finding a suitable methodology, special focus must be given to the activity of requirements elicitation, as it fundamentally shapes the outcome of the project (Wiegiers, 1999, p. 96). There are four aspects that must be understood in order to gather requirements that are relevant in the given setting: the application domain or background, the problem that shall be solved, the business processes involved, and the needs and constraints of system stakeholders (Kotonya & Sommerville, 1998, pp. 54-55). Based on this knowledge, requirements engineers can apply one or more of the variety of available methods that help them perform the essential tasks of requirements engineering: interacting with the customers and/or users (Kujala, 2005). Popular examples are scenarios, prototyping, focus groups, ethnography including observation, or a mixture of several techniques (Jiang, Eberlein, Far, & Mousavi, 2008).

In addition, one of the most straightforward and widely used methods is carrying out interviews, as they offer a great variety of different ways to be conducted, making them flexible and adaptable to many situations (Rowley, 2012). However, as noted by Gillham (2001, p. 11), in defining the interview questions it is important to consider the project constraints, e.g., complexity in terms of local distribution and number of participants, the depth or breadth of the expected answers, and the expected analysis effort, given the overall duration of all interviews. Another challenge especially in geographically distributed teams is the need for virtual, non-personal collaboration (Berenbach, Paulish, Kazmeier, &

Rudorfer, 2009; Knauss, Damian, Cleland-Huang, & Helms, 2014), which makes the decision for the best-fitting requirements elicitation method even more difficult.

After the requirements have been identified, they must be transformed into a meaningful form with the help of the analysis process. As mentioned, one of the most popular approaches for this is to create use cases (Bijan, Yu, Stracener, & Woods, 2013). Use cases offer a structured way of describing the possible interactions between the user and the system. They are particularly helpful when trying to develop a common understanding shared by users and developers, and to find interconnections between requirements (Leffingwell & Widrig, 2003, p. 148; Wiegers, 1999, pp. 137-138). However, it may not always be helpful to transform requirements into use cases, because the pre-defined structure of use cases restricts the requirements analysts in their creative thinking, which could thwart flexibility and a broadminded view of possible outcomes (Lauesen & Kuhail, 2011). It is also important to thoroughly validate the requirements elicited with the active involvement of all stakeholders (Kotonya & Sommerville, 1998, pp. 94-95) so that the risk of misunderstanding and misinterpreting requirements is minimised (Kotonya & Sommerville, 1998, p. 88).

3.2 A PRE-STUDY

This section presents a small-scale preliminary study that was done about a year before the actual thesis project started. First, the study and its parameters are described. This is followed by a summary on how the pre-study informed the research presented in this report.

3.2.1 PROJECT DESCRIPTION

The preliminary project was carried out as part of the researcher's job with SAP from October 2012 to January 2013. Its primary aim was to create an easy-to-use system to collect and present the evidence of backup monitoring needed for the regularly conducted audits. Prior to using the software tool developed by the researcher, the operational team used spreadsheets and a rudimentary form of reporting that involved significant manual processing such as data comparison and fault rate calculation. The average time to create a report, which was at that time only done on a monthly basis due to the time-consuming effort, was usually between 90 minutes and two hours, but could take up to four hours if the number of investigated systems was high, and if failures needed further investigation. The results of the reports were not easily reproducible due to a lack of historicisation; in

particular it was difficult to keep track of system status changes, e.g., from “setup” to “live” or of a change caused by an upgrade to another version. This also led to inaccuracies in the calculation of success and failure rates, as the so called *basic population* (the number of systems in scope for the audit) would change daily which was not represented in the reporting.

The project was carried out in several stages. At the first stage, the researcher was made responsible for the task of creating and sending out the monthly reports in order to fully understand the business requirements by performing the task herself. After that, the tool was designed and developed, focusing on the scope that had been used for the monthly reports. Figure 3-5 shows the very first screenshot of the tool. As it was based on the former manual reporting, it required a date range as user input as well as a selection of the system version (left column), and the system status (right column). The system version was based on the feature pack (FP) or release of ByD, whereas the system status is indicated by a so-called ZH-code. There are several of these codes which can be assigned to a system in various stages of its lifecycle. The four important codes for the tool were ZH001 (shared productive live system), ZH006 (private productive live system), ZH012 (productive live system running the application Sales on Demand; now obsolete), and ZH014 (productive live system running the application Travel on Demand; now obsolete). The tool also allowed exploring the data further by providing lists of scheduled systems and systems that had a problem with their backup.

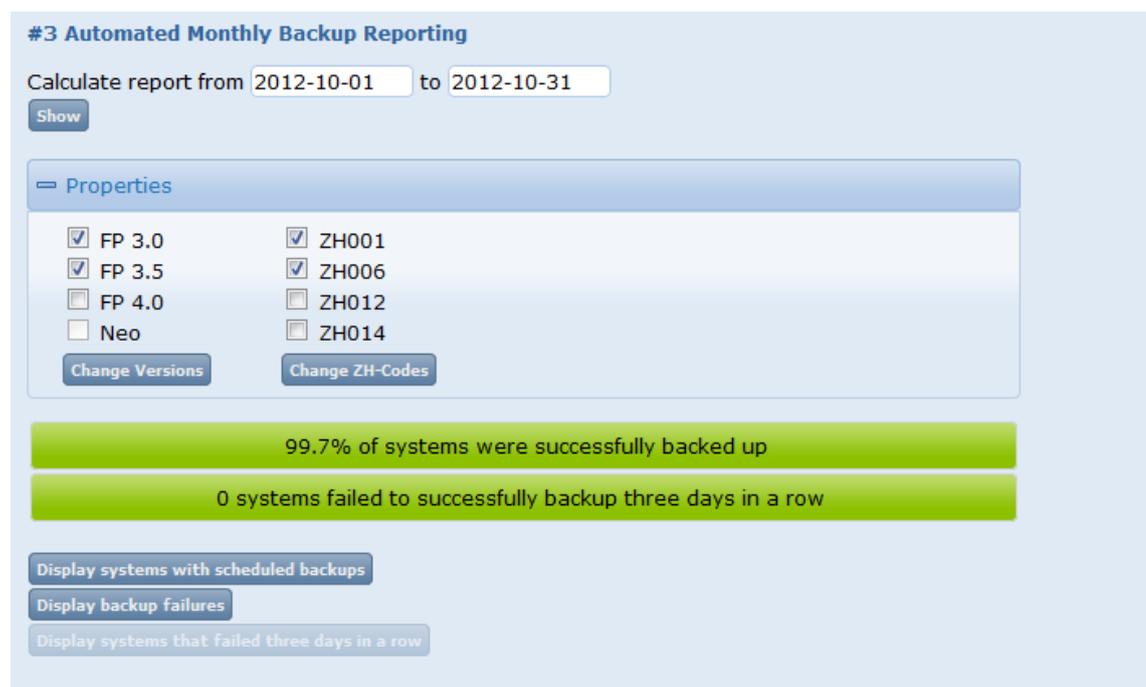


Figure 3-5: First screenshot of the tool developed in the pre-study

The researcher was the only full-time member of the project team. Other employees, such as the future users of the reporting tool or developers working on the management system in which the tool would be embedded, were frequently consulted during the whole project. Once the tool was developed, it was immediately used to produce the reports due next, which were compared to the manual ones. The main benefits of the tool are that it is considerably faster and more accurate than the manual process, and that it supports historicisation by storing the states of each system and its backup for each day, creating reliable and reproducible audit evidence.

3.2.2 PROJECT FINDINGS

The significance of the presented pre-study for this project does not only lie in its similar application area, but also in the conclusions that could be drawn from it in terms of developing a methodology and a solution approach. Even though it was only a small scale project that involved only one of the SAP software applications (*Business byDesign* or ByD), the feedback that was received was very valuable. For example, it was found that the list of systems with failed backups makes it easy for the user to locate and repair underlying problems. At the same time, users asked for an automatic linking to the system's data, and a quick indicator that would show if the problem still exists. This request was fulfilled by making the system ID clickable, loading a new window with more information about the system, and adding a coloured circle in each table entry indicating the status of the system's latest backup (e.g., "*successful*", "*failed*", or "*running*"). Furthermore, users appreciated that the reporting could now be done by taking simple screenshots. However, this would initially require the selection panel above the results to be included in order to show which checkboxes had been selected for this specific result. This issue was resolved by adding a statement above the result boxes that would summarise the scope of the current request. The current layout of the reporting tool is shown in Figure 3-6.

#3 Automated Monthly Backup Reports

Calculate report from to

Systems with ZH-codes

<input checked="" type="checkbox"/> ByD Version 1302	<input checked="" type="checkbox"/> ZH001 - Productive System
<input type="checkbox"/> ByD Version 1305	<input checked="" type="checkbox"/> ZH006 - Single Customer System
<input type="checkbox"/> ByD Version 1308	<input type="checkbox"/> ZH012 - Sales on Demand (3.0)
<input type="checkbox"/> ByD FP 3.0	<input type="checkbox"/> ZH014 - Travel on Demand (3.0)
<input type="checkbox"/> ByD FP 3.5	
<input type="checkbox"/> ByD FP 4.0	
<input type="checkbox"/> CRM on Demand (Sales on Demand)	
<input type="checkbox"/> Financials on Demand (My Money)	
<input type="checkbox"/> Payroll	
<input type="checkbox"/> Travel and Expense Mgmt on Demand	

Systems without ZH-codes

Results for ByD Version 1302 with ZH-codes ZH001 and ZH006 from 2014-02-17 to 2014-02-23:

100.00% of systems were successfully backed up

0 systems failed to successfully back up the third day in a row

Figure 3-6: Current screenshot of the tool developed in the pre-study

Over the past few months the tool has attracted the attention of other departments and was being used by them too. Although this is a very positive development, several issues arose. Since the number of systems that are included in the scope of the reporting has increased, the underlying logical data schema has been found to be insufficient to present the different types of systems. For example, there is one type of systems that does not use ZH-codes; this was not anticipated before the development of the tool and that has caused the need for a second fold-out set of checkboxes below the first one as seen in the screenshot. Another shortcoming lies in the layout of the reporting, which was not intended to capture the current variety of applications, making it look cramped and more complex to use. In addition, the data display was not properly separated from the processing layer, which means that although there are no productive systems with codes ZH012 or ZH014 anymore, the respective checkboxes still have to remain.

The created tool fulfilled its purpose of easily providing reliable and accurate audit evidence, as was proven during several audits. The requests from other departments to make it usable for their cloud applications show that its working concept is successful. Another useful insight is that the used methodology was adequate to discover most requirements that were relevant to its initial context. This methodology consisted of a mix of the researcher performing the task that the tool shall automate, and interviews with stakeholders.

3.3 DESIGN OF THE RESEARCH

This section presents the methodology that was developed in order to address the research question formulated at the end of Chapter 2: *How can we monitor backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment?* Two sub-questions were also formulated:

- (1) What information is needed for the audits?
- (2) How will the outcome be evaluated?

The Design Science framework was identified as being most suitable for the research, because it was anticipated that the outcome would be an artefact that was innovative and improved the current process in a way that had not been done before. Therefore, the outcome could either be a construct, a model, a procedure, or an instantiation of any of these. Due to the nature of the research, however, it was most likely that the outcome would be piece of software, hence, an instantiation of a model that follows a certain procedure. As noted in section 1.3, this model will also be developed in the form of a framework.

The Design Science process was adapted to incorporate the software development lifecycle, as shown in Figure 3-7. It is a combination of the two process flow charts in figures 3-2 and 3-3. The process started with the problem specification and definition of the objective (as elaborated on in Chapters 1 and 2). The design-and-development phase was divided into three activities which were taken from the software development lifecycle. At the beginning, a systematic identification of requirements was performed, providing a well-informed foundation for the design of the solution. The latter was followed by an implementation and testing step. The arrow pointing from the step of requirements identification to implementation and testing indicates that the requirements significantly informed the third phase, because the requirements were the base against which the software tool was tested. Once the development was finished, a demonstration was given to the stakeholders of the project in order to show if the created artefact solved the problem, i.e., functioned according to the users' specifications. This was followed by a thorough evaluation to assess whether the artefact met the research objectives and answered the research questions. The dotted arrow pointing from the evaluation step back towards design and development indicates that, should the outcome of the evaluation show that the problem had not been addressed, the design-and-development stage would have to be repeated. The final step, communication, had to include the generalisation of the

software tool to a framework, so that it would also be applicable outside of the SAP environment in other areas.

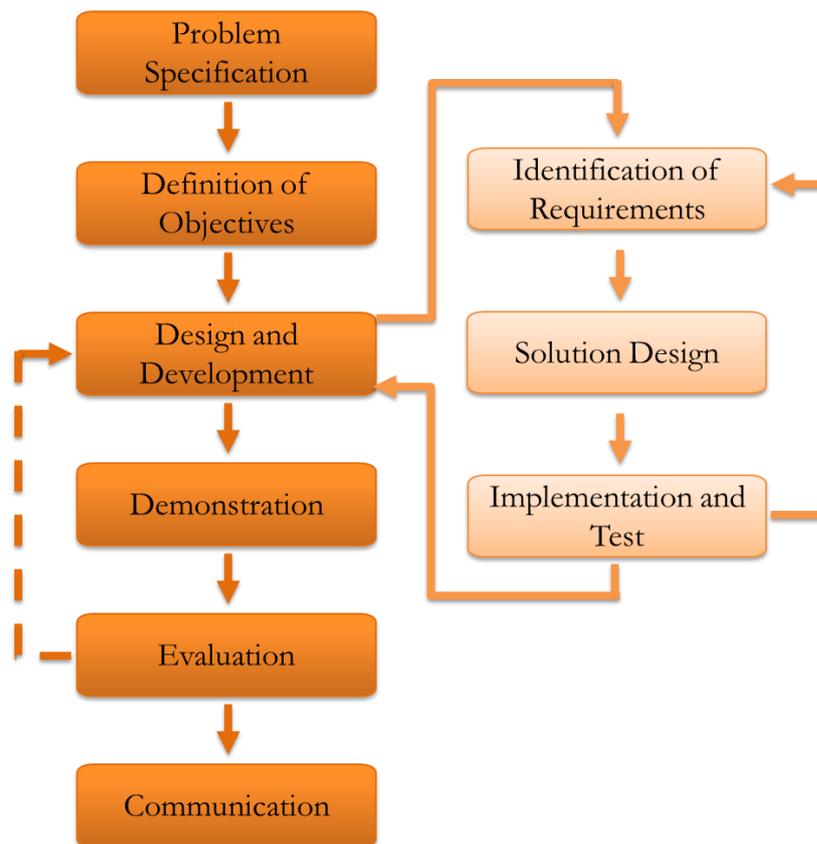


Figure 3-7: Methodology framework

As already pointed out, meticulous requirements elicitation, analysis, and validation were paramount for subsequent software development activities. Therefore, two different approaches, which look at the problem from different angles, were combined. The first one involved the executive responsibility given to the researcher to perform the current backup monitoring process. This was based on the finding from the pre-study that the insight gained by this practice was helpful when creating the software tool; as the researcher was fully integrated in the process, she understood the requirements much better than an uninvolved developer. A limitation of this approach was that it was not feasible for the researcher to have this responsibility for the backup monitoring of all of SAP's cloud applications. However, since processes are being made uniform due to the company's One Delivery strategy, it could be assumed that they would not differ too widely.

The second approach used for the requirement engineering process was interviewing business experts as their opinions were regarded as important and valuable for the research. However, they were geographically dispersed and, therefore, "electronic", or IT-facilitated interviews had to be used, providing an acceptable ratio between cost and result. A new

videoconferencing tool that had been implemented globally throughout SAP locations was utilised, meaning that non-verbal communication was also included, leading to a more comfortable situation for the participants and the researcher. Ethical approval for this research was obtained from the Auckland University of Technology Ethics Committee (AUTEK) on 24th February 2014, AUTEK application number 13/315 (Appendix C).

The interviews followed a semi-structured approach, in which the researcher prepared questions to provide a scaffold for the conversation. The questions could be altered according to the direction that the interview takes. The following questions were expected to cover all areas that were necessary to gather requirements for the artefact:

- (1) What is your job title?
- (2) With which application do you work?
- (3) What are your main responsibilities?
- (4) What does your process do and what is the aim of the process?
- (5) To which other processes is it connected and to what extent?
- (6) What are your responsibilities in the process?
- (7) What are the main problems you are facing in doing your daily work?
- (8) Which functionalities should the software tool have?

The first three questions allowed the researcher to categorise the participant. In order to understand their background and possible context of statements, questions four, five, and six elicited details about the environment that the participant worked in. The final two questions were looking for a broad answer, with follow-up questions generated by the researcher according to how the conversation developed. All questions were formulated as open-ended questions in order to avoid binary (yes/no) answers. In case something was unclear or needed further elucidation, the researcher encouraged the participant to elaborate on the topic. At the end of the interview, the researcher asked the participant to confirm that all information collected was correct. That way, the risk of misinterpreting a response was minimised.

During the data collection the researcher took notes to record the information from the participants. Since the collected information was rather specific, and the content of what was said was more important than the sentence structure, language patterns, or word choices, it was not necessary to provide a word-by-word transcription. In order to preserve the participants' anonymity, each participant was assigned a number which was used throughout the thesis.

After collecting the data they were analysed by the researcher. As the amount of collected data was not very large, it was decided to examine the data manually. A software tool called NVivo, which offered tagging of statements from the transcribed interviews, was initially considered to support the analysis; however, the effort for setting it up in order to create meaningful results was found to be too high compared to just performing the analysis manually. The researcher used an inductive approach in which requirements emerged by interpreting each interview transcript and transforming the participants' statements into requirement themes. That way, it could be ensured that no information from the interviews was lost during the process.

Once the analysis was completed, the elicited requirements were validated with the help of the researcher's manager and with participants who had agreed to support this stage of the process. The requirements were discussed and checked for their consistency and completeness. In order to get a better overview, they were put into relation to each other to identify interdependencies between them. After that, the requirements were prioritised depending on how valuable the respective feature was for the software tool. This was determined by the functionality's contribution to the main purpose of the software tool (perform the task of backup monitoring), by its relevance in the auditing process, and by counting how often a particular requirement was stated by different participants.

After the software tool was developed, it was evaluated in terms of how it met the requirements and how well its overall functionality addresses the research objective. As it was found that the software tool itself does not entirely solve the problem, a generalised framework was developed, enabling the researcher to create a more universally applicable contribution to the body of knowledge.

3.4 CHAPTER SUMMARY

This chapter describes in detail the methodological approach of the research. It introduces the methods used, the Design Science framework, the software development lifecycle, and the requirements engineering process. A pre-study related to this research that was carried out as a small-scale project of this study is also described. The study methodology is discussed in detail in the third section of the chapter, describing the software artefact creation process including data gathering and analysis.

CHAPTER 4 FINDINGS

This chapter presents the first part of the findings of the research, which were derived from interviewing business experts. In the context of the methodology for this study, it is part of the Design and Development phase of the Design Science methodology and represents the first step of the software development lifecycle. In Figure 4-1, this is marked by the brown box.

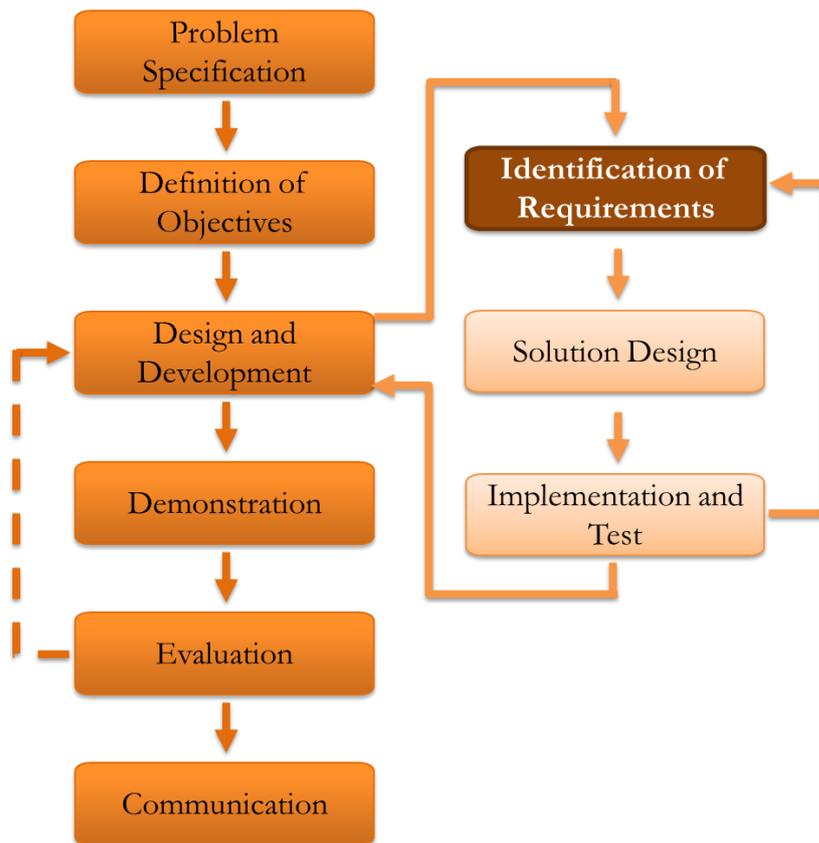


Figure 4-1: Requirements identification in the context of the methodology

This chapter first briefly describes the environment in which the research has taken place and defines necessary terminology. In the second section, the interview findings are presented by giving an overview of the participants and analysing the information that was gathered from them. After that, all found requirements are listed and it is shown how they are related to each other. The last section presents findings that do not directly imply requirements for the software tool, but rather for the circumstances of the development process.

4.1 ENVIRONMENT DESCRIPTION

The software tool that was created in order to address the research objectives was implemented within the SAP environment. The software in which it is embedded is called Global Management Portal, or GMP. The Portal is used by about 1360 SAP employees to manage all SaaS cloud applications that are related or similar to SAP's product *Business byDesign* (ByD). This includes applications such as *Customer Relationship on Demand* (COD), *Travel and Expense Management on Demand* (TEM), *Payroll on Demand* (PAY), or *Financials on Demand* (FIN). In addition, a PaaS solution called *SAP NetWeaver Neo*, or JPaaS (Java Platform as a Service), which allows the user to develop add-ons for the underlying platform of most of SAP's on-premises products, is managed in GMP. Furthermore, SuccessFactors applications are also being transferred into GMP after they were acquired by SAP.

From a technical point of view, GMP serves mainly as a data collection tool. It is connected to other systems that are used to manage hardware and infrastructure in the datacentres, for example the *Service Provider Cockpit* (SPC) which stores deployment information about all ByD systems. In terms of backup, a tool called *NetApp* that creates backups of specified technical entities (e.g., a system, a storage filer, or a storage node) using snapshot technology is employed. It sends back error codes which are then stored and analysed by GMP.

The entities that are important for this study are systems on which a specific SAP cloud application is provided to one or more customers. Each system can have several backups, of which one is usually a snapshot copy of all data stored in the respective system. Systems that are hosted in the same datacentre are grouped together in a so-called technical landscape. Depending on the application that runs on a system, it is also assigned to a so-called usage area.

The tool that was developed during the research reported here was embedded in GMP's reporting section in order to expand the existing functionality in terms of backups. For example, if failed backups are reported (regardless of their origin or current status) this triggers an alerting function that notifies technical support staff to restart the task in case a backup did not finish successfully.

While embedding the solution in GMP was required, using GMP as an implementation environment imposed several constraints. First, the programming language in which GMP is written is Perl, and there are no means of using any other language for developing the

tool and embedding it in GMP. Therefore, the software tool had to be implemented in Perl. A second constraint is that “functionality shall not be duplicated” (F. Reimann [head of development for GMP], personal communication, 25th June, 2014). This means that in case the functionality requested by a research participant already exists in another part of GMP, it should not become part of the tool being developed, at least not for the implementation itself. It can, however, become part of the more general framework which aims to address the research objectives.

Another aspect that has to be taken into consideration is that the source code formatting and the commenting guidelines for GMP have to be followed in order to ensure maintainability by the development team. This also applies to the general layout of the code and the user interface.

4.2 INTERVIEW FINDINGS

This subsection presents the findings of the analysis of the interviews that were held with ten research participants in order to elicit requirements for functionality of the software tool. The section first introduces the participants and then analyses and prioritises their responses. The result of this analysis is a list of requirements that are presented along with a set of evaluation criteria. Afterwards, all requirements are further analysed by putting them into relation with each other. The last part of this section presents some non-technical findings. The complete interview notes can be found in Appendix G as taken by the researcher.

4.2.1 PARTICIPANT SUMMARY

As the researcher had worked at SAP for more than a year, she already knew a few of the potential participants before this study started. Therefore, the researcher herself was not involved in the participant recruitment process; rather a manager who had been working on the topic for several years identified a pool of 37 participants that he considered valuable for and interested in the research. This list was given to an office assistant who sent out email invitations to potential participants (Appendix F). Interested participants could contact the researcher. That way, it was ensured that the participants did not feel under pressure at any time. This procedure was also approved in the ethical approval given by the Auckland University of Technology Ethics Committee (AUTEK) on 24th February 2014 (AUTEK application number 13/315).

The questions asked were related solely to the participants' jobs. Furthermore, throughout the thesis participants are only identified by an identification number that was randomly assigned, their names will never be made public, and they are all being referred to as "they" regardless of gender or number. That way, their privacy is protected.

The ten participants that volunteered to give an interview came from very diverse backgrounds in terms of their job responsibilities within the SAP cloud departments. The applications Business byDesign and ByD-like products, SuccessFactors products, NetWeaver Neo/JPaaS, and HANA Enterprise Cloud were each represented by at least one participant each. Half of the participants were working in the area of ByD, which is probably due to the pre-study already being implemented in that area and, thus, employees being more motivated to talk about a process they are familiar with. Another reason could be that the manager who identified potential participants is more familiar with the area of ByD, since this is his area of expertise. This meant that more information could be collected on ByD-related topics than on the other applications.

The participants also varied in their job roles, ranging from technical (e.g., IT support staff, software developer) to administrative, business-related and managerial positions. Thus, participant level of involvement in the reporting process was uneven. However, this allowed the researcher to gather different opinions on the topic, coming from different viewpoints.

Table 4-1 provides a summary of the participants' data. It includes their job title, the cloud product with which they are working, and their main responsibilities in the area. These three features were used for categorising the participants and were gathered from the responses to the first three questions of each interview.

It happened twice that two people wanted to be interviewed together. Therefore, there are only eight entries in the table, of which the paired participants are marked with a little asterisk. The pairs are treated as one participant as they represent one entity due to similarities in their jobs and the answers they gave; their ideas were developed in collaboration.

Table 4-1: Participant summary

No.	Job title	Cloud products	Main responsibilities
1	Cloud Security Officer	Business byDesign und ByD-like products	Making sure all ByD-related operational processes abide by the Audit-relevant standards
2	Senior Support Engineer	Business byDesign and ByD-like products	Management and operations of VLAB-Landscape, setting up Zabbix monitoring, cloud operations platform services and monitoring, infrastructure (server & storage, network) L2 operations
3	Technical employee	SuccessFactors applications	Ensuring the backup process runs according to Audit standards
4	Software Developer	Business byDesign and ByD-like products	Development of new functionality for GMP in the area of backup
5*	Senior Support Engineers	NetWeaver Cloud, JPaaS	Ensuring audit compliance of JPaaS systems
6	Operations and Expertise – General Management and Admin	Business byDesign and ByD-like products	Creating reports using different tools, administrative tasks
7	IT Business Services Principal Consultant	HANA Enterprise Cloud, SuccessFactors Applications, Ariba, Business by Design	Overall SAP Cloud Compliance Coordination
8*	IT Technology Senior Consultants in Global IT Backup Management	HANA Enterprise Cloud	Ensuring that backup and restore process is running properly

The first participant was employed as Cloud Security Officer for ByD and ByD-like products. They had to ensure that all processes in this area are compliant with the security standards that SAP get certified for during the audits. Their perspective was, therefore, high-level, as they were mainly interested in a reliable reporting.

The second participant worked in the same area as participant 1 as a Senior Support Engineer. Their position was more technical as they were responsible for the operation and monitoring of landscapes and infrastructure. The VLAB (verification lab) is used to test systems before they are set live. Zabbix is a tool which is currently used by SuccessFactors to create and manage their backups, and it is hoped to integrate that tool into GMP soon. L2 operations are part of the second level support team which solves technical issues with systems and is responsible for ensuring error-free operation.

The third employee was a technical employee for SuccessFactors applications. Their area of responsibility was to make sure that the backup process for those systems was running and complied with the standards that were tested in the audits.

The fourth participant was a software developer for GMP in the area of managing and monitoring backup. They were responsible for developing new functionality that was needed for the audit and by technical support staff.

The fifth participant was actually a pair of two Senior Support Engineers who worked in the area of Neo/JPaaS. They were responsible for making the JPaaS systems compliant with the audit standards, which also involved reporting on backups.

The sixth participant was a general administrator in the area of ByD and ByD-like products who created and sent out different reports. They were a frequent user of the existing backup reporting tool which was developed in the pre-study.

The seventh participant was an IT Business Services Principal Consultant who is currently coordinating the implementation of SAP's One Delivery strategy. Therefore, they had to deal with all SAP cloud products in order to integrate their processes.

The eighth participant was again a pair. They were responsible for the backup process in the area of HANA Enterprise Cloud, where they had to ensure that it operated in compliance with the standards certified in their audit.

4.2.2 DATA ANALYSIS

This section analyses the interview responses by each participant and transforms them into requirements by using a technique called *coding*. This includes marking segments of the interviews with labels that categorise and summarise the respective part (Charmaz, 2014, p. 111). Participant statements were interpreted in order to inductively identify a potential requirement by combining emerging themes. The requirement was given a name and an identification number, as well as a working description. During the analysis, the descriptions were amended or adjusted as requirements already identified were re-used.

Since the participants did not only state a list of requirements, but were also encouraged to include information about their work environment, background information could be used to further elaborate on some requirements. During the interviews it was ensured that the requirements were understood correctly by asking the participants to clarify and expound on their statements if necessary. This section summarises the results of the analysis of the interviews. A full list of how relevant interview notes were mapped onto requirements can be found in Appendix H. This document was also used for tracing requirements back to participants when evaluating them. The requirements that were identified during the analysis are referred to by a number enclosed in parentheses, e.g., (5), which matches the list of requirements presented in section 4.2.3. For the reader's reference, a fold-out list of all requirements can be found in Appendix K.

Participant 1 requested “easy, reproducible, trustworthy evidence for the audits”. This was interpreted into three requirements respectively, which are usability (26), historicisation of data in tables and log files (27), and to ensure that data sources can only be manipulated by the script (28). They also needed an “easy way to find the basic population” (16) which describes all systems that fall into the scope of the next audit. This is not directly related to the backup reporting, but is a rather general request; though this functionality already exists in the old reporting, it may need to be made more accessible.

They also requested the ability to filter systems by their technical landscape (1) and display this information in the result tables (29) for easier recognition of error patterns. Another function that was requested is to display the current system status using coloured indicators. This was already implemented in the pre-study; however, more system states may need to be included (9). Furthermore, the participant asked for some means to link failed backups to a list of affected customers (2).

Participant 2 said that it would help them if the reporting could automatically include policies for the backups (17) as they had to check for this manually. Furthermore, they requested a new functionality regarding the storage of the backup. Each backup is stored on a primary filer for three days and then transferred to secondary filer storage for the next 15 days. These transfers do occasionally have problems, so the participant requested that the software tool should also check if all backups were transferred successfully between filers (3), including an alert function for instant response and a reporting option to be presented in the audits. Additionally, they would like to filter the results of the backup reporting by storage filer (30).

Furthermore, they requested more systems to be included in the reporting (4), and “new system types should be added automatically” (32) to avoid possible delays in processing. These requirements were further developed into including a facility to let users configure their own reporting scope and frequency automatically (10), maybe by providing the option to configure sets of checkboxes (20). Another requirement was that the reporting should provide more detailed error messages (12). This would enable the participant to address issues faster as it shortens the time required for finding the root cause of a problem.

In addition to this, a current issue with the migration of systems to another type of database was causing problems, so that the participant requested that the reporting would only take the “best backup” into consideration (5), which means that if there is a backup on one database for the system that was successful, there is no need to check further. The participant also recommended a change to one of the display texts on the screen (31). As a last point, they suggested to implement a functionality reporting over the number of alerts for a given scope (18). Such a report had been requested by the auditors in the previous audit.

Participant 3 provided information from the perspective of SuccessFactors systems. They stated that most systems are currently not managed in GMP, which means that they can not be immediately included in the reporting. For the research, this implies that a portable concept has to be created which can be easily migrated to other management systems (22). Similar to participant 2, they would also like to add new systems (4) and see a benefit in letting users configure their reports on their own (10).

It became evident during the interview with participant 3 that SuccessFactors systems use longer IDs than ByD systems, which meant a different field had to be used to identify a system, namely CCMS_SID instead of SID (23). Furthermore, the “whole database is

backed up” instead of a system-wise backup as for ByD, which means another requirement is scope flexibility (24) in terms of entities.

Participant 4, the developer, was interviewed after participant 2, who had requested functionality that involved interpreting error messages (12), and an alert reporting (18). This meant that participant 4 could be consulted on the topic. According to participant 4, the error messages provided in GMP are not sufficient to fulfil requirement (12). Furthermore, the requested alert reporting function (34) should be included in the same section as the tool implemented by the researcher. In addition to these clarifications, participant 4 raised several points which were combined into two requirements, namely to follow the development guidelines (33) and to produce readable and maintainable code (25) as the implementation will be included in their area of work.

Participant 5 came from the area of NetWeaver Neo (JPaaS). They noted that GMP was actually not used much by them directly at the moment but certainly would be used in future. This feeds into requirement 26 (usability), because new users of the software tool should be able to operate it easily from the beginning. The participant also requested that new systems should be included (4) and that users should be able to configure their own reports (10), because “too many irrelevant systems are collected by the report at the moment” which does not make the reporting very useful for them.

Similar to participant 3 from SuccessFactors, they do not use the three-digit SID but a longer ID which is stored in the field CCMS_SID (supporting requirement (23)), and they also create backups per database so that they need a flexible scope (24).

Another feature that they would find very helpful in their daily job is to have a reporting over non audit-relevant systems (6). This would allow them to check systems that are part of test landscapes. Furthermore, they requested additional system details to be displayed (13) so that they can easily recognise error patterns.

Participant 6 was an active user of the backup reporting and also of other monitoring tools. Therefore, their comments were mainly based on comparisons with those other tools. In order to fulfil their duties in a timely manner, they required good performance (35). They suggested several minor changes: the reports should be sent out automatically (11), and an automatic feedback function should be included so that recipients of the reports could inform a developer about problems with the tool rather than informing them (19). Furthermore, an option to check or uncheck all checkboxes (36) was

seen as helpful. This was found to be related to the requirement for saving a preferred list of marked checkboxes (20).

Participant 7 provided a very high level view on the backup processes of SAP's cloud applications but also some new and interesting insights into the topic from a different perspective due to their involvement in the implementation of the One Delivery strategy. Thus, almost all of their statements implied developing a concept that is portable (22) and could flexibly include new systems (4, 24), as this is what they focused on in their daily job at that moment.

The participant also stated that they “need[ed] a common technical base, which [did] not exist at [that] moment”. However, this is the prerequisite for a properly functioning backup process spanning all cloud applications. Therefore, it is impossible to use software tool implemented in one area (ByD) in other cloud applications without migrating it to other management systems first. This was an important insight for the researcher as it outlined a constraint to the software tool that was being implemented.

Participant 8 began by describing one of those other management tools, namely the SAP in-house system manager (SISM) that is used to manage all HEC systems, which enforced the requirement for concept portability (22). They explained in detail how their tool worked which contributed hugely to a better understanding by the researcher of some global challenges that SAP face.

The insights gained from understanding the workings of their processes could contribute to creating a new management solution that would be able to handle all different cloud applications. Therefore, the information may not be very valuable in terms of software requirements, but would need to be taken into account when creating the general framework.

In addition to what was found from the interviews, there were some more requirements derived by the researcher when taking the perspective of an employee and using the tool she had developed in the pre-study (appendices I). First, there was an error in the code which caused the tool to not update itself when system characteristics were changed. Therefore, a new requirement was formulated that systems shall always be grouped in the area where they belong, and their status shall be checked on a daily basis (7).

From a developer's perspective, another useful feature would be to send error notifications of the script to the researcher or another responsible developer (14) so that bugs can be fixed quickly. Also, the underlying data collection method was not flexible enough, as in

the old tool obsolete systems could not be deleted which meant that more data than necessary were stored. Therefore, another requirement is that system collection criteria can be set to inactive or deleted (8).

Another requirement arose due to the global distribution of SAP's cloud departments, and the respective time zone differences. Since GMP is mainly located in Germany, the automatic data collection script should be scheduled as early as possible in the German time zone (21), so that there is no further delay for employees working ahead. During the operation of the tool it was also found that it would be very helpful if each system type had a responsible employee who can be contacted in case of failures. In the researcher's experience it is vital that the reporting reaches the right people who are able to fix issues with the systems reported as faulty; therefore, as part of the design process, the responsible people for each system type shall be identified (15) in order to send out accurate notifications to them.

4.2.3 SET OF REQUIREMENTS

This section discusses all requirements that were identified in the previous section. Each of them is described briefly and prioritised based on the value which the respective feature adds to the tool. This value is determined by the expected merit for users of the tool in order for them to perform the task of backup monitoring, and by how many of participants 1 to 8 expressed the requirement. The participants were also asked to assess the priority of their requests. Furthermore, a requirement's relevance in the audit process also contributed to its priority rating where applicable. As a result, four levels of priority (high, medium, low, and persistent) were defined and assigned. The term "persistent" describes a requirement that has to be considered throughout the whole development process, including non-functional requirements. Such requirements can not be directly prioritised as low, medium, or high, because they are an integral part of the development process as such. The requirements do not follow a specific order, as they were identified in several rounds of interview analysis. However, they are grouped and put into relation with each other in section 4.2.4.

1 Filter by technical landscape

Adding an option to filter the results of the reporting by the technical landscape in which the systems are located was found to add a high value to most of the participants because the technical landscape usually represents the datacentre in which a system is located.

Including the technical landscape simplifies the creation of location-specific audit evidence which is why this requirement is prioritised as *high*.

2 Show list of affected customers

Due to the impact a failing backup could have on customer operations (i.e. loss of sensitive business data) this feature was found very valuable by all participants. Therefore, it is prioritised as *high*.

3 Filer storage reporting

This requirement describes new functionality which consists of three parts. First, the reporting tool should check if backup files were successfully moved from primary to secondary storage. Second, an alert function shall be implemented to notify if the transfer was not successful. Third, a monitoring function shall be added to account for successful transfers in a given time period. Due to its applicability and usefulness for all SAP cloud solutions, this requirement is prioritised as *high*.

4 Add more/new systems

This requirement was named five times which makes it a *high* priority. It means that new system types shall be added to the reporting. Since it is important to many participants, a good way has to be found to implement this feature.

5 Take only “best backup” into consideration

This means that if there are several suitable backups defined for one system, it should not matter which one of them was completed successfully as long as at least one captured all the system’s data. Due to its high impact (systems are falsely reported as lacking a proper backup) this requirement is prioritised as *high*.

6 Report over not audit-relevant systems

Including this feature would enable all users of the reporting tool to perform test runs on their systems in technical landscapes that are dedicated to trial and, therefore, not part of the audit. Until now, the tool only covers audit-relevant systems, so the addition would be very valuable to the reporting which is why this requirement is prioritised as *high*.

7 Update information on upgraded systems

This requirement arose due to an error in the data collection script which caused systems to be reported in the false category. In case a system gets an upgrade, and, thus, changes the category to which it belongs, this information should be updated in the tool as well in order to avoid inaccurate figures. Due to this huge impact on the calculation of figures for specific categories, this requirement is prioritised as *high*.

8 Take obsolete system categories out of scope

Once a system category goes out of scope, this category should be deleted from the reporting in order to decrease the amount of data that have to be stored and also to avoid confusion amongst users who could be wondering about why “old” systems are still being reported on. For example, this would happen if all productive customer systems of a specific category have been upgraded to other categories. Due to its importance for all users of the tool, this requirement was prioritised as *high*.

9 Display coloured indicator next to system to show its current status

This feature was already partly implemented in the pre-study and in that has proven to be very useful, so that this requirement would mean to keep the existing functionality but increase the number of states that are checked for. This feature was requested by two out of eight participants, which should make it a high priority requirement, but it also already partly exists which means that the value added is rather low, so that its overall priority was set to *medium*.

10 Let users configure automatic reports

Freely configurable reports in terms of system scope, timeframe, and group of recipients was one of the most requested features during the interviews. It shows that users need flexibility in order to address constantly changing audit and customer requirements. The value that would be added to the software tool by including this feature is significant for all users as it would allow them to create reports tailored for their needs in a timely manner. Therefore, this requirement was prioritised as *high*.

11 Send out reports automatically

This feature was requested by the participant who is responsible for sending out the reports weekly. Since it still involves manual work (starting up the tool, making screenshots of the rates of all system categories, and combining them in a document which is sent out), this functionality would be valuable for all those users who are only interested in monitoring the two success conditions defined by the auditors, but not in investigating the issues further. Therefore, this requirement was prioritised as *medium*.

12 Display detailed failure reason

Including this functionality would allow users to quickly find out why an error is occurring and, thus, be in a position to fix it faster than if they had to do a lengthy investigation beforehand. This requirement was prioritised as *medium* because it is a secondary feature; whereas monitoring the actual success rates for the backup process is the primary purpose of the tool.

13 Show more system details

The participant who suggested this feature explained that it would allow them to recognise error patterns more easily so that they could fix the root cause of a problem and not just treat the symptoms. However, just like requirement (12), this is also a secondary feature, as it mainly increases the amount of details shown on the screen. Therefore, it was also categorised as *medium*.

14 Send script error notifications to the developer

In case there is an error in the code, it shall automatically send a notification to the responsible developer so that it can be corrected. This requirement was identified during operations of the tool which was developed in the pre-study. It only provides indirect value to the users of the tool, because only a developer can fix the errors reported by this function. At the same time, the users rely on the tool functioning properly, which is why this requirement was prioritised as *medium*.

15 Find employees who are responsible for each of the applications covered

This is another requirement which was identified during the operations of the tool. It is not so much a feature of the software tool but rather a condition for delivering the right information to the right people, which is an overarching goal when developing software for a lot of users. However, this feature is not vital for the software tool when fulfilling its main purpose of backup reporting. Therefore, this requirement was prioritised as *medium*.

16 Retrieve the basic population

The tool developed in the pre-study already has a functionality that fulfils this requirement. Therefore, it is prioritised as *low* because practically no value would be added. However, the existing functionality should be made more obvious and communicated better.

17 Automatically include policies in the reporting

This requirement refers to the policies that describe how the backup process has to function in order to be compliant with the standards that are certified in the audits. These policies cover a very wide area (not just the backup success rate which is what the tool focuses on) which is why only a few of them are of interest in this context. Furthermore, the policies only change when the audit standard changes, so all in all the expected value for users is considered to be relatively small. Therefore, this requirement was prioritised as *low*.

18 Display total number of alerts for given scope

This requirement means that users could monitor how many backups failed for a given scope, regardless of any further successful runs on the same day that would restore compliance with the audit standards. (A system is compliant if there was at least one successful backup run for it each day.) Again, the value for all users is not very high, because on one hand this number is not important for the main task that the tool fulfils, and on the other hand, there is already functionality available in another reporting tool (which aptly monitors alerts) so that this requirement was prioritised as *low*. This feature was implemented by another GMP developer while the researcher was still interviewing the business experts to gather more requirements. As a result, this requirement has been updated and was subsumed by requirement 34, which requests that this alert reporting shall be integrated into the backup reporting tool.

19 Feedback to be sent to the developer

This requirement was requested by participant 6, who is responsible for sending out the weekly reports and occasionally receives feedback about the functionality of the reporting tool. These suggestions by recipients should instead be sent to a developer, who can actually implement them in the tool. As the value of this feature for all general users of the reporting is relatively small, it was prioritised as *low*.

20 Save sets of checkboxes

This requirement aims at increasing the usability of the reporting tool by providing the users with a feature that allows them to save a preferred set of system categories on which they report (represented by selectable checkboxes on the user interface). Due to its usefulness to all users, but at the same time being only of secondary importance to the main goal of the tool, this requirement was prioritised as *low*.

21 Schedule data collection script early

This requirement was identified by the researcher during the usage of the tool developed in the pre-study. Since SAP are a global company, and cloud departments are distributed around the world, time zone differences play an important role when planning operations. GMP is developed by a German team and their servers are located in Germany. Scheduling the script that collects the data for the reporting at an early time in that time zone ensures that collected data are accurate (given the risk of time misalignments is reduced) and are also available at the earliest possible moment. However, this requirement was not mentioned by any of the participants, which means that it is probably not that important to the end users of the tool, so this requirement was prioritised as *low*.

22 Create a portable concept

According to the number of times it was stated by interviewees, this requirement was one of the most popular ones. This response shows that creating a design that is not bound to the underlying technical infrastructure but is rather universally applicable is not only an objective of the research for this thesis, but also an important topic for the interviewed business experts. Therefore, this requirement was prioritised as *high*.

23 Use CCMS_SID instead of SID

This requirement arose due to technical specifications in GMP. Each system has two IDs, one being the SID and the other one being CCMS_SID (unfortunately, the actual meaning of CCMS could not be gleaned). When the tool was initially developed for ByD systems, both IDs always had the same content, so that SID was chosen for more easily readable code. However, with the addition of new cloud products to GMP, the CCMS_SID has become more important since it was defined as unique value (unlike SID). Therefore, this requirement should be *persistently* considered and all code that only includes SID shall be altered to make use of CCMS_SID.

24 Offer scope flexibility

By implementing this requirement, users would be able to define a scope for their reports that is not restricted to a specific format. In the tool developed in the pre-study, systems were mainly categorised by two characteristics, which were ByD version and ZH code. The new software tool shall be able to handle a more general format by allowing the users to define queries in order to gather data about systems that belong to a specific category. Due to its importance for the versatility of the tool, this requirement was prioritised as *high*.

25 Write readable and maintainable code

This requirement makes the code that is developed compliant with the programming guidelines of GMP, so that other developers could read and maintain it once the implementation is finished. As this has to be considered throughout the whole implementation, it is prioritised as *persistent*.

26 Usability

The tool needs to be easily usable by new and existing users. This requirement was found twice and has to be considered throughout the whole development process. Therefore, it is prioritised as *persistent*.

27 Historicismation of data

This goal can be achieved by permanently storing data in tables and making sure that they are not deleted. This requirement is also *persistent* as all underlying database tables have to be designed in this way.

28 Data source manipulation only by automatic scripts

No data shall be entered manually by employees. This requirement is also *persistent* and can be evaluated by checking if there is any database manipulation access for users.

29 Add landscape as a column to the result table

This feature was requested in order to show in which technical landscape a system is located. Since technical landscapes usually represent datacentres or parts of them, adding this information to the displayed result would benefit the users who want to identify and locate the root cause of an error. Since this is only relevant to a part of all users, this requirement was prioritised as *medium*.

30 Filter per filer storage

Implementing this requirement would allow users to check specifically for backups which are either in primary or in secondary (archive) storage. Since this feature could be potentially valuable to users in order to identify errors more easily when they appear to be common in a certain storage entity, this requirement was prioritised as *medium*.

31 Change display text

Participant 2 requested that the text which is displayed to show the results shall be altered in order to make it clearer what the actual success criteria is and what is measured. In order to increase its merit for the users, the researcher expanded this requirement to revise all labels that are visible on the screen. The value that is added to the software tool by implementing this feature is, however, rather small, so that this requirement was prioritised as *low*.

32 Automatically add new ByD versions

This requirement was found during the operations of the tool developed in the pre-study. It frequently occurred that a new version of the ByD cloud application was available and, consequently, had to be taken account of in the backup reporting, but this only happened after a delay caused by the manual process steps involved. An automated solution for adding new system versions would be valuable for some users, so that this requirement was prioritised as *medium*.

33 Follow development guidelines

As requested by participant 4, who is a developer within GMP, the researcher should follow GMP development guidelines so that her code can be maintained by other GMP developers. Since this requirement has to be considered throughout the whole implementation of the software tool, it is a *persistent* requirement.

34 Include alert reporting

Participant 2 requested a functionality which would report over alerts (requirement 12). This report had been implemented by another GMP developer already, so that they then asked to include it in the software tool. As this functionality would only benefit some users, and it was already implemented, creating only little value for the backup reporting tool, it was prioritised as *low*.

35 Good performance

Although this requirement was only stated by participant 6, it is very important for the success of the developed software tool. Most employees using the software tool have a busy schedule, so that a slowly responding tool is not an option. Therefore, this requirement has to be considered *persistently*.

36 Mark/unmark all checkboxes

This feature was requested so that an overview of all system states can be gained quickly. However, since this was only requested by participant 6, and since the value of this functionality to fulfil the objectives of the software tool is rather small, this requirement was prioritised as *low*.

4.2.4 INTERDEPENDENCIES BETWEEN REQUIREMENTS

After all requirements were identified and their descriptions finalised, they were further analysed in order to reveal possible interdependencies like prerequisites, constraints, complementarity, coherence, and others. For this, each requirement was written on a post-it note and placed on a whiteboard which could then serve as a surface to draw arrows and frames. This method was developed during the requirements analysis process as it was expected to aid a further examination of interconnections between requirements. Two results of this approach are shown in Figure 4-2 and Figure 4-3, each focusing on a different aspect of the analysis.

The colours of the post-it notes indicate the priority of the requirement: Pink stands for high priority, orange for medium priority, yellow for low priority, and blue for a

“persistent” requirement. As the text on the notes is not readable at this resolution, the requirement’s number was superimposed on the photo.

Figure 4-2 demonstrates how the requirements are connected in terms of functionality. The meaning of red arrows is to show prerequisites, green arrows mean examples or sub-categories, and blue arrows show that two requirements are related by some other means and should be considered together. A group of notes framed in a box indicates requirements that belong together as they have a common purpose within the tool.

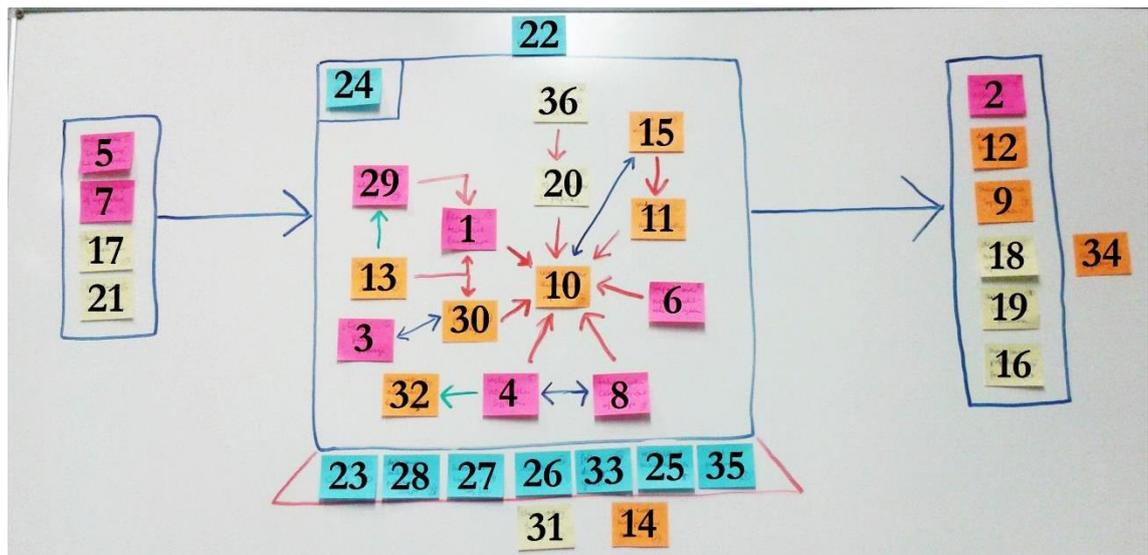


Figure 4-2: Requirement connections in terms of functionality

On the left side, there are four requirements (5, 7, 17, and 21²) which are all related to the underlying logic of the script that runs in the backend of the software tool. These requirements all ensure some sort of input for the tool in order to make it work or fill it with data. The row of requirements in the middle is made up of a set of persistent requirements (23, 28, 27, 26, 33, 25, and 35) that ensure the inner workings of the software tool. By their nature as persistent requirements, they need to be considered continuously throughout the development process, which is why they form the foundation for the tool development. Requirements 14 and 31 can be found below them, as they are a part of the actual technical implementation that also influences the workings of the tool.

At the top of the big central block there is the overarching factor of creating a portable concept (22). Similar to the other persistent requirements, this one influences the whole development; however, its effects are more relevant for using the tool than for its development. Requirement 24 (offering scope flexibility) was placed in the upper left

² A numbered list of all requirements is included as a foldout in Appendix K.

corner of the central block because it is part of the inner workings of the tool, but, at the same time, influences major design aspects.

In the middle of the big block, requirement 10 stands out due to the many red arrows pointing towards it. This requirement was initially prioritised as medium (as indicated by its orange colour), but this analysis showed that it should be of high priority. Furthermore, requirement 29 was initially prioritised as high, but in this analysis it was discovered that it is actually a sub-requirement of requirement 13, as indicated by the green arrow. Both notes are a prerequisite for requirement 1, because the user can only be given the option to filter by technical landscape if there are more system details included (i.e., technical landscape) and if this is also displayed as a column in the table.

Requirement 30 is connected to requirement 3 by a blue double arrow as their functionality is highly dependent upon each other. Another such connection was found between requirements 4 and 8, which both describe opposite functionality that addresses the same goal of altering the scope of the software tool. Furthermore, requirement 32 is an example or a sub-category of requirement 4 as indicated by the green arrow.

The right side of the diagram focuses on functionality that is related to displaying data to the user or interacting with them. This includes requirements 2, 12, 9, 18, 19, and 16. Requirement 34 was placed next to requirement 18 as they both address the alert reporting functionality.

The second diagram is shown in Figure 4-3 and was found to be very helpful for identifying relationships between requirements. This diagram focuses on the interaction between different components of the software tool. Requirements were grouped together based on how similar they were: requirements framed by a blue box are part of the same set of functionality, and should, therefore, be implemented together. A green box denotes a core feature of the backup reporting tool. Arrows between boxes and post-it notes imply a cause-effect relationship, whereas lines show that requirements or groups are of the same kind of functionality.

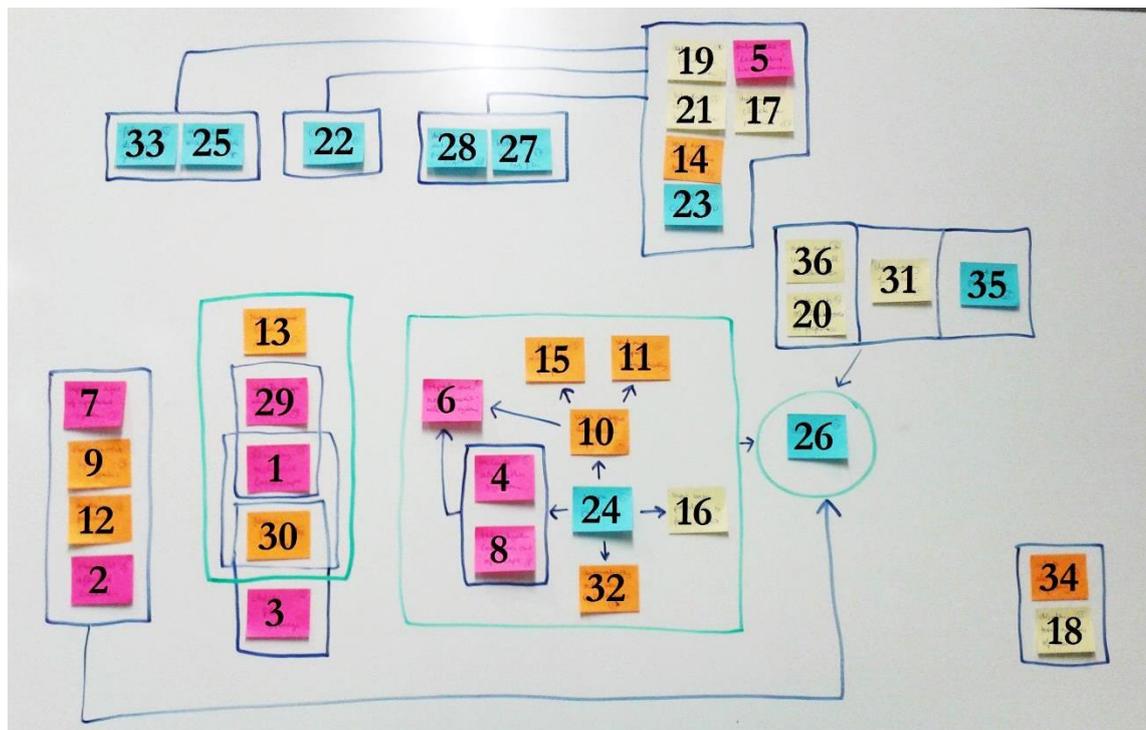


Figure 4-3: Requirement connections in terms of software component interaction

The top part of the diagram summarises all development-related requirements, grouped by similarity. On the left side, there are requirements 25 and 33, which are both related to coding style. Next to them is requirement 22 as creating a portable design is fundamental to the development of the software tool. The right hand group consists of requirements 27 and 28 which both ensure data integrity and reliability. These three aspects are then connected to a block containing more general features that will become part of the software code, namely requirements 19, 5, 21, 17, 14, and 23. All these requirements are directly related to the way the software is developed.

The lower part of the diagram in Figure 4-3 depicts an analysis of features in terms of their relation to the usability of the software tool (26), which is shown by the green circle. Above it is a box with three columns, which contain certain (minor) aspects that are expected to make the users' experience with the tool more enjoyable. Requirements 36 and 20 are grouped together because they are both related to checkboxes.

On the left side of the diagram there is a group of four requirements (7, 9, 12, and 2) pointing towards usability (26). These four requirements are related to features that will be displayed on the UI, which means that they directly influence the usability of the tool. The green block right next to these requirements (13, 29, 1, and 30) is also related to what is displayed on the screen; however, it does not inform usability, which is why the box is not connected to 26. (In hindsight, a dashed arrow showing this relationship would have been most appropriate for this block.)

A set of five requirements (13, 29, 1, 30, and 3) was identified to be closely interrelated. The first four are framed by a green box because they are related to showing additional information on the screen. The bottom four requirements are cascading in terms of how they are connected: a relationship was found between 29 and 1, as they are both dealing with the technical landscape; requirements 1 and 30 both describe filtering functionality, and 30 and 3 both deal with the primary and secondary storage of backups. During the implementation, these three interconnected pairs will have to be considered in their togetherness in order to avoid collisions between different functionalities.

The green square frame in the centre of Figure 4-3 groups all requirements that are related to creating scope flexibility (24). Requirements 4 and 8 had already been discovered as belonging together in the previous diagram, which is why they are now grouped together. Similar to the first diagram, requirement 10 informs other requirements. For example, it points towards requirement 6, as this is an indicator of how flexible the reporting is. Requirement 10 also points to requirements 15 and 11, as it is the prerequisite for the functionality they describe.

In the bottom right of the diagram, fairly disconnected from everything else, sits the set of functionality describing the backup alert reporting. It was found that this functionality was not actually related to the main purpose of the tool, so it could not be connected to any other part of the diagram. This finding caused the priority of requirement 34 to be changed from medium to low, as the benefits of implementing this feature were seen as rather low when taking the overall picture into account.

The lessons learned from this exercise were that certain requirements had to be considered together by the researcher in order to really understand their meaning and scope, and that certain requirements were less related to the main purpose of the tool than others (for example, the filer storage monitoring and the alert reporting). This led to requirements being treated differently during the implementation. Providentially, no contradicting requirements were found so that no issues between them had to be resolved.

4.2.5 OTHER FINDINGS

During the analysis of the interviews, especially with participants 2 (Senior Support Engineer for ByD and ByD-like products), 3 (Technical employee for SuccessFactors applications), 7 (coordinator of compliance for HEC, SuccessFactors, Ariba, and ByD), and 8 (consultant in global IT backup management for HEC), it could be noticed that there are huge operational differences between certain departments working with SAP's cloud

applications, especially when comparing ByD, SuccessFactors, and HEC. All three departments are still certified in separate audits, although parts of ByD and SuccessFactors have been partially combined. Compared with ByD, the processes within the SuccessFactors department are less automated, and the whole compliance process is less “mature” in the sense that some procedures still require much human involvement to prevent, detect, and resolve issues. For example, evidence for auditors is provided by “search[ing] for a customer name to find the associated databases and then check[ing] the backups” (participant 3, personal communication, 5th May 2014). On the other hand, the processes within the HEC department are almost all fully automated, with software tools alerting responsible employees in case of failures, and there is a general awareness in the department in terms of audit compliance. This is due to the employees’ many years of experience in the field of security, which was gained by being assigned to similar tasks in their previous roles.

Differences to this extent were not expected beforehand, which is why the initial thesis idea of a universally applicable software tool was reconsidered. The objective of the study changed to creating a portable framework that will then have to be implemented in the several technical infrastructures that host the respective cloud products until a shared technical base would be put in place. As long as the architecture of the management systems for the different cloud applications varies so widely, no standardised software tool can be implemented. Therefore, it was decided that the concept of the backup reporting would only be implemented in GMP, and, therefore, restricted to uses by the ByD department, as most participants work in that area. However, since a consolidation of processes can be expected, the requirements from those participants not working in the ByD area are also considered.

4.3 CHAPTER SUMMARY

This chapter described the first outcome from this study, namely the findings from the interviews. It first described the environment in which the software tool was to be developed. This was followed by an overview of the participants and an analysis of what they said in the interviews. Out of that, a list of requirements for the software tool was developed. These requirements were then analysed further to identify interdependencies. The last section presented findings that were not related to requirements, but had an impact on the scope of the software, which led to a change to the research objective.

CHAPTER 5 SOFTWARE DESIGN

This chapter describes the design of the tool that was implemented to address the research question and to satisfy the requirements discussed in Chapter 4. In the context of the research methodology presented in Chapter 3, this corresponds to the second and third steps of the software development lifecycle as depicted by the brown boxes in Figure 5-1.

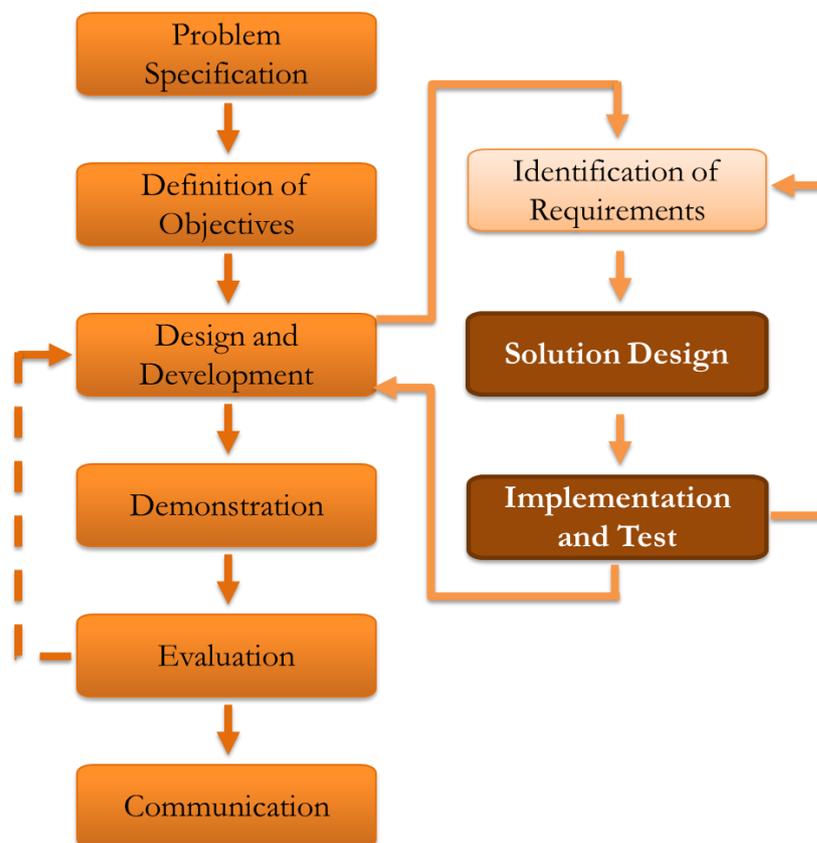


Figure 5-1: Software design in the context of the methodology

First, the chapter identifies constraints that were imposed on the development process and out of which a set of design considerations was developed. It then gives an overview of the functioning of the software tool, which is followed by an explanation of the backend workings, such as the underlying database schema and the data collection procedure. After that, screenshots of the software interface illustrate the frontend design.

5.1 CONSTRAINTS

Due to the software tool being developed in a specific technical environment as described in section 4.1, there are some requirements that could not be implemented. For example, requirement 2 (showing the list of customers affected by a failed backup) can not be

implemented as not all systems in GMP have a list of customers connected to them. The investigation showed that only ByD systems had a properly defined list of customers attached to them, as the system data for most other applications come from other sources and are not entirely replicated in GMP.

Another requirement that can not be implemented due to technical restrictions is requirement 12 (display detailed error messages for backup failures). After analysing the available data, it was concluded that gathering detailed error messages was not possible at the moment. The error codes which were delivered by the database were not detailed enough to allow a thorough analysis. Therefore, the root cause of the problem could not be identified without human involvement.

In addition, it was found that requirement 17 (automatically include policies in the software tool) could not be implemented as it required the analysis of natural language, in which the policies are formulated. Therefore, developing this feature would go beyond the scope of this study. Requirement 32 (automatically add new ByD versions to the scope) falls into the same category as this would involve an automatic check and analysis of all added systems to see if a new system version was added. After cross-checking with the researcher's manager it was decided that the addition of new system versions would be the responsibility of the employees who set up those systems (H. Cevajka, personal communication, 2nd August 2014).

Furthermore, two functionalities that were not included were requirements 3 and 30 (both pertaining to monitoring the transition of backups to secondary storage) and requirements 18 and 34 (both related to the reporting over backup alerts). Requirements 3 and 30 imply developing a new set of functionality that does not directly contribute to monitoring backup success rate, which is why their implementation was postponed. Requirements 18 and 34 only have a small contribution to the purpose of the monitoring tool, as they look at the issues from a different angle. From the auditors' point of view, a backup is seen as failed if all attempts to create a copy of the data on a particular day failed; therefore, they ignore failed attempts that are followed by successful ones, whereas requirements 18 and 34 take every backup alert into consideration regardless of future outcomes. Furthermore, the functionality requested in requirement 18 was already implemented by another GMP developer, who also fulfilled requirement 34 by including his report in the general backup reporting section of GMP.

5.2 DESIGN CONSIDERATIONS

During the pre-study, it was found that following the traditional approach of the three-layer architecture with data, logic, and presentation layers is sufficient for the complexity of the problem domain. In the data layer, information about systems and their backups has to be stored in the most economical yet accessible way. In the implemented software tool, this was achieved by storing information about active systems using their go-live and go-offline dates in order to save data storage; yet it was found beneficial in terms of computational complexity to create an extra table for storing the number of scheduled and failed systems per day instead of computing these figures during the user interaction. The logic layer, represented in the software tool by the data collection script, provides the means to analyse and pre-process data for optimised storage. In the presentation layer, the display functions of the user interface allow the user to interact with the backup data. For this, it was found that interviewing business experts on the topic and asking them about their expectations of the reporting tool produced some interesting requirements that add value to the tool.

Managing a vast number of systems requires optimising data storage and providing an expandable solution. The first factor is addressed by storing only go-live and go-offline dates of systems instead of daily creating a new entry containing full details. (Such details can be retrieved on user request if necessary.) The second factor is considered in the general design of the software tool, which is centred around the idea of freely configurable queries. When new systems are added, users can diversify their queries in order to re-group the systems according to their changed needs.

The utilisation of the freely configurable queries also addresses the issues related to heterogeneity. As cloud computing becomes more and more important in the world of business software, the number of different cloud applications that are available will increase rapidly, calling for management solutions that are easily adaptable to changing circumstances. Giving the users the ability to freely configure a group of systems they would like to monitor while using any criteria that is available in the development environment only leaves the prerequisite of a capable technical base to be dealt with.

The two criteria that are important for SAP in their audits are the percentage of successful backups and the amount of systems that did not have a backup for three or more days. For other companies, it may be necessary to use different criteria for monitoring than those SAP use. Depending on what is specified in the company policies and in customer contracts, the company has to decide what criteria they need and how these are calculated. Another important factor that has to be considered is the interval at which backups are

run. If they are run daily, like in the SAP environment, it is sufficient to run the data collection script daily. However, if the company policies require backups to be run more frequently, the interval of the automatic script also has to be changed.

When programming the data collection script, a way of handling potential errors is imperative to ensure the integrity of the data. The prerequisite for this is that the information that was not captured is still somehow available in the development environment, which was found to be a problem in GMP. The risk of data loss caused by the data collection script not running can be minimised if the issue is detected early and the responsible person is informed.

Regarding the automatic sending of reports (11) it has to be noted that this feature has not been implemented yet due to the limited timeframe of the study. It has, however, been considered in the design of the software, so that this functionality can be included at a later stage.

5.3 SOFTWARE OVERVIEW

The software tool consists of two parts that handle data processing and presentation respectively. The first part is an automated script which runs daily in the background to collect data used for calculating the backup success rates. Every day, data are gathered from GMP. The script analyses and structures those data and stores them in the appropriate tables. More details on how exactly data are processed are provided in section 5.2.2.

The second part is the user interface which contains all functionality to analyse the collected data. It is started when a user requests information from the software tool. The data which were stored in the tables are shown to the user depending on the functionality they select. A detailed description of the user interface with screenshots can be found in section 5.2.3.

In order to work together, the two parts of the software tool have to use a common database schema, which is described in the following section.

5.3.1 DATABASE SCHEMA

The database schema that was used in the pre-study assumed that each system type can be identified as a combination of ByD version and ZH code. Hence, a lot of empty entries and unnecessary combinations were created, filling up the database with empty rows, therefore, wasting memory and slowing down processing. In order to prevent this from

happening and to provide the greatest possible flexibility to the user (as per requirement 24), a more suitable way of storing system types had to be found.

Since interview participants requested more flexibility with the tool reporting on system categories, it was decided that using a search query to identify required systems would fulfil this requirements best. The query could be extracted from GMP's search function, which allows users to configure a detailed search string that is used to identify a set of systems meeting certain criteria. For the backup reporting, this means that users can build a query that collects data on exactly those systems which they have to report on in the audits. Once the query is entered into the reporting tool, data will be collected according to the users' specifications.

The central table of the schema is the `QUERIES` table as shown in Figure 5-2. It facilitates creating a software-wide dependency on those queries. Each entry has an *ID* to identify each query uniquely, a field to store the actual *query* in text format (as its length is varies depending on the level of detail the users specify), and a *name* that will be displayed on the user interface (UI). Furthermore, an *owner* who requested the query has to be included to provide traceability. This field uses the employee's ID which is unique in SAP.

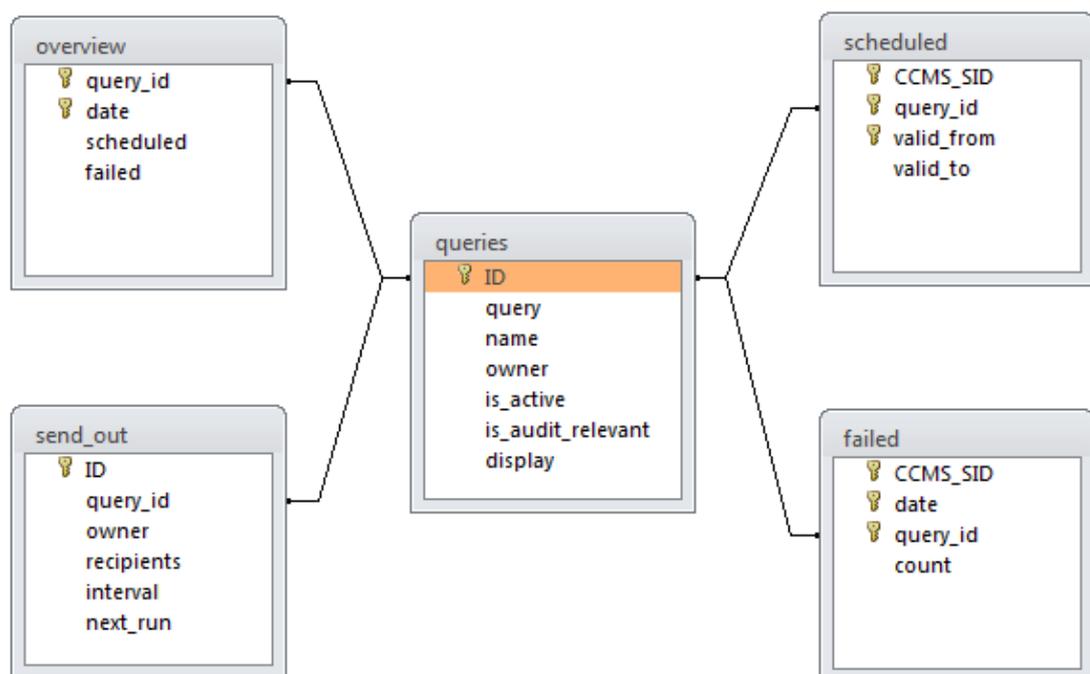


Figure 5-2: Database schema

In addition, each query has three Boolean values which determine if a query is actively collected at the moment (a `false` value means the query is obsolete), if it is audit relevant (in order to fulfil requirement 6 and to group systems on the screen), and if the entry will still be displayed as being selectable on the UI. The difference between `is_active` and `display`

is that queries that are obsolete (i.e., not executed anymore) can still be important for audits which are usually only held every six months. Such a query would be set to *is_active* = false but *display* = true. Using the *display* field also upholds data integrity, as the query can simply be set to invisible instead of being deleted from the table.

The *query_id* is used as a foreign key in all other four tables, because it is the central component of the database design. For example, the OVERVIEW table on the top left stores the number of scheduled systems and failed backups per day per query as an overview. The primary key for this table combines the *query_id* as foreign key from the query table and the *date* in the format YYYYMMDDhhmmss, which is the standard format for dates used in GMP's database. As backups run only once a day so data are only collected once a day, the last part hhmmss is not necessary. Therefore, the date format used within the software tool is YYYYMMDD000000.

Further columns in the OVERVIEW table are *scheduled* and *failed*, which use integer numbers to count how many systems were found on a specific day for the specific query, and how many of those systems did not have a successful backup, respectively. This table was created in order to make the calculation of the backup success rate easier and faster: When requesting the results for certain queries over a specific time period, all rows meeting these criteria can be found and their values for *scheduled* and *failed* are aggregated. Without the OVERVIEW table, both the SCHEDULED and FAILED tables would have to be accessed in order to retrieve these numbers.

The SCHEDULED table on the upper right side stores all system IDs that were collected as part of a query. (Section 5.2.2 explains into detail how this table is populated.) Its columns are: *CCMS_SID*, which is a string of up to 10 alphanumerical characters, *query_id* as foreign key from the query table, and two dates, *valid_from* and *valid_to*, which show when a system went active, and inactive, respectively, as a part of a specific query. A system is seen as active if it is collected when searching with the specific query. As most systems host productive customer systems, and, thus, stay active over a long period of time, using a start and end date was the most economical way to store this information, as compared to creating an entry every day for each active system. However, it is possible that systems become inactive for a few days for maintenance or other reasons, and afterwards become active again for the same query. This would mean that there would be multiple entries for a primary key that would only consist of *CCMS_SID* and *query_id*. Therefore, *valid_from* is needed as part of the primary key in order to uniquely identify an entry.

If a system did not have a successful backup on a specific day, an entry is created in the FAILED table. This table's key consists of three fields, namely *CCMS_SID*, the *date*, and the *query_id*. Since systems can be part of multiple queries, the *query_id* is needed to uniquely identify an entry. This means that, for a system belonging to three queries, three entries would be created if its backup failed. This is needed to ensure a proper calculation of the failure rates when only one affected category is selected. For each entry, it is also important to count how many times in a row a system has failed, which is stored in the *count* field. This information is later used to determine how many systems had a backup failure for three or more days in a row, which is an audit criterion and also an indication for how urgently technical employees have to take action.

The last table in the schema is the SEND_OUT table on the bottom left, which was created to fulfil requirement 11 that asks for reports to be sent out automatically. Unlike the other tables, this one does not use the *query_id* as part of its primary key, but has its own *ID* to identify entries, since all other parameters are can be combined randomly and do not need to fulfil the constraint of being unique. Other columns in this table are the *owner* (which is again represented by an employee ID), and a list of *recipients'* email addresses in text format. In order to specify when the information on the queries has to be sent, the table uses two fields called *interval* and *next_run*. The *interval* field determines how frequently the emails shall be sent, for example daily, weekly, or monthly. Depending on the requirements of the users, this field will contain a word that describes the interval best, such as “weekly”. *Next_run* stores the date on which the next report of this kind has to be sent out, again using the format YYYYMMDD000000. This value is updated each time an automated email was sent by adding the appropriate interval to it.

5.3.2 BACKGROUND DATA COLLECTION

This section explains how the tables in the database schema are filled by the script which is automatically run in the background every day. A flowchart diagram of the script is shown in Figure 5-3. The source code of the script can be found in Appendix J.

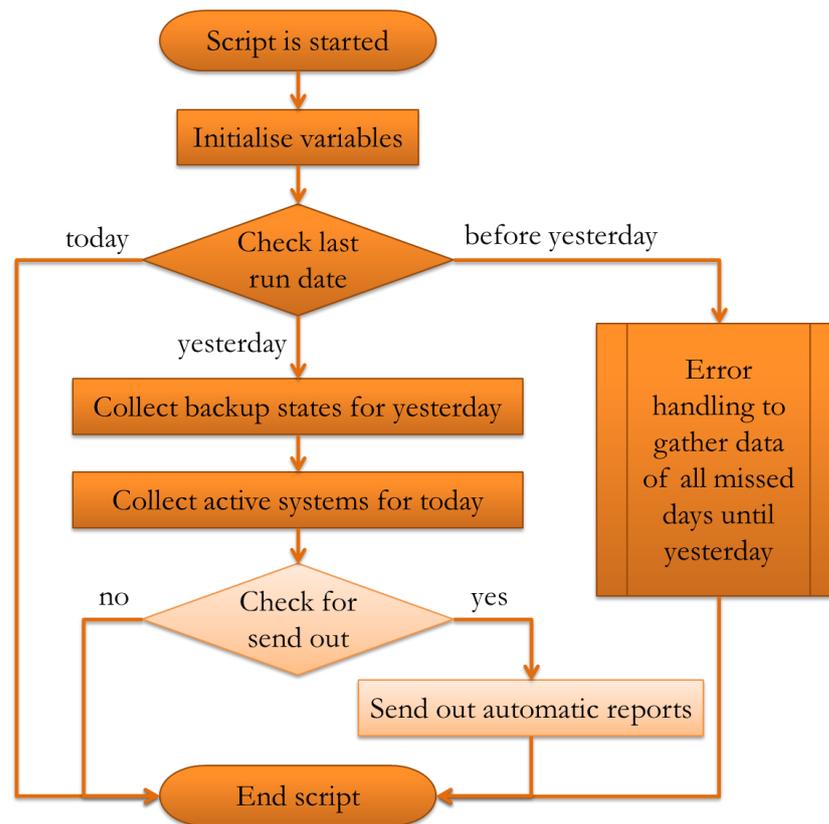


Figure 5-3: Flowchart of the data collection script

The script is automatically started by GMP’s server. It then initialises the necessary global variables, such as the current date, and loads all required Perl modules. Then it checks when the script was last run. In the unlikely event that the script was not run on a particular day, data for that day have to be collected when the script is run next. The last run date of the script is stored in a globally used customising table of GMP. If the script is accidentally run a second time in a day, it ends immediately, as all data for that day have already been collected. In order to avoid the script exiting without having finished its task and then not being able to be restarted, the last run date is only set to the current date if the script was executed successfully until its very end.

The expected outcome of this initial check for the last run date of the script is that the script has run the previous day. In this case, the “normal” run first collects the latest backup states for all systems that were active yesterday, and then gathers all systems that are active today. These two steps each happen in a loop that sequentially processes each query. In order to get the states for the backups, the script selects all systems that were active for that specific query on the previous day. By checking the previous day, it can be ensured that all backups have had the opportunity to finish on that day. The script then loops at this list of systems and checks their last backup status for the previous day. Checking only the last backup status acknowledges that unsuccessful backup attempts are restarted by technical support employees until they run successfully. Thus, selecting the

very last backup run, the script automatically ignores all possible previous unsuccessful runs (which do not count towards the backup success rate as per the audit definition). After all backup failures are collected they are counted and added to the OVERVIEW table by updating the appropriate row.

As a second step, the script updates the systems that are active for the specific query on that specific day. This step should follow the collection of backup states because that way all systems that needed a backup on the previous day will still have the *valid_to* date in the SCHEDULED table set to that day, so their backups can be checked more easily before this information is overwritten. The second step, however, would overwrite this value. When the script has extracted the list of active systems for the query from GMP, it sequentially processes them by checking if there already exists an entry for the respective system in which it is collected as part of the same query and its *valid_to* date is the previous day. If this is the case, the script simply updates *valid_to* to the current date; otherwise a new entry is created that sets *valid_from* and *valid_to* to the current date. By using the *query_id* as part of the primary key of the SCHEDULED table, it can happen that systems have two entries in this table, as they can be part of two queries. This ensures that all conditions are taken into consideration, and system updates are acknowledged as requested in requirement 7. At the end of this step, the number of active systems is computed and written to a new row in the OVERVIEW table.

As the last step of a normal run, the script checks if any reports have to be sent out that day by selecting all entries from SEND_OUT in which *next_run* is the current date. The script calculates how many days each report has to cover by looking at the *interval*. Then, it computes the two audit criteria, which are backup success rate and the number of systems that have not had a backup for three or more days, and sends them out in an email to all people specified as *recipients*. Finally, it updates *next_run* by adding the *interval*. The boxes showing this functionality in Figure 5-3 are of a lighter colour because this feature has not been implemented yet due to the limited timeframe. However, it is an important component of the reporting, which is why it was included in the planning of the script.

The **error handling** is more complex than the normal run, which is why it was included as a sub-programme in the main script. The flowchart for the error handling is shown in Figure 5-4. It includes the same features as the normal run; however, it has to take into consideration all those days on which the script did not run. Therefore, it first has to check which systems were active during the skipped time period, and then has to sequentially process each missed day for collecting backup states and sending out the automatic reports.

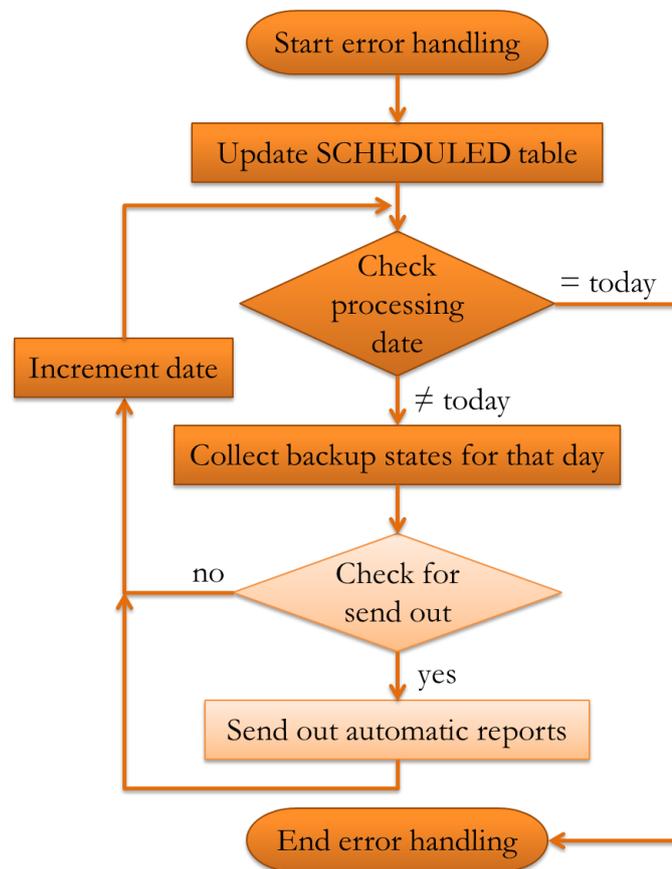


Figure 5-4: Flowchart of the error handling

In order to update the SCHEDULED table, the script has to check which systems became active, stayed active, or went inactive during the skipped time period. For this, it gathers all systems that are active for the specific query on the current day and also all those systems which were active for that query on the last day the script ran (which can easily be retrieved by comparing *valid_to* with the current date).

All systems that were active on the last run date but are not active on the current date must have been set to inactive during the missed time period. In order to find out the last day of its activity, the system's change log files are searched, and it is assumed that the very last change in those log files set the system inactive. This simplification was used because there are generally not many changes made to systems, and because it would be very costly to find out if the specific change affected the system's activity within the query, as a query can become very complex. This principle was also employed conversely, for systems that were not active on the last run date of the script but are active on the current date. Again, it was assumed that the last change to the system set it active.

If a system is found on both lists, it is presumed that it stayed active during the whole time. This is again a simplification; however, it would be very complex and costly to find out if any changes made during that time period actually affected the system's status within the

query. As the error handling is anticipated to not run too often and not cover long time periods, the resulting error in the figures is expected to be negligible.

The next step in the error handling is based on the sequential processing of all days that were skipped. For each day, it is checked which systems were active for which queries, and then their backup states for that day are retrieved. As in the normal run, only the very last backup is taken into consideration, as it shows the most recent state of the system on that specific day. Both the number of scheduled systems and the number of backup failures are written to the OVERVIEW table as base for the calculation of the backup success rate. All failed backups are added to the FAILED table using the date and query which are currently being processed.

After all data for the specific day were collected, the script checks if it has to send out an automatic report by retrieving all entries from SEND OUT in which *next_run* is the currently processed date. Like the collection of backup states, this procedure follows the one used in the normal run. Again, the respective boxes in Figure 5-4 are of a lighter colour as this feature has not been implemented yet.

At the end of each day that is processed, the current processing date is incremented so that the next day can be checked by the error handling script. The truncation criterion for this date loop is set to the day after the run date of the script, so that the very last run of the loop will collect data for the current day and, thus, provide a consistent end state for the collected information.

5.3.3 USER INTERFACE

After all data have been collected as described in the previous section, the users can access and analyse them by using the user interface (UI). The software tool that had been developed in the pre-study was already used by many users at SAP; therefore, it was decided that the layout was already established and should be kept in order to minimise the changes to which the users had to adapt. Also, the interviewed participants did not suggest changes to the UI which means that its current status is sufficient for their needs. The layout utilises the widgets and colours that are generally available in GMP, as it was requested by participant 4, in order to fulfil requirement 33 and keep the appearance of GMP consistent for the users.

The usage of the software tool is simple as the user only needs to take three steps before they get the required report. First, the user selects the date range over which they want to

generate the report. This can be done with the help of a little popup calendar as shown in the orange frame in Figure 5-5. In order to increase the usability, the selectable date range was limited so that only timeframes for which data were already collected could be selected. The default dates are set to cover the previous week, as this is the most frequently requested report.

The screenshot displays the 'Backup Reports' section of the Global Management Portal (GMT). At the top, the portal name and version (v5.10.0) are shown. The main area features a 'Report name' dropdown menu currently set to '#4 Automated Backup Reports'. Below this, there are two date input fields: 'Calculate report from' (2014-10-01) and 'to' (2014-11-09), with a 'Generate Report' button to the right. A date picker calendar for November 2014 is open, showing the date '09' selected. The calendar is highlighted with an orange border. Below the calendar, a list of 'Audit-relevant systems' is displayed, each with a checkbox. The first two items are checked: 'ByD 4.0 with ZH001 (ID 6)' and 'ByD 4.0 with ZH006 (ID 7)'. Other systems include various ByD versions (1305, 1308, 1311, 1402, 1405, 1408) and CRM/TEM/Financials/Payroll systems. At the bottom, there is a section for 'Non-Audit-relevant systems' which is currently collapsed.

Figure 5-5: Data selection screen of the software tool

In the second step, the user can select the queries required for their report. The queries are grouped by the table field *is_audit_relevant* to make it easier for users to find the query they need. Since there existed many combinations of ByD version and ZH code in the tool that was developed in the pre-study, and this existing data had to be transformed to fit the query-based database schema, there are usually two entries for each ByD version which include the main two ZH codes: ZH001 and ZH006. (For example, the two selected

queries 6 and 7 were created by combining ByD version 4.0 with ZH001 and ZH006.) The data transformation had to be done before the software tool went live, so that the existing database schema could be replaced by the one presented in section 5.2.1.

Once the user has selected their required queries and timeframe, the third step is to click the “Generate report” button. The result screen is shown in Figure 5-6. From top to bottom on the left hand side, the screen shows which report type is currently selected (as the software is embedded in other backup-related reporting tools), what data were entered in the selection pane; beneath the horizontal line that partitions the screen are the results for the user’s input, followed by a line of buttons that lead to the respective detail screens.

The results are shown in two boxes that are coloured green when the respective audit criterion is fulfilled, and red otherwise. Due to the way this information is stored in the OVERVIEW and FAILED tables, the values for both boxes can be calculated easily by aggregation and counting. The value in the first box is determined by the formula

$$\frac{\textit{number of active systems with successful backup}}{\textit{number of active systems}} * 100\%$$

which can also be written as

$$\frac{\textit{number active systems} - \textit{number of systems with failed backups}}{\textit{number of active systems}} * 100\%$$

In order to get the number of active systems, the software tool sums up all entries of the *scheduled* column in the OVERVIEW table in which the *query_id* is among the ones that were selected by the user and the *date* is within the specified date range. The same is done for the number of systems with failed backups by using the *failed* column of the OVERVIEW table. As specified in the contracts that SAP have with their cloud customers, this result is acceptable (green) if it is 98% or higher, and unacceptable (red) otherwise.

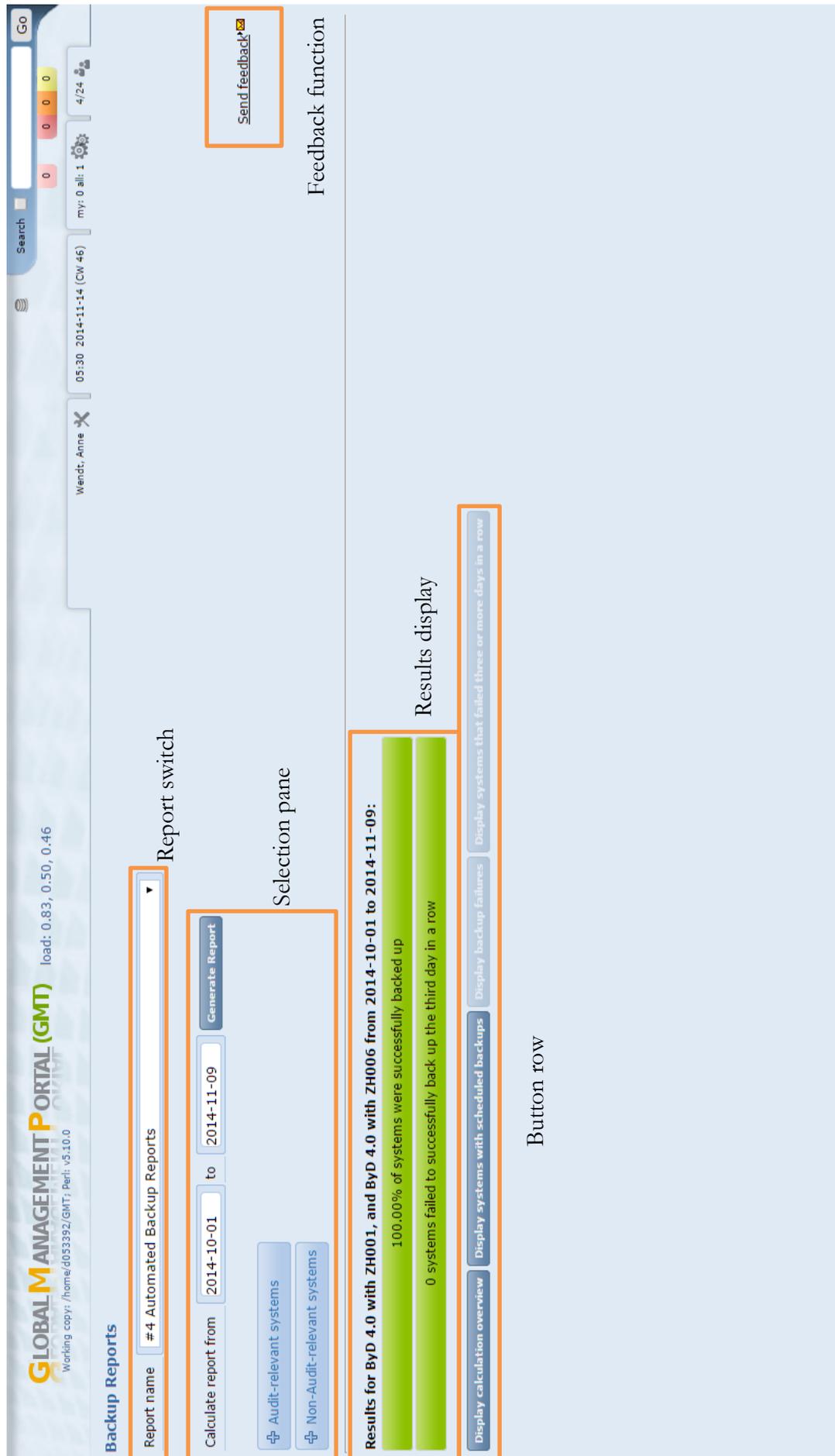


Figure 5-6: Result display of the software tool

The value in the second box is retrieved by counting how many systems failed for three or more days in a row. All entries from the FAILED table whose *date* is within the timeframe and whose *query_id* is among the ones specified by the user are selected when their *count* value is greater than two. This also covers cases where a system's backup first fails for three days in a row, then the system has a successful backup, and then the backup fails again three or more days in a row. As specified in SAP cloud policies, such systems are counted twice. In addition, there is a special case that has to be considered for this calculation: A system that has failed for the third time in a row before the user-specified timeframe begins (thus, having a *count* value greater than three on the first day of the timeframe) also has to be counted in the report. The sum of this calculation is displayed in the second box. As specified in SAP's cloud customer contracts, this value is only acceptable (green) if it is zero.

Below the two result boxes there is a row of buttons through which the user can access more detailed information about which systems were active and which systems had backup failures. The first three buttons display all entries in the OVERVIEW, SCHEDULED, and FAILED tables, respectively, that fulfil the selected *date* and *query_id* criteria. The fourth button displays all entries from the FAILED table that fulfil the selected criteria and whose *count* value is three or higher. This distinction was made so that users can instantly get a list of systems that need their attention most, whereas systems that only failed once or twice and then had a successful backup on the following day do not require further investigation from the auditors' perspective. On the screenshot shown in Figure 5-6, the third and the fourth button are disabled. This is due to the fact that there are no systems available which could be displayed, as the success rate is 100%. The third button is enabled if the success rate is lower than 100%, and the fourth button is enabled if there is at least one system that has not had a successful backup for three or more days.

A feature that can also be found on the result screen (Figure 5-6) is the feedback button in the top right corner. Clicking the mail icon opens a new email that is addressed to the researcher, so that users can send their feedback to the responsible person. This feature fulfils requirement 19.

When clicking on buttons two, three, or four, a list of systems is displayed. As requested in requirement 9, all systems have a little coloured dot next to them which indicates the system's current backup status. In order to determine this, the software tool retrieves the last backup status for each system while loading the table. The seven different outcomes for the possible system states are summarised in Table 5-1.

Table 5-1: List of mapping system states into coloured dots

Condition	Colour
System does not have a backup defined or backup is set to inactive	red
The system's backup has finished with an error or a warning	orange
The system has multiple backups defined	yellow
System's backup is currently running	blue
System is not part of the query anymore	turquoise
The last status of the system's backup was successful	green
System was decommissioned	grey

The colours were chosen based on what was available in GMP, and the general perceived associations with colours as summarised by F. M. Adams and Osgood (1973). Red, orange, and yellow were chosen to represent errors as they generally have a more negative connotation than green or blue. Also, the more severe the error is, the “redder” the dot is. Systems that are marked blue, turquoise, or green, on the other hand, do not require further investigation. Grey was chosen for a decommissioned (deleted) system as in information technology it is also associated with inactivity.

The requirements that were found to be not implementable with the current technical base have also been accommodated for in the UI. For example, it would not require much effort to include a list of affected customers (2) or a detailed error message (12) for each system in the tables that are displayed when further details are requested by the user. Requirements 17 and 32 could also be included if a technique was found to formalise the process of updating policies and ByD system versions, respectively, and trigger the creation or modification of queries. The additional or changed descriptions can be displayed together with the other checkboxes in the selection pane.

5.4 CHAPTER SUMMARY

This chapter discusses in detail the design and the implementation of the software tool that was developed as a solution for the research problem and incorporated the requirements elicited in Chapter 4. The inner workings of the software, such as the layout of the database schema and how information is gathered, are also described. This is followed by an introduction to the user interface, which is illustrated by screenshots.

CHAPTER 6 DISCUSSION

This chapter discusses the research findings described in the previous chapters. Within the context of the research methodology used in this study, it represents the Demonstration and Evaluation stages of the Design Science methodology. In Figure 5-1, these stages are marked by brown boxes.

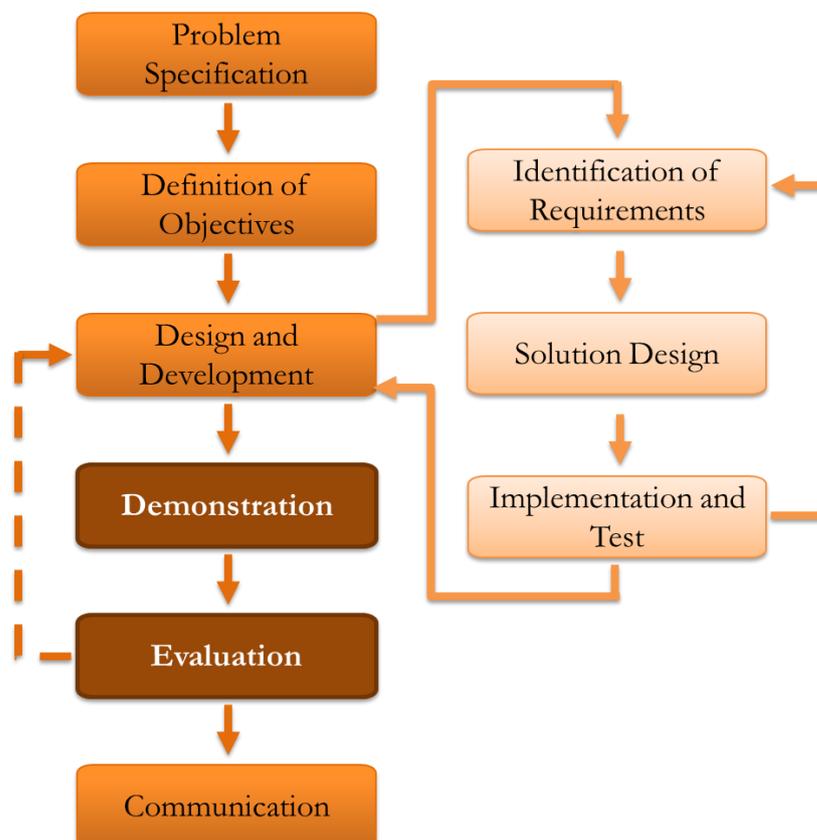


Figure 6-1: Demonstration and evaluation in the context of the methodology

First, the software tool that was developed is evaluated in terms of how it meets the interviewees' requirements and how its functionality addresses the problems they faced. Then, the study's research validity is discussed. Finally, a general, implementation-independent framework is proposed.

6.1 SOFTWARE ARTEFACT

This section discusses the software tool that was developed. First, the features of the software are compared to the interviewees' requirements in order to find out to what extent the experts' requirements were followed. Then, the software's ability to support the SAP cloud backup compliance processes is analysed.

6.1.1 COMPLIANCE WITH USER REQUIREMENTS

During the testing phase of the implementation the software’s functionality was also evaluated in relation to the user requirements to see if it included all features that were requested. For this purpose the requirements were grouped into five categories that are described separately. These groups are presented in Figure 6-2 on the bottom from left to right: requirements that could not be implemented due to technical restrictions, requirements that could not be implemented due to the limited timeframe, requirements that were implemented and can be evaluated by checking if the functionality exists, requirements that became part of the query functionality, and requirements that were implemented but need to be evaluated by other means. The number in brackets indicates the number of requirements in the respective group.

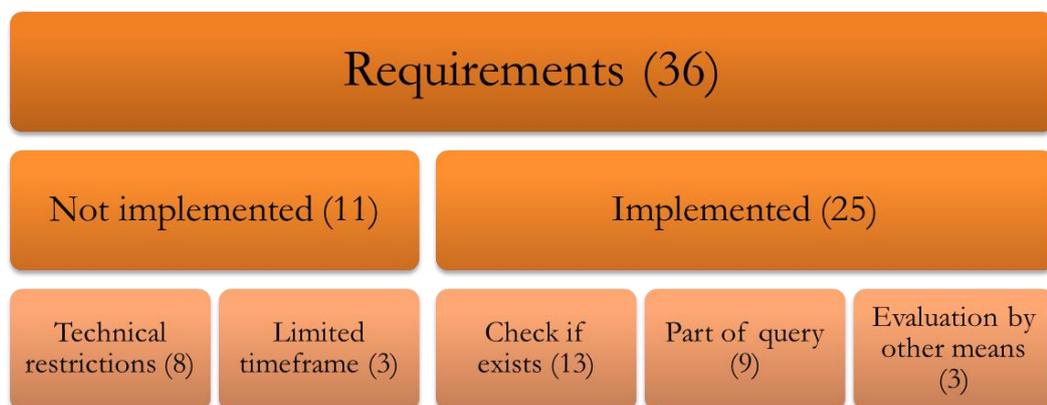


Figure 6-2: Categorisation of requirements implementation

As explained in section 5.1, there were several requirements that could not be implemented due to technical restrictions. This included requirement 2 (show list of affected customers; *high* priority), requirement 12 (display detailed error messages for backup failures; *medium*), requirement 17 (automatically include policies in the software; *low*), and requirement 32 (automatically add new ByD versions; *medium*). Furthermore, requirements 3 and 30 (both dealing with monitoring the transition of backups to secondary storage; *high* and *medium*) and requirements 18 and 34 (both related to the reporting over backup alerts; both *low*) were excluded from the implementation as they are not directly related to the core task of monitoring the backup success rate. Therefore, there is no functionality in the software tool matching these requirements.

The second group of requirements includes all requirements that were found to be useful but could not be included in the software tool due to the limited timeframe of the study. These are requirements 36 (mark/unmark all checkboxes; *low*), 20 (save sets of checkboxes; *low*), and requirement 11 (send out automatic reports; *medium*).

From the interviewees' points of view, the exclusion of requirements is unsatisfactory, as their requests could not be fulfilled. Since they are mainly concerned about the actual software tool and less about its generalisation as a framework, a missing feature means that not all of their requirements are addressed in the tool. However, it has to be noted that requirements 18 and 34 were implemented in another area of GMP, and it is planned to implement requirements 3 and 30 in an additional reporting tool. Furthermore, the functionality to send out automatic reports (requirement 11) is already part of the software design, which facilitates a quick implementation.

More importantly it is worth reiterating the overall intent of this work. As indicated in section 1.3, the research objective of this study is to develop and implement an automated solution that can deal with all varieties of technical entities and tasks that arise while monitoring backups in a cloud infrastructure. This solution should not be limited like the software tool is. Therefore, the requirements that were not implemented in the tool are included in the framework, which is independent from any underlying technical implementation and is described in section 6.3.

The remaining 25 requirements (out of the 36 in total) were addressed in the developed software tool. The first group in this category consists of 13 requirements that can be evaluated by checking if their specified functionality exists.

- Requirement 5 (take only the best backup into consideration; *high* priority) was implemented by iterating over all backups that are connected to a system, and then selecting one which was successful on the specified day. In the source code of *bkp_rep_daily_data_collector.pl* (Appendix J), this is the `foreach` loop in lines 130-154.
- Requirement 9 (display coloured dot to show a system's status; *medium*) is implemented in the front end script *Monitoring.pm* in the method `_prepare_CCMS_SID` (Appendix J Source Code, lines 564-741). This method checks for all possible states of a system and its backup, and adds the dots to the string that displays the system ID.
- Requirement 14 (send error notifications to the developer; *medium*) was implemented by adding the data collection script to the list of systems in GMP that can raise an alert if they end with any error message. As this is meta information which is maintained in GMP, it can not be shown in the source code.
- Requirement 15 (identify responsible employees; *medium*) can not be shown in the source code either. It was fulfilled by reading operation documentations for the different cloud applications and then verifying the information with the people whose names were in those files.

- Requirement 16 (retrieve basic population; *low*) had already been implemented in the pre-study, but the participant who requested this feature was not aware of it. The participant was informed how they could use the tool to gather the required information. In the source code, it is implemented in the *Monitoring.pm* lines 389-473.
- For requirement 19 (send feedback to the developer; *low*) a link in the form of an envelope is displayed on the UI, which when clicked opens a new email addressed to the developer (the researcher is the leading developer for the tool at the moment) so that the user can communicate with the developer. This can be found in the source code of *Monitoring.pm* in lines 83-96.
- For requirement 21 (schedule collection script early; *low*) the researcher chose a time in collaboration with the GMP development team that would satisfy the constraints of being as early as possible, but yet considering the German time zone since the main GMP server is located in Germany.
- Requirement 23 (use CCMS_SID instead of SID; *persistent*) was fulfilled as it can be seen in the source code for all occasions that involve uniquely identifying a system, for example in line 101 of *bkp_rep_daily_data_collector.pl* or line 194 of *Monitoring.pm*.
- Requirements 25 (write readable and maintainable code; *persistent*) and 33 (follow development guidelines; *persistent*) were both checked by participant 4, who raised the request. According to the participant, the “programming style needs improvement, but [the researcher] followed [their] development method” so that these two requirements can be seen as fulfilled (participant 4, personal communication, 22nd September 2014).
- Requirement 27 (historicisation of data; *persistent*) was achieved by designing the data flow in such a way that current (i.e., daily) information on systems is collected and stored permanently in a table.
- Requirement 28 (data source is manipulated only by automated script; *persistent*) was fulfilled by making the data collection script the only way of writing into the three data tables, which ensured their integrity as calculation base for everything else.
- Requirement 31 (change the display text; *low*) is the last requirement belonging to the group of requirements that can be assessed by checking if the functionality exists in the software. It was fulfilled by consulting the participant who raised this requirement and using their advice on which wording to use.

At the end of the development process, the participants who raised these requirements were consulted for a short evaluation session in which they confirmed that “their” requirement had been implemented according to their specification.

Nine more requirements were implemented in the newly developed functionality of using queries. This feature was assessed by watching participants use the software. Requirements 1 and 29 (filter by and display technical landscape; *high* and *medium* priority) can now both be fulfilled by adding a new query that includes a filter criterion which specifies the technical landscape. Participant 1 stated that even though the query functionality was not exactly what they requested, it still “covers [their] needs and will be more efficient in the long run” (Participant 1, personal communication, 24th September 2014). Requirements 4 (add more system types; *high*), 6 (report over non audit-relevant systems; *high*), 7 (update information on upgraded systems; *high*), 8 (take obsolete system categories out of scope; *high*), 13 (show more system details; *medium*), and 24 (offer scope flexibility; *high*) were also included in the new query functionality. As the query can be defined freely, it allows the users to specify a group of systems that fits their needs. All participants appreciated the idea of freely configurable queries. However, some were concerned that there were now too many checkboxes on the screen which made it difficult for them to identify their queries quickly. For this reason, another solution will need to be developed in the future as a follow-up project, which could include the functionality of defining frequently used sets of checkboxes for easier handling of queries. (This was already stated in requirement 20, but not implemented due to the limited timeframe.) Finally, requirement 10 (let users configure automatic reports; *high*) is the requirement most directly related to the query functionality when compared to the other requirements that resulted in this feature. Overall, designing the software tool to use a freely configurable search query as basis for creating groups of systems was seen to be very beneficial for the users as it addresses multiple requirements at once and offers them the flexibility they need when reacting to auditors’ requests for evidence.

The last three requirements need more diverse means of evaluation. Requirement 26 (usability; *persistent*) was evaluated by watching four participants (1, 4, 5, and 6) use the software tool for the first time. Since three of them already knew the previous tool, they could compare the two applications. Participant 6 appreciated that the UI design of the software did not change too much (personal communication, 24th September 2014). However, participant 5 remarked that the old design did not seem suitable for the new query functionality, as the list of selectable checkboxes was too long and, therefore, confusing (personal communication, 25th September 2014).

Requirement 35 (good performance; *persistent*) also needed a refined way of evaluation. For this, it was measured how much time it would take for data to load. On average, the display of the two success criteria was very quick (under one second). For displaying details,

however, the loading time varied widely (up to 14 seconds), depending on the amount of data that was requested. Since this is a rather long time, participant 4 suggested including a progress bar that would show the users the current status of their request (personal communication, 22nd September 2014).

The remaining requirement 22 (create a portable concept; *high*) is not just a requirement by the user, but addressing it also contributes to the body of knowledge, as the generalisation of the software tool induces a solution that is independent of its surroundings and can be implemented in other environments as well. The full realisation of this requirement will be further discussed in section 6.3.

6.1.2 ASSESSMENT OF THE OVERALL FUNCTIONALITY

This section looks at the overall functioning of the software tool and discusses issues that were identified during the implementation and the operation of the tool.

The main purpose of the tool is to provide evidence for the audits in the form of reports. The two criteria the auditors are mainly interested in are the percentage of systems that had a successful backup every day in a given time period, and the number of systems that did not have a backup for three or more consecutive days. As the tool allows calculating and retrieving those measures, it can be concluded that it fulfils its main purpose, even though not all the interviewees' suggestions were implemented.

There were two major challenges to be overcome – the system heterogeneity and the large number of systems. Heterogeneity means that different systems represent different cloud applications and have different characteristics. Their few commonalities, however, allow creating the reports, as in GMP each system is represented by a unique entity to which a backup object that stores all information about the system's backup is connected. The configurable queries provide a flexible option that allows selecting a specified group of systems that can be viewed together, thus the issue of heterogeneity is addressed satisfactorily.

Due to the large number of systems to report on, the database tables holding information need to store large amounts of data. This was addressed by optimising the data storage process. For the systems that are scheduled for a backup, this is done by only saving their go-live and go-offline dates, rather than creating an entry for every day. Also, using the IDs from the query table as a foreign key in the other tables avoids duplicate data.

The circumstances in which the tool was developed imposed constraints on the implementation in terms of feasibility of certain features. During the interviews, it was discovered that the differences between some departments are huge, and that some processes are irreconcilable at the moment. While, for example, the department supporting HANA Enterprise Cloud has very mature processes which are almost fully automated, the employees in the SuccessFactors department still do many things manually and, as a direct consequence of this, can not provide the same level of regularity. As described in Chapter 1, these internal differences were caused by acquisitions, and the issues are currently being addressed by creating and implementing the One Delivery strategy. However, as the implementation of this strategy is still work in progress, a common underlying base of similar processes does not exist yet within the SAP cloud departments. Therefore, these circumstances posed a constraint on the development of the software.

This constraint also affected the way the research objective was met. Initially, it was planned to implement the reporting tool across all SAP cloud applications. However, the interviews with participants 7 and 8 showed that this is not possible yet due to the divergent backup processes. Therefore, the objective had to be modified. The scope of the software tool was narrowed down to only supporting cloud applications that were managed in GMP. However, it was decided that a generic framework should be developed that could be applied to a variety of contexts. The software tool developed as part of this study can be viewed as an initial example of implementing this framework. The framework is described in detail in section 6.3.

6.2 RESEARCH VALIDITY

This section provides a brief discussion of the methodology that was used in the study. It assesses possible shortcomings of the chosen data collection process in order to evaluate the validity of this study.

Regarding the participants, three weaknesses were identified that could have possibly affected the findings. Firstly, the participant identification process implicates a certain level of limitation in terms of which business experts were consulted for this study. As agreed on with the AUT University Ethics Committee, the researcher's workplace supervisor identified potential participants who were then invited and could respond to the invitation in order to be interviewed for the research. This means that the pool of participants was restricted by the manager's knowledge and expertise.

Secondly, five of the ten participants who responded to the invitation came from the area of ByD, which could have shifted the problem focus towards that application. However, the remaining five non-ByD participants covered several other products, such as SuccessFactors, NetWeaver cloud (JPaaS), and HANA Enterprise Cloud, which created a more balanced view. Furthermore, the participants from ByD work in different roles, so that their viewpoints are quite diverse. As the collected requirements have shown, no two participants gave very similar answers. It has to be noted that not all of SAP's cloud products were covered by the participants; however, transforming their responses and the software tool into a more general set of guidelines alleviates this factor.

Thirdly, the personal situation of each participant could have influenced their responses. If they were currently working on a specific problem, their solution focus would be much stronger on that problem than on the general functionality. However, in the end the software tool was designed to help them solve their problems, so that the impact of this limitation should be low. In addition, it could be argued that the participants were pre-disposed to assess the researcher's work, given that they were work colleagues. However, the researcher tried to emphasise the importance of their open and honest feedback for the proper functioning of the software tool, which mitigated this risk.

6.3 THE FRAMEWORK

As noted in section 6.1.2 when discussing the overall functioning of the software, the objective of the research had to be modified based on the findings from the interviews. Especially participant 7, who coordinates the implementation of the One Delivery strategy that aims to unify all processes for SAP's cloud applications, made it clear that the original research objective could not be met as the unification is still in progress. Thus, the research objective had to be changed from designing and implementing a software-based solution that can be used by all cloud departments to a more generically applicable solution, in the form of a universally implementable framework for the development of backup monitoring tools.

This decision had several implications. First, the scope of the software had to be narrowed down to only considering cloud applications that are managed in GMP. Second, due to the need for a generalised approach, the major outcome would now be a method rather than an instantiation, based on the definition by the Design Science methodology which was employed in this research (Hevner et al., 2004). Third, this made the software tool become an instantiation of the method (again, as defined in the Design Science framework), and, at

the same time, the base from which the framework was abstracted. The following two sections describe and discuss the framework.

6.3.1 FRAMEWORK DESCRIPTION

The framework that was developed is based on the design considerations for the software tool as described in section 5.2. A graphical overview of the framework is presented in Figure 6-3. It shows how the four main parts of the framework (analysis, development, operation, and communication) are related to each other, which role is required to fulfil each part (indicated by the faces) and what the outcome of each part is.

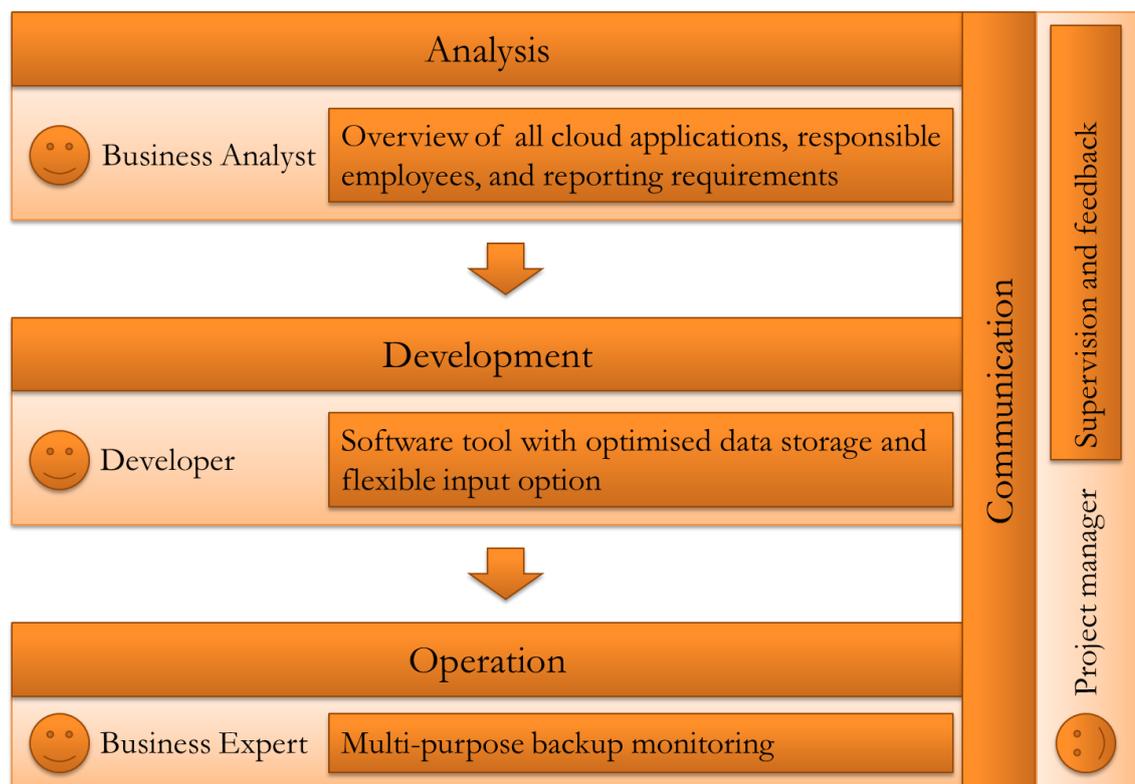


Figure 6-3: The proposed framework for the development of a backup monitoring tool

The framework consists of three stages which should be executed in order. First, a detailed **analysis** should be performed to get an initial overview of the company's processes for creating and monitoring backups of their customer data. Ideally, this analysis involves all departments that deal with the operation of cloud systems in the company, even if they were just recently acquired or newly established within the company. This is important for properly understanding the requirements which different departments have towards the task of backup monitoring in terms of compliance with standards that they are audited for, but also for internal controls of operational processes. Only if these requirements are fully understood, will the outcome of employing the framework be beneficial. It is also important to identify those employees who are responsible for monitoring a certain group

of systems, record their details with the definition of the group of systems for future reference, and keep this information updated regularly. That way, issues can be communicated with the appropriate people and, thus, be resolved faster.

As a result of the first stage, there should be a list of all cloud applications, attached customer systems, and employees who are responsible for a certain group of systems within the company. By creating these lists, the business analyst gains a thorough understanding of the problem space which is needed to create the best possible solution.

This knowledge is then transferred to the developer, who is the main actor in the second stage: **developing** an automated software-based solution. Since the number of systems that has to be dealt with is assumed to be large, a manual solution would impose an unachievable workload on the employees. The prerequisite for developing the software tool is that there already exists an automated tool which holds all data that are necessary for the reporting, such as a repository that contains information on systems and their backups, and that preferably has a connection to the backup software so that data on backup states can be retrieved without human intervention. However, when applying the framework in smaller companies, the issues associated with the number of systems might be smaller, which means that a semi-automated approach could be feasible.

When designing the software, the framework recommends employing a top-down approach to create a three-tier architecture containing presentation, logic, and data layers as this approach was used successfully in the implemented software tool. This means that first it should be decided how the data shall be displayed on the screen. This decision has to be made with the help of the business experts who are going to use the software tool to retrieve information which they need for their daily job.

The second layer, which contains the backend logic, shall then be built to fulfil the needs of the presentation layer. This means that those calculations needed for displaying the most important information for the user, such as the one for the backup success rate, shall be sufficiently fast and inexpensive in terms of calculation effort. Upon user request, details can be shown that the user needs to further analyse the data.

Furthermore, the logic layer is responsible for extracting the necessary data, i.e., filling the database tables with relevant information from the underlying management system. In order to achieve greatest accuracy, the data collection should be performed within the same time interval that is used for creating the system backups. That way, it is ensured that no backup cycle is missed, and thus, no backup failure is overlooked.

The logical layer should also accommodate a flexible input option in terms of user-defined groups of systems. This is especially important for addressing the issue of heterogeneity. For example, if the underlying system repository has a search feature, it can be employed to gather data on all systems belonging to a particular group by defining a search query. In any case, a group of systems shall be defined by certain characteristics which are shared among all systems in that group, and it must be ensured that this combination of characteristics is accurate enough to retrieve exactly those systems needed for a specific report. When testing the definition of a system group, the result of the query can be compared to the list of systems created in the first stage in order to check their correctness.

Based on the functioning of the logical layer, an optimised data storage model has to be created which forms the bottom tier of the architecture. One of the challenges that could be faced by the company implementing this framework is the vast number of systems and backups that has to be managed; special attention needs to be paid to the layout of the database tables in terms of storage optimisation for fast retrieval of information. The writing of data does not necessarily have to be optimised for time efficiency, because it does not involve user interaction but only runs in the background. Therefore, the process of filling the database tables can and should involve a reorganisation of the input information so that it is stored in a format that is targeted at efficiently providing those data that were identified as most important by the business experts in the analysis stage.

The final task of the development stage is testing the developed software. This task has to be performed with two aspects in mind. First, the functionality of the software has to be tested against the business experts' requirements by checking if they were implemented in the way they were expected. Second, it has to be verified if the figures computed by the reporting tool match the results of the calculations by the users.

The third stage in the framework is the **operation** of the tool where the software is used by an employee to monitor the backup success of their systems. It provides evidence for audits but can also be used for other purposes. As the software tool is meant to monitor a large number of systems, the need for sending out automated reports can arise, which means that reports that have to be created regularly do not require human intervention anymore. This issue can be addressed by implementing an email sending functionality within the automatically running data collection script, so that it is triggered in the same interval that is used for creating backups, and it always yields the most recent results.

Another important aspect of the operation is ongoing technical support for the software. The developer has to make sure that errors are detected at the earliest possible moment,

and fixed in a timely manner. Since the data that the software tool deals with is important for the company's compliance, it has to be made sure that in case of an error no information is lost and data that were not collected due to the error have to be gathered and entered into the tool to uphold its integrity and correctness.

All three stages of the framework are overarched by frequent and effective **communication**. An important part of this is to create a feedback cycle throughout the whole implementation project. Therefore, the person responsible for this part is the project manager. All employees involved in the project should be encouraged to communicate with each other from the early stage onwards in order to establish a mutual understanding of commonly used terminology and also of each other's roles within the project. As noted above, it is equally important that employees working in different positions understand each other's viewpoints so that they can agree on a common goal for the project.

Throughout the project, there are four roles involved which can either be shared responsibility or assigned to individual employees, depending on the size of the implementing department and the expertise of the employees involved. The first stage should be performed by a business or system analyst or someone with similar knowledge. This person's task is to analyse the company's processes as explained in the analysis stage. They then have to transfer their gained knowledge to a developer who is responsible for implementing the software. The third role is the business expert who delivers requirements for the software tool and assesses the tool's efficacy in the relevant development cycles. The fourth role is that of a project manager who oversees the project and is responsible for setting up functioning communication between the involved parties. For example, this can be done at short daily meetings in which everyone shares their current work progress. As a result, it is expected that misunderstandings are alleviated.

A cloud provider following the stages of the framework should be able to set up a functioning backup reporting system that is robust, scalable, and flexible.

6.3.2 ANALYSIS OF THE FRAMEWORK

This section analyses how the framework addresses the objective of the research.

The main research objective was to develop and implement an automated solution that could deal with all varieties of technical entities and tasks that arise while performing backup monitoring in a cloud infrastructure. The required implementation was carried out and a software tool was developed that was based on the user requirements elicited from

interviews and described in Chapter 5. However, it was also found that the software could not yet address the research objective fully, as it faced limitations due to the technical base in which it was implemented. For this reason, a more general framework was developed, which aimed at overcoming those limitations and creating a universally deployable solution.

Having a framework rather than only a piece of software as the outcome also addresses another objective, which was to make the developed underlying theories independent from the actual implementation in the SAP environment. Since the framework describes a generic process that could be followed by the implementing departments, it can be adapted to the company's situation and specific circumstances. As long as the described framework is used, the result of the implementation is believed to be of similar quality for the task of efficiently monitoring backups as the software tool described in Chapter 5.

The criteria on which the company that is implementing the framework has to report are freely definable. The design of the solution shall follow these criteria, which are determined by interviewing business experts. This means that the framework is not just technologically independent, but also adaptable to the company's policies related to backup monitoring. That way, it addresses the research aim to support cloud providers when setting up their monitoring processes in terms of customer data backups.

The two main challenges that the framework needs to be able to deal with are heterogeneity and multitude. These were addressed by instructing the developer in the second stage of the framework to provide a flexible input option and an optimised way of data storage. In combination with the information gathered in the first stage about the company's cloud infrastructure, the framework attempts to simplify the process of tailoring the software to the company's needs.

The practical application of the study was to support SAP's implementation of their One Delivery strategy in the area of backup reporting, as described in section 1.3 as one of the aims of the research. As SAP were the research partner of the project, they presented the problem definition according to their understanding. Therefore, the outcome is oriented towards fulfilling SAP's needs in order to address the described aim. However, the framework does not use any SAP terminology; neither does it assume any SAP-specific prerequisites. The only constraint for implementing the framework is that there has to be a software-based tool available which is used to manage all system and backup data. This management tool can then be employed to serve as the base for the framework. Consequently, the framework can also be implemented in other areas of the SAP

infrastructure, which addresses the aim of supporting them with the execution of their One Delivery strategy.

The implementation of the framework requires four roles, one for each stage of the framework. Depending on the size of the company, roles can be combined and taken up by only one employee; given that employee has the required skills and knowledge. Being able to complete the assigned task properly is essential for a good result that creates value for the company.

6.4 CHAPTER SUMMARY

This chapter discussed the outcome of the research. First, the implemented software tool was evaluated in terms of how well it met the requirements gathered by interviewing business experts. Then it briefly discussed possible issues with the data collection. In the third section, a framework was developed and discussed that addressed the research objectives independently from the underlying technical and organisational environment.

CHAPTER 7 CONCLUSION

This thesis describes an application for monitoring backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment, from which later a framework was derived. The objective of the research was to address the issues that arise when a cloud provider wants to ensure compliance with certain audit standards in the area of backup monitoring. The research partner for this study was SAP, a software company that offers cloud services to businesses and operates globally. Due to acquisitions of other cloud providers, they needed to integrate their operational processes, especially in the area of backup where procedures varied widely between departments. Therefore, the practical application of the study was to support this integration of processes.

In the literature review no publications could be found that addressed this issue directly. However, four areas of research context were consulted, namely backup technology, heterogeneity, providing cloud services, and cloud security. Out of this analysis, the research question was formulated as follows: **How can we monitor backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment?**

The methodology that was used to address this question was a combination of Design Science and the software development lifecycle. According to the Design Science methodology, the artefact that was created had to be designed considering a clear problem definition and afterwards had to be thoroughly evaluated for its efficacy to address the research problem. A pre-study that had been conducted prior to this study gave valuable insights into the design process of the artefact. Several methods for gathering requirements were reviewed with the result that interviewing business experts would be most appropriate in this case. Considering the practical and theoretical insights, the Design Science methodology was expanded by using the software development lifecycle to design and develop the artefact.

The first outcome of the study was a set of 36 requirements which were gathered from interpreting the interview responses. The requirements were then further analysed for interdependencies using a method developed by the researcher during the study. In addition to the three major outcomes of the study, this method can be viewed as an intermediate contribution to the research as it provided a base for further exploration of the data. It was found that there were sets of functionality which were not strongly related to the actual task of monitoring backup performance, while some requirements were

heavily interconnected with others and, therefore, classified as highly important for being included in the subsequently developed software tool. The software tool was implemented in the SAP environment and aimed to address the research question by providing a way to monitor backups in an actual business cloud computing setting. The two challenges, heterogeneity and large number of systems, were addressed by providing a flexible way of defining system collection criteria, and by designing the underlying database tables to store information efficiently (for example, defining a go-live and go-offline date for scheduled systems instead of creating a new entry every day). This tool is the second major outcome of the study and together with the requirements provided the foundation for the development of the framework.

This framework is the third outcome of the study and its main contribution to the body of knowledge. As stated in the research objective in Chapter 1, there are two outcomes of this study, which are a software tool that was implemented in the SAP environment and a framework that was developed as a generalisation of the software tool. As the software tool is specific to SAP, its usefulness for other cloud providers is probably rather low, whereas the framework is independent of technical constraints. Therefore, it can be implemented in other environments with alterations specific to the needs of the respective provider. The framework's efficacy for the task of backup monitoring was achieved by interviewing business experts and incorporating their recommendations in the design of the software tool, from which the framework was later generalised. Hence, the interviewees' expertise contributed directly to this outcome. Since the audits follow standardised processes that require maintaining a reliable and stable backup process, it can be anticipated that other cloud providers face challenges similar to the ones at SAP. Therefore, the insights gained during the research will be valuable for them, too.

As identified by Brinberg and McGrath (1985) there are three major types of contributions: methodological (novel approaches to the work that could be used by others), substantive (findings and outcomes of value specific to the study), and conceptual (models and frameworks of value that could generalise beyond the study). All three types were covered by this research. A methodological contribution is the approach of identifying interconnections between requirements by using post-it notes on a whiteboard. Substantive outcomes are the set of requirements and the software tool, as they are both specific to the study within the SAP environment. The framework is a conceptual contribution as it is designed to provide value beyond the study setting.

There are a few aspects that limit the scope of the contributions. From a technical point of view, it is possible that the implementation in GMP imposed restrictions on potential solutions that could have been identified in a different environment. Furthermore, the relatively low number of participants might have created a bias as the participants' focus was limited to their experience. However, this issue was addressed by asking a manager to select potential interviewees. Lastly, the One Delivery strategy is still work in progress, which means that circumstances will change and could create opportunities for new solutions. However, those limitations mostly affect the software tool, while the framework is independent of the implementation environment. Furthermore, the tool was made to be extensible and portable to cope with those changes to come.

In the future, more functionality and reporting options could be included in the software tool. For example, the feature of sending out reports automatically could be implemented, and also options could be added for the users to specify queries and to configure their reports in more detail and for a greater scope. Furthermore, a feature that allows the users to maintain countermeasures for specific backup failures could be helpful, as this would be a direct consequence of what the tool reports. The framework itself could also be used and further refined by others in order to investigate its efficacy and applicability in different settings.

Once the One Delivery strategy is implemented, the scope of the backup process will greatly expand, and the software tool will have to be adapted in order to meet the new requirements that will arise during the consolidation. For this, the proposed framework can be used to develop a new automated solution.

REFERENCES

- 451 Research LLC. (2013). *2013 Cloud Computing Outlook – Cloud Computing Wave 5*. Retrieved from <https://451research.com/report-long?icid=2831>
- Adams, F. M., & Osgood, C. E. (1973). A Cross-Cultural Study of the Affective Meanings of Color. *Journal of Cross-Cultural Psychology*, 4(2), 135-156. doi: 10.1177/002202217300400201
- Adams, M. (2001). How snapshot technology will change the future of backup and recovery. *Computer Technology Review*, 21(1), 26-27.
- Badger, L., Grance, T., Patt-Corner, R., & Voas, J. (2012). Cloud computing synopsis and recommendations. *NIST special publication*, 800-146.
- Bauer, E., & Adams, R. (2012). *Reliability and availability of cloud computing*. Retrieved from <http://aut.ebib.com.au.ezproxy.aut.ac.nz/patron/FullRecord.aspx?p=875898>
- Beckers, K., Côté, I., Faßbender, S., Heisel, M., & Hofbauer, S. (2013). A pattern-based method for establishing a cloud-specific information security management system. *Requirements Engineering*, 18(4), 343-395. doi: 10.1007/s00766-013-0174-7
- Benington, H. D. (1983). Production of large computer programs. *IEEE Annals of the History of Computing*, 5(4), 350-361.
- Berenbach, B., Paulish, D. J., Kazmeier, J., & Rudorfer, A. (2009). *Software & systems requirements engineering: in practice*. Retrieved from <http://accessengineeringlibrary.com.ezproxy.aut.ac.nz/browse/software-and-systems-requirements-engineering-in-practice>
- Bijan, Y., Yu, J., Stracener, J., & Woods, T. (2013). Systems requirements engineering - State of the methodology. *Systems Engineering*, 16, 267-276. doi: 10.1002/sys.21227
- Bjorvatn, K. (2004). Economic integration and the profitability of cross-border mergers and acquisitions. *European Economic Review*, 48(6), 1211-1226. doi: 10.1016/j.euroecorev.2004.03.007
- Boomer, J. (2012). Do your backup strategies involve the cloud? *CPA Practice Advisor*, 22(11), 34.
- Braude, E. J. (2004). *Software design: From programming to architecture*. Hoboken, NJ: John Wiley & Sons, Inc.
- Brinberg, D., & McGrath, J. E. (1985). *Validity and the research process*. Beverly Hills, CA: SAGE.
- Chang, W. Y., Abu-Amara, H., & Sanford, J. F. (2010). *Transforming enterprise cloud services*. Dordrecht, The Netherlands: Springer.
- Chaput, S. R., & Ringwood, K. (2010). Cloud compliance: A framework for using cloud computing in a regulated world. In N. Antonopoulos & L. Gillam (Eds.), *Cloud computing: Principles, systems and applications* (pp. 241-255). London, England: Springer.
- Charmaz, K. (2014). *Constructing grounded theory*. Los Angeles, CA: Sage Publications.
- Cisco Inc. (n.d.). Acquisitions - Acquisition summary - Cisco systems. Retrieved 20th April, 2014, from http://www.cisco.com/web/about/doing_business/corporate_development/acquisitions/ac_year/about_cisco_acquisition_years_list.html

- Credo General Reference. (2006). *High definition: an A to Z guide to personal technology*. Retrieved from <http://search.credoreference.com.ezproxy.aut.ac.nz/content/title/hmhighdef>
- Dölitzscher, F., Reich, C., Knahl, M., & Clarke, N. (2013). Understanding cloud audits. In S. Pearson & G. Yee (Eds.), *Privacy and security for cloud computing* (pp. 125-163). London, England: Springer.
- Durbano, J. P., Rustvold, D., Saylor, G., & Studarus, J. (2010). Securing the cloud. In N. Antonopoulos & L. Gillam (Eds.), *Cloud computing: Principles, systems and applications* (pp. 289-302). London, England: Springer.
- Gaskin, F. (2009). Goodbye SAS 70 hello ISAE 3402? *Accountancy Ireland*, 41(3), 28-31.
- Gaskin, F. (2010). Roll over, SAS 70 it's time for ISAE 3402 and SSAE 16. *Accountancy Ireland*, 42(5), 12-14,16.
- Gilb, T. (1985). Evolutionary delivery versus the "waterfall model". *ACM SIGSOFT Software Engineering Notes*, 10(3), 49-61. doi: 10.1145/1012483.1012490
- Gillham, B. (2001). *Research interview*. Retrieved from <http://aut.ebib.com.au.ezproxy.aut.ac.nz/patron/FullRecord.aspx?p=436490>
- Guntamukkala, V., Wen, H. J., & Tarn, J. M. (2006). An empirical study of selecting software development life cycle models. *Human Systems Management*, 25(4), 265-278.
- Halpert, B. (2011). *Auditing cloud computing: A security and privacy guide*. Retrieved from <http://aut.ebib.com.au.ezproxy.aut.ac.nz/patron/FullRecord.aspx?p=697627>
- Heitmann, R. (2007). Back up the virtual machine. *Communications News*, 44(10), 30,32.
- Hevner, A. R., & Chatterjee, S. (2010). *Design research in information systems: theory and practice*. New York, NY: Springer.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105. doi: 10.2307/25148625
- Hill, R., Hirsch, L., Lake, P., & Moshiri, S. (2013). *Guide to cloud computing: principles and practice*. London, England: Springer.
- Hogan, M., Liu, F., Sokol, A., & Tong, J. (2011). NIST cloud computing standards roadmap. *NIST special publication 500-291*. http://bigdatawg.nist.gov/_uploadfiles/M0009_v1_7425925966.pdf
- Hunter, P. (2004). Better backup and recovery with snapshots. *Computer Technology Review*, 24(10), 34.
- IBM Inc. (n.d.). IBM Investor relations - Our strategy | Selected acquisitions. Retrieved 20th April, 2014, from <http://www.ibm.com/investor/strategy/acquisitions.wss>
- International Auditing and Assurance Standards Board. (2010). International Standard on Assurance Engagements (ISAE) 3402. <http://www.ifac.org/sites/default/files/downloads/b014-2010-iaasb-handbook-isae-3402.pdf>
- International Organization for Standardization. (n.d.-a). ISO Standards - ISO. Retrieved May 2, 2013, from <http://www.iso.org/iso/home/standards.htm>
- International Organization for Standardization. (n.d.-b). ISO/IEC 27001 - Information security managements. Retrieved 22nd July, 2014, from <http://www.iso.org/iso/iso27001>
- International Organization for Standardization. (n.d.-c). ISO/IEC CD 19086-1 - Information technology -- Cloud computing -- Service level agreement (SLA) framework and Technology -- Part 1: Overview and concepts. Retrieved 25th July,

- 2014, from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67545
- International Organization for Standardization. (n.d.-d). ISO/IEC CD 27017 Information technology -- Security techniques -- Code of practice for information security controls for cloud computing services based on ISO/IEC 27002. Retrieved 22nd July, 2014, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=43757
- International Organization for Standardization. (n.d.-e). ISO/IEC DIS 17788:2014 - Information technology -- Cloud computing -- Overview and vocabulary. Retrieved 25th July, 2014, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=60544
- International Organization for Standardization. (n.d.-f). ISO/IEC DIS 17789:2014 - Information technology -- Cloud computing -- Reference architecture. Retrieved 25th July, 2014, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=60545
- International Organization for Standardization. (n.d.-g). ISO/IEC PRF 27018:2014 - Information technology -- Security techniques -- Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors. Retrieved 22nd July, 2014, from http://www.iso.org/iso/catalogue_detail_ics.htm?csnumber=61498
- Jansen, W., & Grance, T. (2011). Guidelines on security and privacy in public cloud computing. *NIST special publication 800-144*.
- Jiang, L., Eberlein, A., Far, B. H., & Mousavi, M. (2008). A methodology for the selection of requirements engineering techniques. *Software & Systems Modeling*, 7, 303-328. doi: 10.1007/s10270-007-0055-y
- Jiménez-Domingo, E., Lagares-Lemos, Á., & Gómez-Berbís, J. M. (2011). Multitenancy: A new architecture for clouds. In L. Wang, R. Ranjan, J. Chen & B. Benatallah (Eds.), *Cloud computing: Methodology, system, and applications* (pp. 261-275). Boca Raton, FL: CRC Press.
- Jin, H., Ibrahim, S., Bell, T., Qi, L., Cao, H., Wu, S., & Shi, X. (2010). Tools and technologies for building clouds. In N. Antonopoulos & L. Gillam (Eds.), *Cloud computing: Principles, systems and applications* (pp. 3-20). London, England: Springer.
- Kamel, M. N., & Kamel, N. N. (1992). Federated database management system: Requirements, issues and solutions. *Computer Communications*, 15, 270-278. doi: 10.1016/0140-3664(92)90110-Z
- Katsaros, G., Kübert, R., Gallizo, G., & Wang, T. (2011). Monitoring: A fundamental process to provide QoS guarantees in cloud-based platforms. In L. Wang, R. Ranjan, J. Chen & B. Benatallah (Eds.), *Cloud computing: Methodology, system, and applications* (pp. 325-341). Boca Raton, FL: CRC Press.
- Knauss, E., Damian, D., Cleland-Huang, J., & Helms, R. (2014). Patterns of continuous requirements clarification. *Requirements Engineering*. doi: 10.1007/s00766-014-0205-z
- Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: Processes and techniques*. Chichester, England: John Wiley & Sons Inc.
- Kujala, S. (2005). Linking user needs and use case-driven requirements engineering. In A. Seffah, J. Gulliksen & M. C. Desmarais (Eds.), *Human-centered software engineering - Integrating usability in the software development lifecycle* (Vol. 8, pp. 113-125). Amsterdam, The Netherlands: Springer.

- Lauesen, S., & Kuhail, M. A. (2011). Use cases versus task descriptions. In D. Berry & X. Franch (Eds.), *Requirements Engineering: Foundation for Software Quality* (Vol. 6606, pp. 106-120). Berlin, Germany: Springer.
- Leffingwell, D., & Widrig, D. (2003). *Managing software requirements: A use case approach*. Boston, MA: Addison-Wesley.
- Li, H., & Su, S. Y. W. (2001). Business object modeling, validation, and mediation for integrating heterogeneous application systems. *Journal of Systems Integration*, 10, 307-328. doi: 10.1023/A:1011246500072
- Little, D. B., & Chapa, D. A. (2003). *Implementing backup and recovery: The readiness guide for the enterprise*. New York, NY: Wiley.
- Maloney, M. M., & Zellmer-Bruhn, M. (2006). Building bridges, windows and cultures: Mediating mechanisms between team heterogeneity and performance in global teams. *Management International Review*, 46, 697-720. doi: 10.1007/s11575-006-0123-5
- Marinescu, D. C. (2013). *Cloud computing: Theory and practice*. Retrieved from <http://aut.eblib.com.au.ezproxy.aut.ac.nz/patron/FullRecord.aspx?p=1213925>
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *NIST special publication 800-145*. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- Missbach, M., Stelzel, J., Gardiner, C., Anderson, G., & Tempes, M. (2013). *SAP on the cloud*. Heidelberg, Germany: Springer.
- Nelson, S. (2011). *Pro data backup and recovery*. New York, NY: Apress.
- Nicolescu, G., O'Connor, I., & Piguet, C. (2012). Introduction. In G. Nicolescu, I. O'Connor & C. Piguet (Eds.), *Design technology for heterogeneous embedded systems* (pp. 1-10). Dordrecht, The Netherlands: Springer.
- O'Bannon, I. M. (2012). Five reasons cloud-based backup is better. *CPA Practice Advisor*, 22(3), 6-7.
- Onwubiko, C. (2010). Security issues to cloud computing. In N. Antonopoulos & L. Gillam (Eds.), *Cloud computing: Principles, systems and applications* (pp. 271-288). London, England: Springer.
- Parthasarathy, S. (2013). Potential concerns and common benefits of cloud-based enterprise resource planning (ERP). In Z. Mahmood (Ed.), *Cloud computing: Methods and practical approaches* (pp. 177-195). London, England: Springer.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77. doi: 10.2307/40398896
- Pham, H. (2007). Software development lifecycle and data analysis *System Software Reliability* (pp. 121-151). London, England: Springer.
- Preston, W. C. (2007). *Backup and recovery*. Sebastopol, CA: O'Reilly.
- Pricewaterhouse Coopers. (2013). *PwC Global 100 Software Leaders*. Retrieved from http://www.pwc.com/en_US/us/technology/publications/assets/pwc-global-software-100.pdf
- Raj, P. (2012). *Cloud enterprise architecture*. Retrieved from <http://aut.eblib.com.au.ezproxy.aut.ac.nz/patron/FullRecord.aspx?p=1048981>
- Ralph, P. (2013). The illusion of requirements in software development. *Requirements Engineering*, 18, 293-296. doi: 10.1007/s00766-012-0161-4

- Rangachari, R. (1992). Enterprise-wide backup both heterogeneous and scalable. *Computer Technology Review*, 12(12), 34.
- Ristov, S., Gusev, M., & Kostoska, M. (2012). Cloud computing security in business information systems. *International Journal of Network Security & Its Applications*, 4(2), 75-93. doi: 10.5121/ijnsa.2012.4206
- Rowley, J. (2012). Conducting research interviews. *Management Research Review*, 35, 260-271. doi: <http://dx.doi.org/10.1108/01409171211210154>
- Royce, W. W. (1970). Managing the development of large software systems. *proceedings of IEEE WESCON*, 26(8).
- Rumbaugh, J. (1994). Getting started - Using use cases to capture requirements. *Journal of Object-Oriented Programming*, 7(5), 8-12, 23.
- Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), 8-13. doi: 10.1145/1764810.1764814
- Sahandi, R., Alkhalil, A., & Opara-Martins, J. (2012). SMEs' perception of cloud computing: Potential and security. In L. M. Camarinha-Matos, L. Xu & H. Afsarmanesh (Eds.), *Collaborative networks in the internet of services* (pp. 186-195). Heidelberg, Germany: Springer.
- SAP AG. (2013). *2013 Annual Report*. Retrieved from <http://global.sap.com/corporate-en/investors/pdf/sap-2013-annual-report.pdf>
- SAP AG. (n.d.-a). Acquisitions | Financial news and reports, acquisitions | Investor relations | About SAP AG | SAP. Retrieved 9th April, 2014, from <http://global.sap.com/corporate-en/investors/newsandreports/acquisitions.epx>
- SAP AG. (n.d.-b). Cloud infrastructure | Cloud solutions | SAP. Retrieved April 16, 2014, from <http://www.sap.com/pc/tech/cloud/software/hana-enterprise-cloud/index.html>
- SAP AG. (n.d.-c). History 1972-1981 | Company information | About SAP AG | SAP. Retrieved 3rd April, 2014, from <http://global.sap.com/corporate-en/our-company/history/1972-1981.epx>
- SAP AG. (n.d.-d). History 2002-2012 | Company information | About SAP AG | SAP. Retrieved 3rd April, 2014, from <http://global.sap.com/corporate-en/our-company/history/2002-present.epx>
- SAP AG. (n.d.-e). SAP cloud software portfolio | Cloud solutions | SAP. Retrieved 8th April, 2014, from <http://www.sap.com/pc/tech/cloud/software/overview/index.html>
- SAP AG. (n.d.-f). SAP company information | Corporate | SAP. Retrieved 3rd March, 2014, from <http://global.sap.com/corporate-en/our-company/index.epx>
- SAP AG. (n.d.-g). SAP HANA infrastructure services | New | SAP HANA marketplace. Retrieved April 16, 2014, from <http://marketplace.saphana.com/New/SAP-HANA-Infrastructure-Services-/p/1807>
- SAP Cloud Compliance (Producer). (2013). Cloud Data Security and Compliance at SAP. *SAP Business byDesign*. [PowerPoint Presentation]
- SAP News. (2014a). SAP continues to invest in data centers worldwide. Retrieved 6th May, 2014, from <http://www.news-sap.com/sap-continues-invest-data-centers-worldwide/>

- SAP News. (2014b). SAP to acquire Fieldglass, the global cloud technology leader in contingent workforce management. Retrieved 9th April, 2014, from <http://global.sap.com/news-reader/index.epx?articleId=22589>
- Scott, M., Boardman, R. P., Reed, P. A., & Cox, S. J. (2014). Managing heterogeneous datasets. *Information Systems*, 44, 34-53. doi: 10.1016/j.is.2014.03.004
- Scriba, A. (2009). Point in time copies (snapshots). In V. Herminghaus & A. Scriba (Eds.), *Storage management in data centers* (pp. 233-317). Berlin, Germany: Springer.
- Sheth, A. P., & Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), 183-236. doi: 10.1145/96602.96604
- Shimizu, K., Hitt, M. A., Vaidyanath, D., & Pisano, V. (2004). Theoretical foundations of cross-border mergers and acquisitions: A review of current research and recommendations for the future. *Journal of International Management*, 10, 307-353. doi: 10.1016/j.intman.2004.05.005
- Singleton, T. W. (2011). IT audit basics: Understanding the new SOC reports. *ISACA Journal*, 2, 1-3.
- Sotto, L. J., Treacy, B. C., & McLellan, M. L. (2010). Privacy and data security risks in cloud computing. *Electronic Commerce & Law Report*.
- Spiceworks Voice of IT. (2013). *The cloudification of the network*. Retrieved from http://itreports.spiceworks.com/reports/spiceworks_voice_of_it_pertino_iaas.pdf
- Srinivasan, U., Ngu, A. H. H., & Gedeon, T. (2000). Managing heterogeneous information systems through discovery and retrieval of generic concepts. *Journal of the American Society for Information Science*, 51(8), 707-723.
- Stober, T., & Hansmann, U. (2010). *Agile software development: Best practices for large software development projects*. Heidelberg, Germany: Springer.
- Tsai, B.-Y., Stobart, S., Parrington, N., & Thompson, B. (1997). Iterative design and testing within the software development life cycle. *Software Quality Journal*, 6(4), 295-309. doi: 10.1023/A:1018528506161
- Vinca Corp. (1995). SnapShot backs up live databases. *Computer Dealer News*, 11(14), 52.
- Waschke, M. (2012). *Cloud standards: Agreements that hold together clouds*. New York, NY: Apress.
- Wiegers, K. E. (1999). *Software requirements*. Redmon, WA: Microsoft Press.
- Young, R. R. (2003). *The requirements engineering handbook*. Boston, MA: Artech House.

APPENDICES

APPENDIX A PGR1 FORM



FORM PGR1 POSTGRADUATE RESEARCH PROPOSAL

PLEASE NOTE

- This form must be typed. Handwritten forms will not be accepted.
- Double clicking on the check boxes enables you to change them from not-checked to checked.

Student ID No	1386235	Name	Anne Wendt	
Faculty	Design and Creative Technologies	School/Dept.	Computing and Mathematical Sciences	
Ethnicity	German	Residency	Citizen/PR <input type="checkbox"/>	International <input checked="" type="checkbox"/>
Programme	Master of Computer and Information Sciences		Full-time <input checked="" type="checkbox"/>	Part-time <input type="checkbox"/>
Is an Ethics application required?		Yes	<input checked="" type="checkbox"/>	No <input type="checkbox"/>
Has a supervision agreement been completed? Supervision agreements are compulsory		Yes	<input checked="" type="checkbox"/>	No <input type="checkbox"/>
Thesis	<input checked="" type="checkbox"/> Dissertation	<input type="checkbox"/>	Points Value	120
Pathway 1	<input type="checkbox"/>	Pathway 2	<input type="checkbox"/>	Pathway 3 <input type="checkbox"/>

ADDITIONAL INFORMATION

- **Referencing:** when citing references please specify the referencing style you will use (e.g. APA). This specified style should be used throughout the entire proposal.
- **External parties:** when working with external parties, contractual arrangements should be in place to clarify your rights and responsibilities. The University Research Office is responsible for such contracts and is available to assist and answer questions.
- **Conflicts of Interest:** Please refer to the current version of the Postgraduate Handbook for information on conflict of interest. This section establishes a set of questions to help identify any potential conflicts. If a conflict is identified, the supervisory agreement should outline how this will be managed.
- **Confidential Material:** If the subject matter of the research is confidential and may require an embargo once completed refer to the current version of the Postgraduate Handbook for further information and the procedure. Your primary supervisor must supply a justification for confidentiality.
- **Intellectual Property:** Please refer to the current version of the Postgraduate Handbook. If you or your supervisor have reason to believe that Intellectual Property implications will arise out of independent research, your primary supervisor is responsible for discussing this with you and liaising with the University's commercialisation arm – AUT Enterprises Ltd.
- **Ethics Approval:** please seek advice from the Ethics Secretariat (AUTEK) if you have any queries on ethics requirements.
- **Postgraduate Handbook:** the postgraduate handbook is the overarching guide to postgraduate academic policies and procedures.

Confidential Material

Will you be requesting that your research be embargoed?
(If "yes", please include documents to support this request from your supervisor)

Yes **No**

Scholarships

Do you hold any scholarships/awards/sponsorship by an external organisation? **Yes** **No**

Are fees included in the scholarship? **Yes** **No**

Give the name(s) of scholarships/awards/sponsors German Academic Exchange Service Year Scholarship

Language

Do you wish to present your research in a language other than English?

Yes

No

If yes, state which language _____

DECLARATION BY APPLICANT

I declare that the information provided by me in this application is true and complete. I recognise that it is my responsibility to provide all necessary documentation to support my application and I authorise Auckland University of Technology, where necessary, to obtain further relevant documentation and to verify my qualifications as detailed in this application. I acknowledge that AUT reserves the right to vary or reverse any decision regarding admission to candidature on the basis of this application. I have read and understand the conditions of candidature outlined in the current Postgraduate Handbook and am prepared to accept them in full.

Applicant's signature: *Wendt*

Date: 4th October 2013

SUPERVISION

Primary Supervisor	_____	Supervised Master's to Completion	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Signature	<i>Wendt</i>	Date	4 th October 2013
Second Supervisor	_____	Supervised Master's to Completion	<input type="checkbox"/> Yes <input type="checkbox"/> No
Signature	_____	Date	_____
Additional Supervisor	_____	Supervised Master's to Completion	<input type="checkbox"/> Yes <input type="checkbox"/> No
Signature	_____	Date	_____
Capacity of Additional Supervisor	_____		

SCHOOL RESOURCES

HOD/HOS/DepChair/ Authorised staff member	<i>A. NARAYANAN</i>	Confirm Resources Available for this project	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Signature	<i>[Signature]</i>	Date	16/10/13

FACULTY POSTGRADUATE COMMITTEE ENDORSEMENT

Associate Dean (name) _____

Signature _____ Date _____

EMBARGO APPROVAL

This research has been approved as confidential and an embargo of months approved.

Associate Dean (name) _____

Signature _____ Date _____

RESEARCH PROPOSAL

Provide 2-3 pages containing a description of your proposed research. Use the following headings.

Title

An approach of managing backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment

Abstract/Summary: (100 words or fewer)

Cloud Computing is an essential part of the IT field with increasing popularity for private users and enterprises. While much literature is available from the users' point of view, only little research has been undertaken on how to actually provide such Cloud services. Using the example of one of the world's biggest Business Cloud providers, SAP, this thesis aims to fill the research gap specifically in the important area of effectively and efficiently backing up the customers' data with the added complexity of many and heterogeneous systems in place. A software-based solution is implemented and evaluated.

Literature/Past Research Review

When searching academic literature with the keyword "backup" in combination with "cloud", most results are helpful for companies that want to use cloud services for their daily business, from selecting the right service model over criteria which provider to choose up to how to negotiate security constraints (Heitmann, 2007; Zhonglin & Yuhua, 2011). In contrast, nothing was found that would guide the providers how to actually meet those demands. Therefore, literature on "traditional" backup strategies was consulted.

Two very good introductions into the topic of enterprise backups could be found by Preston (2007) and Little and Chapa (2003). They state the philosophy and business needs behind backing up company data, explain the backup process and also what can be expected for backup technology in the future. Furthermore, Preston included chapters on how to backup large file systems and databases. These books will be of great help in defining requirements, design, and evaluation criteria for the actual research.

A deeper look into the technology of backups yielded the term "snapshot". A snapshot is an exact copy of the system at a certain point in time and is seen to be a resource-efficient and reliable form of backing up data (Adams, 2001; Hunter, 2004). Due to their benefits and general acceptance in industry, snapshot technology is already used in SAP's datacentres. A detailed understanding of the underlying concepts of snapshots will, therefore, be of great help in implementing an appropriate solution.

When it comes to the efficient management of heterogeneous systems, scheduling tasks plays a big role. This is especially true for backups, but also for the general performance of systems. Qin and Jiang (2006) proposed a fault-tolerant algorithm to schedule precedence constrained tasks.

As some backups rely on others, and they can be seen as tasks in this context, the authors' findings must be taken into consideration when designing the new application.

Pamies-Juarez, García-López, Sánchez-Artigas and Herrera (2011) put their research focus on reducing data redundancy while retaining full data availability. Their theory is certainly noteworthy for the overall backup strategy that will be developed during the thesis. However, also their work mainly concentrates on using cloud services rather than providing them.

The literature that was found on the topic can roughly be divided into two categories, namely how to backup data, and how to manage heterogeneous systems and databases. What could not be found, even with several different queries like "heterogeneous backup", "cloud systems backup" or "backup of large number of heterogeneous systems", was literature that would directly contribute to the outcome of the research. This indicates that the chosen topic is of scientific relevance.

The novelty in the approach lies in the two major challenges within the project: heterogeneity and multitude, caused by acquisitions of other Cloud providers by SAP. As shown in the literature review, not much research has been conducted in this area before, which makes it a topic worth to investigate. Proposing a new method to overcome the identified challenges will make it also relevant for future researchers, as it is very likely that similar problems arise while cloud computing is more and more emerging.

Design/Plan of the Study:

As the research includes the creation of something new, it belongs to the area of action research or design science. The priority here is to point out the actual scientific value of the thesis (i.e., how it adds to the body of knowledge), and not let it become a sole software implementation. Therefore, the focus of the project lies on finding a beneficial solution for the problem in the given context, rather than on just adding all necessary methods to the programme. However, in order to easier evaluate the result with the actual users, it would be helpful if most of the tool could be developed. A major part of the evaluation will be the use in an external audit, in which SAP's cloud processes are tested for their compliance with certain industry standards.

Ethics

As the project will follow the software development lifecycle, in the beginning requirements will have to be gathered and at the end the tool will have to be evaluated. This will be done by interviewing business experts from the company who are working with the processes to be considered in the new solution. Therefore, ethical approval is required. The questions that will be asked will be exclusively about the interviewees' work tasks and responsibilities.

Resources and Budget

Proof-reading of the thesis \$500.

Location

The two main locations where the research will be conducted are AUT and SAP offices. Interviewing the business experts will mainly be done via Telepresence, a technology that could be described as HD video conferencing. As the interviewees are distributed around the world, this is the easiest way to access them. The technology is available in the SAP office in Auckland, to which the researcher has unrestricted access. Most likely, some part of the evaluation will also be done in person in the SAP office in Germany.

Timetable for Completion

As of now, the design specifies four major steps in the project: identify requirements, design a new solution, implement the solution, and evaluate it. The first step includes identifying the stakeholders of the project and interviewing them in order to get all requirements for the software. This step also includes an in-depth analysis of the problem together with literature review. Being the foundation for the whole project, this step must be done very carefully and will take two months, so that it should be finished by the beginning of May 2014.

The next step is the design of the software. This will involve structuring, analysing, and consolidating the gathered data in order to create the best possible solution fulfilling all necessities. This will likely take a month so it is finished by the beginning of June 2014.

Then the implementation will take place. Including the buffer that should always be considered in software development, this phase will take three months, so that by the beginning of September it should be finished. In September there will be an audit where the tool should be used. This is the first big external evaluation. Before that, testing will be done to check the overall functioning of the tool. After that, a big evaluation will be conducted, taking into account what was found during the audit and basically comparing the requirements with the functionality of the software.

Using the notes taken during the practical work, finally everything will be written down by the end of February 2015.

References

- Adams, M. (2001). How snapshot technology will change the future of backup and recovery. *Computer Technology Review*, 21(1), 26-27.
- Heitmann, R. (2007). Back up the virtual machine. *Communications News*, 44(10), 30-30,32.
- Hunter, P. (2004). Better Backup and Recovery with Snapshots. *Computer Technology Review*, 24(10), 34.
- Little, D. B., & Chapa, D. A. (2003). *Implementing backup and recovery: the readiness guide for the enterprise*. New York: Wiley.
- Pamies-Juarez, L., García-López, P., Sánchez-Artigas, M., & Herrera, B. (2011). Towards the design of optimal data redundancy schemes for heterogeneous cloud storage infrastructures. *Computer Networks*, 55(5), 1100-1113. doi:<http://dx.doi.org/10.1016/j.comnet.2010.11.004>
- Preston, W. C. (2007). *Backup and recovery*. Beijing: O'Reilly.
- Qin, X., & Jiang, H. (2006). A novel fault-tolerant scheduling algorithm for precedence constrained tasks in real-time heterogeneous systems. *Parallel Computing*, 32(5-6), 331-356. doi:<http://dx.doi.org/10.1016/j.parco.2006.06.006>
- Zhonglin, H., & Yuhua, H. (2011). A Study on Cloud Backup Technology and Its Development. In M. Dai (Ed.), *Innovative Computing and Information* (Vol. 231, pp. 1-7): Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-23993-9_1. doi:10.1007/978-3-642-23993-9_1

memo

School of Computing and Mathematical Sciences

To: Shoba Tegginmath, Postgraduate Programme Leader (MCIS)
From: Krassie Petrova, Senior Research Lecturer
CC: Anne Wendt, Postgraduate Student (MCIS)
Date: 10/8/2013
Re: MCIS Thesis Embargo

Reason The thesis needs to be embargoed for the maximum allowed period as the work and the expected outcomes are commercially sensitive.

As part of the thesis work a software artefact will be developed with the support of and in cooperation of an international company (SAP). The product of the development (software) will be tested with the SAP environment and it is hoped that it will be adopted and used by SAP in their internal daily processes, providing an opportunity for the company to gain competitive advantage. Therefore, it is necessary to put an embargo on the thesis in order to deny SAP's competitors early access to the thesis findings and prevent these competitors using the analysis and design to their benefit.

APPENDIX B EMBARGO APPROVAL



FORM PGR16 APPLICATION FOR RESTRICTED ACCESS TO A THESIS/DISSERTATION/EXEGESIS

PLEASE NOTE

- This form must be typed. Handwritten forms will not be accepted.
- Double clicking on the check boxes enables you to change them from not-checked to checked.
- The completed form, signed by the student and the primary supervisor, should be submitted to the appropriate Faculty Postgraduate Office when the thesis/exegesis is lodged for examination. If the application is approved by the Faculty Postgraduate Committee, the form will be signed by the Dean and sent to the University Postgraduate Centre for insertion into the print copies deposited. For more information consult the Postgraduate Handbook.

Student ID No	1386235	Name	Anne Wendt
Faculty	DCT	School/Dept	SCMS
Programme	MCIS	Date of submission for examination	Expected submission date Feb 2015 or earlier
Research Output	Thesis <input checked="" type="checkbox"/> Dissertation <input type="checkbox"/> Exegesis <input type="checkbox"/>	Points Value	120
Thesis Title	An approach of managing backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment		

EMBARGO TIMEFRAME

An embargo is requested on the public availability of the print and digital copies of the above thesis/exegesis from the date of submission for examination (maximum normally 36).

24 months

EMBARGO CATEGORIES

The thesis/dissertation/exegesis contains confidential or sensitive information which if publicly available may (Tick all that apply)

- Jeopardise the future intellectual property rights of the author (e.g. a patent application or publication)
- Breach a prior contractual arrangement with an external organisation (Please attach a copy of the relevant agreement(s))
- Infringe or endanger the right to privacy or cultural respect of an individual or group

The embargo would apply to

- The complete thesis/dissertation/exegesis
- A portion of the work (specify) : _____

SIGNATURES

Student		Date	17/10/2013
Primary Supervisor		Date	17/10/2013
Secondary Supervisor	_____	Date	_____
Additional Supervisor	_____	Date	_____

RESTRICTED ACCESS APPROVED BY FACULTY DEAN (or delegate)

Signature		Date	_____
OFFICE USE	RELEASE DATE	Digitally signed by Dr Rosser Johnson DN: cn=Dr Rosser Johnson, o=AUT, ou=DCT Faculty Office, email=rjohnson@aut.ac.nz, c=NZ Date: 2013.10.18 14:49:34 +13'00'	

APPENDIX C ETHICAL APPROVAL



24 February 2014

Krassie Petrova
Faculty of Design and Creative Technologies

Dear Krassie

Re Ethics Application: **13/315 An approach of monitoring backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment.**

Thank you for providing evidence as requested, which satisfies the points raised by the AUT University Ethics Committee (AUTC).

Your ethics application has been approved for three years until 24 February 2017.

As part of the ethics approval process, you are required to submit the following to AUTC:

- A brief annual progress report using form EA2, which is available online through <http://www.aut.ac.nz/researchethics>. When necessary this form may also be used to request an extension of the approval at least one month prior to its expiry on 24 February 2017;
- A brief report on the status of the project using form EA3, which is available online through <http://www.aut.ac.nz/researchethics>. This report is to be submitted either when the approval expires on 24 February 2017 or on completion of the project.

It is a condition of approval that AUTC is notified of any adverse events or if the research does not commence. AUTC approval needs to be sought for any alteration to the research, including any alteration of or addition to any documents that are provided to participants. You are responsible for ensuring that research undertaken under this approval occurs within the parameters outlined in the approved application.

AUTC grants ethical approval only. If you require management approval from an institution or organisation for your research, then you will need to obtain this. If your research is undertaken within a jurisdiction outside New Zealand, you will need to make the arrangements necessary to meet the legal and ethical requirements that apply there.

To enable us to provide you with efficient service, please use the application number and study title in all correspondence with us. If you have any enquiries about this application, or anything else, please do contact us at ethics@aut.ac.nz.

All the very best with your research,

A handwritten signature in black ink, appearing to read 'K O'Connor'.

Kate O'Connor

Executive Secretary

Auckland University of Technology Ethics Committee

Cc: Anne Wendt jrw4569@aut.ac.nz

APPENDIX D PARTICIPANT INFORMATION SHEET

Participant Information Sheet

**Date Information Sheet Produced:**

15 January 2014

Project Title

An approach of monitoring backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment

An Invitation

My name is Anne Wendt, a Master's student at Auckland University of Technology (AUT) in Auckland, New Zealand. I am also an employee of SAP (D053392) in the area of Cloud and responsible for the backup reports for Business byDesign and related systems. Earlier I developed a tool in GMP to create these reports automatically. One of the outcomes of my thesis which I am undertaking in cooperation with our employer (SAP) will be an extension of this tool – an application that facilitates the day-to-day management of the cloud customer data backup. I would like to invite to you to participate in this project and provide information about the processes you are working with. This information will contribute to the functional design of the software. Your participation in this project is voluntarily and you may withdraw at any time prior to the completion of the data collection. Your support will be extremely valuable and highly appreciated.

What is the purpose of this research?

The research is undertaken as part of the requirements to complete my degree (Master of Computer and Information Sciences). The research objective is to develop and evaluate a tool that may help cloud service providers meet the expectations of customers in terms of data security and backup. One of the anticipated project outcomes is a software product tool; the findings of the research will be presented in my Master's thesis and possibly in a related academic publication.

How was I identified and why am I being invited to participate in this research?

You were invited as a potential participant in the research because you are knowledgeable about managing cloud systems, particularly in the area of backing up the customer data. Your expertise in this area is highly valuable for my research. You were identified as a potential participant by my professional supervisor Heinrich Cevajka (D051541; Security Compliance Officer) based on your professional experience and involvement with the company's systems. Your contact data were obtained from the corporate address book. A total 40 participants were invited; the first 20 to accept the invitation will be recruited as research participants. Neither the receptionist nor my professional supervisor will know who has responded to the invitation.

What will happen in this research?

The project involves the development of a software tool. Your support will be needed in identifying requirements for the software based on your concrete knowledge and understanding of the backup management processes. Requirements data will be gathered during an interview of approximately one hour length which will be conducted at the beginning of the project; during the interview I will ask you about your expectations towards the tool. Your responses will be recorded and later transcribed by me. You will be able to receive a copy of your interview transcript and review it if you wish. I may contact you again via phone or email in order to clarify a point that you have made if during the process of transcribing and/or analysing the interview data something you have said is unclear to me. After the software is developed you will be asked to conduct a brief test of the tool and then tell me if it meets your expectations. The test and the evaluation will take approximately one hour

What are the discomforts and risks?

There is a limited risk of potential participant identity inadvertently being revealed within the organisation. While every effort to maintain the confidentiality of participants will be made, the pool of potential participants is small, and as a result only a limited confidentiality may be offered. However only the researcher (me) and my AUT project supervisor (Krassie Petrova) will know who actually participated.

What are the benefits?

The successful completion of the project will enable me to complete my academic qualification. For participants a potential benefit is the fact that their knowledge and expertise would have informed the design of a software product that may be adopted in the future as part of the company's set of operational tools.

How will my privacy be protected?

No other participant and none of their respective managers will know the identities of the actual and potential participants. The data record resulting from the interview and the product evaluation will be available only to me and my thesis supervisor. Furthermore, your name will not be part of any permanently stored research data. The only background information that will be collected is your job title, the type of the application you work with, and your main responsibilities. As all questions that will be asked will be solely related to your work knowledge and expertise, some of the information you will give to me may have been already given to other participants in the course of your routine work (as all participants and the researcher work for the same company). In order to limit the possibility of inadvertently revealing your identity in any written reports, direct data excerpts will not be used in any reports related to the research, or if needed to be used you will be asked for a permission to do so. Finally, information about participant expertise will be presented in summarised form.

What are the costs of participating in this research?

There will be no financial costs involved for you or your department as we will use the Telepresence rooms. Scheduling of the interviews will be done according to your availability to minimise any disruption in your daily work. It is estimated that you will need to spend approximately two hours in total, including answering questions and testing the tool.

What opportunity do I have to consider this invitation?

You will have the opportunity to consider the invitation while it is open (five working days after the day the invitation was sent out). If you have any questions about the research project please contact the researcher, or her AUT project supervisor (contact details provided in the last section of this document).

Your participation is voluntary. You will be able to withdraw during the study and up to the completion of data collection without any adverse consequences of any kind; all relevant records already made will be destroyed.

How do I agree to participate in this research?

You agree to participate in this research by contacting Anne by email (anne.wendt@sap.com), or by phone (+64 223 919 063 - New Zealand, +49 173 459 1135 - Germany), or by using corporate Lync, and by completing and signing the consent form. The completed and signed consent form will need to be emailed to Anne prior to commencing your participation.

Will I receive feedback on the results of this research?

You can indicate on the consent form whether you would wish to receive a copy of the report once completed (once examined and finalised the thesis will be made available to the public within the AUT online space).

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the project supervisor, Krassie Petrova krassie.petrova@aut.ac.nz, +64 9 921 9999 x 5045, or +64-021-906-794 (New Zealand).

Concerns regarding the conduct of the research should be notified to the Executive Secretary of AUTEK, Kate O'Connor, ethics@aut.ac.nz, +64 9 921 9999 ext. 6038 (New Zealand).

Whom do I contact for further information about this research?

Researcher Contact Details:

Anne Wendt

anne.wendt@sap.com or jrw4569@aut.ac.nz

Project Supervisor Contact Details:

Krassie Petrova

krassie.petrova@aut.ac.nz

+64 9 921 9999 x 5045 (New Zealand)

Approved by the Auckland University of Technology Ethics Committee on 24th February 2014, AUTEK
Reference number 13/315.

APPENDIX E PARTICIPANT CONSENT FORM

Consent Form



Project title: An approach of monitoring backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment

Project Supervisor: Krassie Petrova

Researcher: Anne Wendt

- I have read and understood the information provided about this research project in the Information Sheet dated 15 January 2014.
I have had an opportunity to ask questions and to have them answered.
I understand that notes will be taken during the interviews and that they will also be audio-taped and transcribed.
I understand that I may withdraw myself or any information that I have provided for this project at any time prior to completion of data collection, without being disadvantaged in any way.
If I withdraw, I understand that all relevant information including tapes and transcripts, or parts thereof, will be destroyed.
I agree to take part in this research.
I wish to receive a copy of the report from the research (please tick one): Yes No

Participant's signature:

Participant's name:

Participant's Contact Details (if appropriate):

.....
.....
.....
.....

Date:

Approved by the Auckland University of Technology Ethics Committee on 24th February 2014, AUTEK Reference number 13/315.

Note: The Participant should retain a copy of this form.

APPENDIX F INVITATION LETTER

Subject line: Invitation to participate in a research project in managing cloud data backup conducted by Anne Wendt

Text:

Dear [*title and name*],

On behalf of Anne Wendt – the researcher involved in the project mentioned in the subject line, titled “*An approach of monitoring backups and their properties for a vast number of heterogeneous systems in a business cloud computing environment*” I would like to invite you to participate in Anne’s research and help her with gathering requirements for the software product that will be built, and also testing and evaluating it.

Anne is a part time SAP employee, currently located at SAP New Zealand, in Auckland. She is conducting this research project as part of her work on completing her Master of Computer and Information Sciences degree at the Auckland University of Technology (AUT).

The project is fully supported by SAP. This invitation was sent to a large number of potential participants identified by Heinrich Cevajka (D051541), Security Compliance Officer. However, only Anne and her AUT supervisor (Krassie Petrova, Krassie.petrova@aut.ac.nz) will be aware of the identities of those who have volunteered to participate.

You can find more information about the project, and about the way participants were identified and selected as well as about and SAP’s involvement and the nature of your participation in the enclosed participant information sheet and consent form. It is hoped that the software product developed as a result of the research project may become part of the SAP’s set of operational tools in the future.

Please consider the invitation and contact Anne directly (anne.wendt@sap.com) within the next five working days if you would like to participate in her research project. Otherwise, please ignore the invitation and dispose of this email and the attachments. Please also contact Anne and/or her AUT supervisor if you have questions about the research (full contact details are provided on the information sheet).

With regards,

Receptionist, SAP New Zealand (Auckland)

APPENDIX G INTERVIEW NOTES

This appendix contains the notes that were taken during the interviews. At the beginning of each section, a short overview is given about the interviewee, including their position in the company and the application they work with. The following questions served as a guide throughout the interviews. However, they were mostly unstructured, and not all questions were answered by all participants, which is why the notes do not necessarily follow the question structure.

1. What is your job title?
2. With which application do you work?
3. What are your main responsibilities?
4. What does your process do and what is the aim of the process?
5. To which other processes is it connected and to what extent?
6. What are your responsibilities in the process?
7. What are the main problems you are facing in doing your daily work?
8. Which functionalities should the software tool have?

PARTICIPANT 1

Job title	Cloud Security Officer
Cloud products	Business byDesign und ByD-like products
Main responsibilities	Making sure all ByD-related operational processes abide by the Audit-relevant standards

- Current recipient of the reporting
- Provide easy, reproducible, trustworthy evidence for the Audits so that he can focus on other things in his job
- Include an easy way to find the basic population (= all systems that are productive and, therefore, in Audit scope)
- Filter by technical landscape (which usually represents the datacentre)
 - Also display landscape in table of systems
 - For easier pattern recognition (“is there an issue that only appears in one location or for one type/version of system?”)
- Show current status of system’s backup
 - Backup is running fine again (green)
 - Last backup of that system finished with warning (orange/yellow)

- Backup for that system is currently running (blue)
- Backup does not exist (red)
- System is now out of scope (turquoise)
- System was decommissioned (grey)
- Function to show a list of customers/tenants that were affected by the failure

PARTICIPANT 2

Job title	Senior Support Engineer
Cloud products	Business byDesign and ByD-like products
Main responsibilities	Management and operations of VLAB-Landscape, setting up Zabbix monitoring, cloud operations platform services and monitoring, infrastructure (server & storage, network) L2 operations

- Check policies for different products in SAP and make sure that they're compliant
- Include check for filer storage
 - At the moment they go to vfiler objects in GMP to check it
 - Each backup is kept on a filer in primary storage for 3 days
 - Then it is transferred to secondary storage automatically where it will stay for another 15 days (called "nearstore" in GMP)
 - Sometimes this goes wrong but they have no way to find out except for checking manually → alert function needed
 - A reporting about this would also be needed as it is part of their SLA's
 - It would be helpful to have "storage" as a filter category (primary/nearstore)
- Include other ByD-like products/HANA Apps
 - Those are "small" products that usually do not have many systems
 - These systems should be included in the reporting
- new system types should be added automatically so that there is no delay from manual uploading
- Find out failure reasons for backups automatically from GMP so that only the actual fix will be manual
 - There is an error code that says what is wrong
 - That could be used to provide more information
 - Referred the researcher to participant 4 who could provide more information on error codes

- Current issue: report should just check the best backup if there are two defined
 - Issues during migration from MaxDB to HANA
 - Sometimes during the migration there are two active backups
 - So maybe preferably check HANA backups
 - Check for multiple backups and use the one that's working (not inactive)
- It would be helpful to have an interface to configure automatic reports, frequency with which they're sent out, and recipient list
 - Interface for authorised people
 - Alter frequency (weekly, monthly, semi-annually, annually) and scope (query for systems that shall be included)
 - They would need a 6-month reporting
 - Offer a pre-generated grouping function that automatically selects a bunch of checkboxes (specified beforehand by user)
- Change wording to make 3-time-rule clearer
- Find out the total number of alerts
 - At the moment they use the other report to count backup failures and then report on that
 - Use reporting of participant 4
- Other cloud management systems
 - Zabbix widely used for monitoring SuccessFactors products and infrastructure
 - Rota manages SF backups so there is not backup reporting needed for those systems

PARTICIPANT 3

Job title	Technical employee
Cloud products	SuccessFactors applications
Main responsibilities	Ensuring the backup process runs according to Audit standards

- Most SuccessFactors systems are managed by GMP
 - There is another big group (mainly SuccessFactors' own datacentres) in which the backup is managed by an external company
 - To find out about those backups we would have to contact them
 - This means only part of their systems needs to be integrated in the reporting at the moment, but the other systems might follow

- Would like to integrate systems from newly opened DC12 (Amsterdam) into the reporting
- Don't use SID → 6-digit CCMS-SID instead!
 - Standard systems end with STD
 - Premium systems end with PRE
- They get number of backup failures as weekly report
- Whole database is backed up as a whole instead of system-wise
- When providing evidence for auditors, they search for a customer name to find the associated databases and then check the backups
- Currently using detailed backup alerts reporting (#2) with filters

PARTICIPANT 4

Job title	Software Developer
Cloud products	Business byDesign and ByD-like products
Main responsibilities	Development of new functionality for GMP in the area of backup

- Current backup tools in GMP include set up, schedule, delete, restart in case of failure, and reporting
- All entities of GMP can be found under Inventory using the search function
- In case of a failure an alert is created that is sent to a group of responsible people who then restart the backup
- Alerts are collected and stored in a table, and can be found in report #2 (detailed alert reporting)
- Error codes are created automatically based on what the database status is
 - there are only 4 types of codes
 - they are very general
 - so probably I won't be able to get sufficient information out of them
- When a backup fails it is added to a status table together with error code and error message
- Current code structure of backup reporting is quite messy and has to be updated to abide by the most recent development guidelines for GMP (he'll send me the link to the coding guidelines via email)
- It will have to become a separate Perl module with re-worked UI

- There is another alert reporting that he wrote and that should be integrated into my reporting (he'll send me the link via email, too)
- When I want to upload code to GMP I have to go via SAP GitHub (again link with instructions will be in email) and follow the ordinary process
- He is happy to have a look at my code and the final tool to give feedback

PARTICIPANT 5

Job title	Senior Support Engineer
Cloud products	NetWeaver Cloud, JPaaS
Main responsibilities	Ensuring audit compliance of JPaaS systems

- Until a few weeks ago, virtual machines, systems, and backups were requested via CSS tickets
 - Then a support employee would create them manually
 - GMP was a black box for them because they didn't have to use it for their daily work
- Now there is an API to create a new system or backup
 - Backups are scheduled automatically based on their needs
 - Systems get ZH-code ZH998
 - However, this is not part of the participant's responsibility
- Too many irrelevant systems are collected by the report at the moment
 - Filter criteria must be changed
 - Exclude systems from landscapes ROT4, ROT5, ROT6, development, and VLAB
 - Include systems from landscapes ASH (Ashburton), SYD (Sydney), ROT3
 - Use CCMS-SID instead of 3-digit SID, because their systems use longer SID's
- They backup per database, so it would make sense to group the actual reporting per landscape
- Keep an option to report over non-audit relevant systems just for internal checks
- Show technical landscape, resource pool, and installation template for easier pattern recognition in list of failed backups

 PARTICIPANT 6

Job title	Operations and Expertise – General Management and Admin
Cloud products	Business byDesign and ByD-like products
Main responsibilities	Creating reports from different tools, administrative tasks

- user of the current tool but also of SISM, TIC, SPC
- sends out weekly and monthly backup reporting, among others
- would be happy if that can be automated so that she doesn't have to send it anymore
- tool should have automatic feedback function so that issues or requests for including new system types are not addressed to her anymore, but rather to me or to the developer who is currently responsible for the tool
- maybe functionality that marks/unmarks all checkboxes

 PARTICIPANT 7

Job title	IT Business Services Principal Consultant
Cloud products	HANA Enterprise Cloud, SuccessFactors Applications, Ariba, Business by Design
Main responsibilities	Overall SAP Cloud Compliance Coordination

- Attempt to integrate ByD into SuccessFactors compliance framework was reversed (ByD and SuccessFactors operate separate compliance frameworks again in 2014)
- The partly huge differences between the environments (people, processes, tools) make the implementation of the “One Delivery” strategy a complex task. The overall roadmap of the work package is to finalize the control framework harmonization in 2016. Nevertheless the approach will be phased and aligned to the audit cycles. First synergy effects of the harmonization were realized in spring 2014.
- Overall the SAP Cloud compliance frameworks will be harmonized
 - First harmonization of HEC and ByD has been implemented for SOC 2 audits in spring 2014
 - Further harmonization will take place also including Ariba and SuccessFactors, who are using a different process at the moment
 - Common infrastructure services shall be audited on a corporate base (using a shared control provider as virtual organisation)

- As part of the harmonization SAP is striving for a unified backup management and reporting across all landscapes and applications
- However, for that we would need a common technical base, which does not exist at the moment, as current solutions are too different
- Mainly used system for HEC: SISM (SAP in-house system manager)
 - All systems are stored there with relevant information
 - Flag that marks if a system is backup-relevant or not (set automatically depending on other features like status of the system)
 - Back-up are done for systems in status “Quality check in progress” and “live”
 - KUG = key user group
- Suggested to invite participant 8 for my research if I would like some more specific information about backups of HEC systems

PARTICIPANT 8

Job title	IT Technology Senior Consultant Global IT Backup Management
Cloud products	HANA Enterprise Cloud
Main responsibilities	Ensuring that backup and restore process is running properly

- Main tool that was centrally decided to be used for all their operations: SISM (SAP in-house system manager)
- Additionally other tools are used that have to be connected to the SISM via Excel-exported files due to incompatibility with SAP systems
- HEC backups are replicated between Rot and Amsterdam (as defined in control no. 5 in their risk control matrix)
- Audit-relevant list of productive customer systems (basic population) can be found by filtering
 - KUG (key user group) = HEC External Business
 - Backup type = backup relevant (checkbox marked)
 - Status = live (as opposed to “build-up” and “quality check in progress” → however, when a system is set to “quality check in progress”, a CSS ticket is automatically created that requests the creation of a backup)
 - One customer can have several systems, but customers never share systems
- Almost everything is automated

- Creation of backup is automatically requested via CSS on status change of the system
- Automatic backup if amount of changed data since last backup is greater than 5GB
- All user activity is logged → even if new transactions happened in the systems since the last backup was done, the status at time of a crash can be re-created
- List of backup alert history
- Backups are done in daily intervals and kept for 30 days
- Step-by-step restore request tool to restore a system
 - Only a small group of pre-defined people can actually execute a restore
 - Keeping this list updated is also an Audit control
- New reporting is currently added that can show backup statistics based on geographical region and per customer
- Current reporting and other backup-related functionalities are much more detailed than for any other cloud product at the moment (caused by different requirements of departments and auditors), which makes unification a challenge

APPENDIX H MAPPING INTERVIEW NOTES INTO REQUIREMENTS

PARTICIPANT 1

Note	Requirement	No
Current recipient of the reporting	No requirement	-
Provide easy, reproducible, trustworthy evidence for the Audits so that he can focus on other things in his job	Easy → usability	26
	Reproducible → historicisation in tables and log files	27
	Trustworthy → data sources can only be manipulated by the script, not by people	28
Include an easy way to find the basic population (= all systems that are productive and, therefore, in Audit scope)	Show scheduled systems for a given scope (note: this is already implemented by the “show scheduled systems” button → make it more obvious)	16
Filter by technical landscape (which usually represents the datacentre) <ul style="list-style-type: none"> • Also display landscape in table of systems • For easier pattern recognition (“is there an issue that only appears in one location or for one type/version of system?”) 	Filter by technical landscape	1
	Add landscape to table as a column	29
Show current status of system’s backup <ul style="list-style-type: none"> • Backup is running fine again (green) • Last backup of that system finished with warning (orange/yellow) • Backup for that system is currently running (blue) • Backup does not exist (red) • System is now out of scope (turquoise) • System was decommissioned (grey) 	Display little coloured dot next to system in “failed systems” table	9
Function to show a list of customers/tenants that were affected by the failure	From the list of failures, link to the list and number of customers affected	2

PARTICIPANT 2

Note	Requirement	No
Check policies for different products in SAP and make sure that they're compliant	Automatically include policies into the reporting	17
<p>Include check for filer storage</p> <ul style="list-style-type: none"> At the moment they go to vfiler objects in GMP to check it Each backup is kept on a filer in primary storage for 3 days Then it is transferred to secondary storage automatically where it will stay for another 15 days (called "nearstore" in GMP) Sometimes this goes wrong but they have no way to find out except for checking manually → alert function needed A reporting about this would also be needed as it is part of their SLA's It would be helpful to have "storage" as a filter category (primary/nearstore) 	<p>New feature: check for filer storage</p> <p>Alert function to notify if the transfer was not successful</p> <p>Monitoring over how successful the transfers were in a given time period</p> <p>Filter per filer storage</p>	<p>3</p> <p>3</p> <p>3</p> <p>30</p>
<p>Include other ByD-like products/HANA Apps</p> <ul style="list-style-type: none"> Those are "small" products that usually do not have many systems These systems should be included in the reporting 	<p>Add more systems</p> <p>Let users configure automatic reports</p>	<p>4</p> <p>10</p>
<p>Find out failure reasons for backups automatically from GMP so that only the actual fix will be manual</p> <ul style="list-style-type: none"> There is an error code that says what is wrong That could be used to provide more information Referred the researcher to participant 4 who could provide more information on error codes 	<p>Display failure reason</p> <p>(Error codes are not as helpful as the participant thought)</p>	12
<p>Current issue: report should just check the active backup if there are two defined</p> <ul style="list-style-type: none"> Issues during migration from MaxDB to HANA Sometimes during the migration 	<p>Only take "best backup" into consideration to avoid false negatives</p>	5

<p>there are two active backups</p> <ul style="list-style-type: none"> • So maybe preferably check HANA backups • Check for multiple backups and use the one that's working (not inactive) 		
<p>It would be helpful to have an interface to configure automatic reports, frequency with which they're sent out, and recipient list</p> <ul style="list-style-type: none"> • Interface for authorised people • Alter frequency (weekly, monthly, semi-annually, annually) and scope (query for systems that shall be included) • They would need a 6-month reporting • Offer a pre-generated grouping function that automatically selects a bunch of checkboxes (specified beforehand by user) 	<p>Let users configure automatic reports for a specific scope in a specific time frame that is sent out automatically to a specific group of people</p> <p>Let the user save a set of their preferably selected checkboxes</p>	<p>10</p> <p>20</p>
<p>Change wording to make 3-time-rule clearer</p>	<p>Change wording for 3-day-rule</p>	<p>31</p>
<p>Find out the total number of alerts</p> <ul style="list-style-type: none"> • At the moment they use the other report to count backup failures and then report on that • Use reporting of participant 4 	<p>Display total number of alerts for the given scope</p>	<p>18</p>
<p>Other cloud management systems</p> <ul style="list-style-type: none"> • Zabbix widely used for monitoring SuccessFactors products and infrastructure • Rota manages SF backups so there is not backup reporting needed for those systems 	<p>No requirement</p>	<p>-</p>

PARTICIPANT 3

Note	Requirement	No
<p>Most SuccessFactors systems are managed by GMP</p> <p>There is another big group (mainly SuccessFactors' own datacentres) in which the backup is managed by an external</p>	<p>No direct requirement. However, this means that not all systems can be included at the moment → concept portability</p>	<p>22</p>

company To find out about those backups we would have to contact them This means only part of their systems needs to be integrated in the reporting at the moment, but the other systems might follow		
Would like to integrate systems from newly opened DC12 (Amsterdam) into the reporting	Add new systems Let users configure automated reports	4 10
Don't use SID → 6-digit CCMS-SID instead! Standard systems end with STD Premium systems end with PRE	Use CCMS_SID instead of SID	23
They get number of backup failures as weekly report	No requirement	-
Whole database is backed up as a whole instead of system-wise	Offer scope flexibility	24
When providing evidence for auditors, they search for a customer name to find the associated databases and then check the backups	No requirement	-
Currently using detailed backup alerts reporting (#2) with filters	No requirement	-

PARTICIPANT 4

Note	Requirement	No
Current backup tools in GMP include set up, schedule, delete, restart in case of failure, and reporting	No requirement	-
All entities of GMP can be found under Inventory using the search function	No requirement	-
In case of a failure an alert is created that is sent to a group of responsible people who then restart the backup	No requirement	-
Alerts are collected and stored in a table, and can be found in report #2 (detailed alert reporting)	No requirement	-
Error codes are created automatically based on what the database status is	No requirement. Feeds into requirement 12	-
When a backup fails it is added to a status table together with error code and error	No requirement. Feeds into requirement 12	-

message		
Current code structure of backup reporting is quite messy and has to be updated to abide by the most recent development guidelines for GMP (he'll send me the link to the coding guidelines via email)	Follow development guidelines Readable and maintainable code	33 25
It will have to become a separate Perl module with re-worked UI	Follow development guidelines Readable and maintainable code	33 25
There is another alert reporting that he wrote and that should be integrated into my reporting (he'll send me the link via email, too)	Include alert reporting	34
When I want to upload code to GMP I have to go via SAP GitHub (again link with instructions will be in email) and follow the ordinary process	Follow development guidelines	33
He is happy to have a look at my code and the final tool to give feedback	No requirement. However, this implies readable and maintainable code	25

PARTICIPANT 5

Note	Requirement	No
<p>Until a few weeks ago, virtual machines, systems, and backups were requested via CSS tickets</p> <ul style="list-style-type: none"> • Then a support employee would create them manually • GMP was a black box for them 	Easy usage for new users (usability)	26
<p>Now there is an API to create a new system or backup</p> <ul style="list-style-type: none"> • Backups are scheduled automatically based on their needs • Systems get ZH-code ZH998 • However, this is not part of the participant's responsibility 	No requirement	-
<p>Too many irrelevant systems are collected by the report at the moment</p> <ul style="list-style-type: none"> • Filter criteria must be changed • Exclude systems from landscapes ROT4, ROT5, ROT6, development, and VLAB • Include systems from landscapes ASH (Ashburton), SYD (Sydney), ROT3 	<p>Include new/other systems</p> <p>Let users configure reports</p> <p>Include new/other systems</p>	<p>4</p> <p>10</p> <p>4</p>

<ul style="list-style-type: none"> Use CCMS-SID instead of 3-digit SID, because their systems use longer SID's 	Use CCMS_SID instead of SID	23
They backup per database, so it would make sense to group the actual reporting per landscape	Offer scope flexibility	24
Keep an option to report over non-audit relevant systems just for internal checks	Report over non-audit relevant systems	6
Show technical landscape, resource pool, and installation template for easier pattern recognition in list of failed backups	Show more system details	13

PARTICIPANT 6

Note	Requirement	No
user of the current tool	No requirement	-
sends out weekly and monthly backup reporting, among others	Good performance	35
would be happy if that can be automated so that she doesn't have to send it anymore	Send out reports automatically	11
tool should have automatic feedback function so that issues or requests for including new system types are not addressed to her anymore, but rather to the researcher or to the developer who is currently responsible for the tool	Automatic feedback function	19
maybe functionality that marks/unmarks all checkboxes	Mark/unmark all checkboxes. This could feed into requirement 20	36

PARTICIPANT 7

Note	Requirement	No
Attempt to integrate ByD into SuccessFactors compliance framework was reversed (ByD and SuccessFactors operate separate compliance frameworks again in 2014)	No requirement. However, this is a very interesting finding that could feed into requirements 22 and 4	-
The partly huge differences between the environments (people, processes, tools) make the implementation of the "One Delivery" strategy a complex task. The overall roadmap of the work package is to	No requirement. However, this is a very interesting finding that could feed into requirements 22 and 4	-

finalize the control framework harmonization in 2016. Nevertheless the approach will be phased and aligned to the audit cycles. First synergy effects of the harmonization were realized in spring 2014.		
<p>Overall the SAP Cloud compliance frameworks will be harmonized</p> <ul style="list-style-type: none"> • First harmonization of HEC and ByD has been implemented for SOC 2 audits in spring 2014 • Further harmonization will take place also including Ariba and SuccessFactors, who are using a different process at the moment • Common infrastructure services shall be audited on a corporate base (using a shared control provider as virtual organisation) 	Concept portability	22
As part of the harmonization SAP is striving for a unified backup management and reporting across all landscapes and applications	Concept portability	22
However, for that we would need a common technical base, which does not exist at the moment, as current solutions are too different	Concept portability	22
<p>Mainly used system for HEC: SISM (SAP in-house system manager)</p> <ul style="list-style-type: none"> • All systems are stored there with relevant information • Flag that marks if a system is backup-relevant or not (set automatically depending on other features like status of the system) • Back-up are done for systems in status “Quality check in progress” and “live” • KUG = key user group 	No requirement	-
Suggested to invite participant 8 for my research if I would like some more specific information about backups of HEC systems	No requirement	-

PARTICIPANT 8

Note	Requirement	No
Main tool that was centrally decided to be used for all their operations: SISM (SAP in-house system manager)	Concept portability	22
Additionally other tools are used that have to be connected to the SISM via Excel-exported files due to incompatibility with SAP systems	No requirement	-
HEC backups are replicated between Rot and Amsterdam (as defined in control no. 5 in their risk control matrix)	No requirement	-
<p>Audit-relevant list of productive customer systems (basic population) can be found by filtering</p> <ul style="list-style-type: none"> • KUG (key user group) = HEC External Business • Backup type = backup relevant (checkbox marked) • Status = live (as opposed to “build-up” and “quality check in progress” → however, when a system is set to “quality check in progress”, a CSS ticket is automatically created that requests the creation of a backup) • One customer can have several systems, but customers never share systems 	No requirement. However, this could give some inspiration on how to create the new tool	-
<p>Almost everything is automated</p> <ul style="list-style-type: none"> • Creation of backup is automatically requested via CSS on status change of the system • Automatic backup if amount of changed data since last backup is greater than 5GB • All user activity is logged → even if new transactions happened in the systems since the last backup was done, the status at time of a crash can be re-created • List of backup alert history 	No requirement. However, this could give some inspiration on how to create the new tool	-
Backups are done in daily intervals and kept for 30 days	No requirement	-
Step-by-step restore request tool to restore a system	No requirement	-

<ul style="list-style-type: none"> • Only a small group of pre-defined people can actually execute a restore • Keeping this list updated is also an Audit control 		
<p>New reporting is currently added that can show backup statistics based on geographical region and per customer</p>	<p>No requirement. However, this could feed into extending requirement 10</p>	<p>-</p>
<p>Current reporting and other backup-related functionalities are much more detailed than for any other cloud product at the moment (caused by different requirements of departments and auditors), which makes unification a challenge</p>	<p>No requirement. However, this could feed into requirement 22</p>	<p>-</p>

APPENDIX I NOTES TAKEN DURING THE OPERATION OF THE TOOL

Note	Requirement	No
<p>There is a bug in the update function, causing backup errors to be reported even though there are no scheduled systems</p> <ul style="list-style-type: none"> • It does not take into consideration that a system can be upgraded • Any other features should also be checked to ensure the system is always categorised correctly • If a system has changed its status, this should be acknowledged in the reporting so that no false errors are reported 	<p>Update information on upgraded systems</p> <p>Show current system status</p>	<p>7</p> <p>9</p>
<p>If there is an error with the script, an email shall be sent to the researcher so that she can fix the problem as soon as possible</p>	<p>Send error notifications to the developer/ researcher</p>	<p>14</p>
<p>It should be possible to disable ZH012 and ZH014 because they are not used anymore</p> <ul style="list-style-type: none"> • This also applies to older system versions such as ByD 3.0 or 3.5 • There are no active systems with those properties anymore, however, they are still collected because the tables are joint • Making this more accurate would avoid the creation of unnecessary table entries 	<p>Take system categories out of scope</p>	<p>8</p>
<p>Time zone differences → schedule the collection script earlier in the morning because otherwise there might be an error with systems that were not checked for yesterday even though the user can select the timespan up to yesterday</p>	<p>Schedule collection script shortly after midnight German time</p>	<p>21</p>
<p>Find the person who is responsible for fixing the systems and tell them directly</p>	<p>Find responsible people</p> <p>Feeds into requirement 10</p>	<p>15</p>
<p>Add new systems more easily</p> <ul style="list-style-type: none"> • Automatically add new ByD versions • Allow users to configure system scope 	<p>Include new/other systems</p> <p>Automatically add new ByD versions</p> <p>Let users configure automatic reports</p>	<p>4</p> <p>32</p> <p>10</p>

APPENDIX J SOURCE CODE

BKP_REP_DAILY_DATA_COLLECTOR.PL

This script is run daily and gathers all data for the reporting.

```

1 #!/usr/bin/perl
2 # this script gets a list of all systems that must have a backup on
3 that day
4 # then it looks if those backups were successful (done for the day
5 before to make sure all backups could finish)
6 # $Id$
7
8 use strict;
9 use warnings;
10
11 use FindBin qw($Bin);
12 use lib "$Bin/./cgi-bin/lib";
13
14 use GMP::DateTime;
15 use GMP::DB;
16 use GMP::Inventory;
17
18 #####
19 ###  SETUP SECTION  ###
20 #####
21
22 # get the current date in the format yyyyymmdd
23 my $date = GMP::DateTime->get_datetime( undef, 1 )->{'date'};
24 $date =~ s/--//gxms;
25
26 # get today and yesterday in the format yyyyymmddhhmmss (database
27 format)
28 my $today      = GMP::DateTime->get_datetime( $date . '000000', 1 );
29 my $yesterday = GMP::DateTime->get_datetime( $today->{epoch} - 24 *
30 3600, 1 )->{'tst'};
31 $today = $today->{'tst'};
32
33 #####
34 ###  SCRIPT LOGIC  ###
35 #####
36
37 # check the script's last run date
38 my ( $lr_rows, $lr_content, $lr_message, $lr_sql ) =
39   GMP::DB->exec_sql(
40     { action => "SELECT",
41       fields => "VALUE",
42       tables => "ZBPA_GEN_CUST",
43       where  => "SCRIPT = 'bkp_rep_daily_data_collector.pl'
44                 AND KEY = 'LAST_RUN_DATE'",
45     }
46 );
47 my $last_run_date = $lr_content->[0]->{VALUE};
48
49 # if it ran today already we don't need further execution
50 die "Script bkp_rep_daily_data_collector.pl already ran today"
51 if $last_run_date == $today;
52
53 # get all queries that are active because we need that regardless of
54 which path we go
55 my ($q_rows, $q_content, $q_message, $q_sql) = GMP::DB->exec_sql( {

```

```

56     action => 'SELECT',
57     fields => 'QUERY_ID, QUERY',
58     tables => "ZPRC_BKP_QUERIES",
59     where  => "IS_ACTIVE = '1'",
60     options => 'ORDER BY QUERY_ID'
61 } );
62
63 # decide if we can do a normal run or if we need error handling
64 if ($last_run_date == $yesterday) {
65     _run_normally();
66 } else {
67     warn "Script bkp_rep_daily_data_collector.pl has not run since
68         $last_run_date";
69     _error_handling();
70 }
71
72 # if that ran through fine, update the last successful run date of the
73 script to today
74 my ($rows, $r_message, $r_content, $sql) = GMP::DB->exec_sql( {
75     action => 'UPDATE',
76     tables => 'ZBPA_GEN_CUST',
77     fields => [ [ [ 'VALUE', $today ] ] ],
78     where  => "SCRIPT = 'bkp_rep_daily_data_collector.pl'
79             AND KEY = 'LAST_RUN_DATE'",
80 } );
81
82 #####
83 ###     METHODS     ###
84 #####
85
86 sub _run_normally {
87
88     # it is better structured if we go by query
89     QUERY: foreach my $query (@$q_content) {
90         my $query_id = $query->{QUERY_ID};
91         my $query_lstql = $query->{QUERY};
92         my $fail_count = 0;
93
94         ### COLLECT BACKUP STATES ###
95
96         # get all systems that were live yesterday and, thus, needed a
97         backup
98         my ($ls_rows, $ls_content, $ls_message, $ls_sql) =
99             GMP::DB->exec_sql( {
100                 action => 'SELECT',
101                 fields => 'CCMS_SID, QUERY_ID',
102                 tables => "ZPRC_BKP_SCHED_SYS",
103                 where  => "QUERY_ID = $query_id AND VALID_TO = '$yesterday'"
104             }
105         );
106
107         # loop at the list of live systems and get their backup status
108         SYSTEM: foreach my $system (@$ls_content) {
109             my $ccms_sid = $system->{CCMS_SID};
110             my $query_id = $system->{QUERY_ID};
111
112             # get the DATA backups for that system from Inventory
113             my $lstql = "[ FROM 'Backup' WHERE 'Type' = 'DATA' ]
114                 AND [ FROM 'Backup' WHERE 'SAP Application' <=i2i=>
115                 [ FROM 'SAP System' WHERE 'CCMS SID' = '$ccms_sid' ] ] ";
116             my @backups = GMP::Inventory->list(
117                 "Backup", { lstql => $lstql } );
118
119             # check if this system has a backup

```

```

120 my $number_of_bkps = @backups;
121 if ($number_of_bkps == 0) {
122     $fail_count++;
123     next SYSTEM; # nothing else to check for this system if
124                 there was no backup at all
125 }
126
127 # says if a failed backup was found
128 my $has_backup_failed = 0;
129
130 foreach my $backup (@backups) {
131     # get the very last backup run for the given day (we
132     # assume that the last one was the successful one)
133     my $bkp_id = $backup->id;
134     my ($rows, $r_content, $r_message, $sql) =
135     GMP::DB->exec_sql( {
136         action => "SELECT",
137         fields => "BKP_RET_CODE, RETURN_STATUS",
138         tables => "ZPRC_BKP_STATUS",
139         where => "APP_ID = '$bkp_id'
140                 AND BKP_START_TIME < '$today'
141                 AND BKP_START_TIME >= '$yesterday'",
142         options => "ORDER BY BKP_END_TIME DESC LIMIT 1",
143     } );
144
145     # if we found a good backup we can go directly to the next
146     # system (and thus ignore $has_system_failed)
147     if ($rows and $r_content->[0]->{BKP_RET_CODE} == 0
148     and $r_content->[0]->{RETURN_STATUS} == 0) {
149         $has_backup_failed = 0;
150         next SYSTEM;
151     } else {
152         $has_backup_failed = 1;
153     }
154 } #loop at system's backup
155
156 # $has_system_failed will only be set if there was no
157 # successful backup (which would have started a new SYSTEM
158 # loop)
159 if ($has_backup_failed) {
160     $fail_count++;
161     # check if system has failed the day before
162     my $tmp = GMP::DateTime->get_datetime( $yesterday, 1 );
163     my $day_before_yesterday = GMP::DateTime->get_datetime(
164     $tmp->{epoch} - 24 * 3600, 1)->{'tst'};
165     my ($f_rows, $f_content, $f_message, $f_sql) =
166     GMP::DB->exec_sql( {
167         action => 'SELECT',
168         fields => 'COUNT',
169         tables => 'ZPRC_BKP_FAILED',
170         where => "CCMS_SID = '$ccms_sid'
171                 AND QUERY_ID = '$query_id'
172                 AND M_DATE = '$day_before_yesterday'"
173     } );
174     # add the system to the failed table (ZPRC_BKP_FAILED)
175     my $count = 1;
176     $count = $f_content->[0]->{COUNT} + 1 if $f_rows > 0;
177     my ($if_rows, $if_message, $if_content, $if_sql) =
178     GMP::DB->exec_sql( {
179         action => 'INSERT',
180         tables => 'ZPRC_BKP_FAILED',
181         fields => [[
182             [ "M_DATE", $yesterday ],
183             [ "CCMS_SID", $ccms_sid ],

```

```

184             [ "QUERY_ID", $query_id ],
185             [ "COUNT",    $count ],
186         ]],
187     } );
188 }
189 } #loop at systems
190
191 # now update the result table for yesterday
192 my ( $sir_rows, $sir_message, $sir_content, $sir_sql ) =
193 GMP::DB->exec_sql( {
194     action => 'UPDATE',
195     tables => 'ZPRC_BKP_RESULT',
196     fields => [ [ [ 'FAILED', $fail_count ] ] ],
197     where => "M_DATE = '$yesterday' AND QUERY_ID = '$query_id'"
198 }) if $fail_count > 0;
199
200 ### COLLECT ALL ACTIVE SYSTEMS ###
201
202 # get all systems that are live today
203 my @sys_list = GMP::Inventory->list( "SAP System",
204     { lstql => $query_lstql } );
205 my $system_count = @sys_list;
206
207 # check if there are any systems for that query or if we can
208 skip it
209 next QUERY if $system_count == 0;
210
211 # update result/overview table (ZPRC_BKP_RESULT)
212 ($sir_rows, $sir_message, $sir_content, $sir_sql) =
213 GMP::DB->exec_sql( {
214     action => 'INSERT',
215     tables => 'ZPRC_BKP_RESULT',
216     fields => [[
217         [ "M_DATE",    $today ],
218         [ "QUERY_ID",  $query_id ],
219         [ "SCHEDULED", $system_count ],
220         [ "FAILED",    0 ],
221     ]],
222 } );
223
224 # update list of scheduled systems (ZPRC_BKP_SCHED_SYS)
225 foreach my $system (@sys_list) {
226     my $ccms_sid = $system->ccms_sid();
227
228     # check if we already have an entry for that system from
229     yesterday that we can reuse
230     my ($sc_rows, $sc_content, $sc_message, $sc_sql) =
231     GMP::DB->exec_sql( {
232         action => 'SELECT',
233         fields => 'VALID_FROM', #what we get here doesn't matter
234             because we only need the number of rows
235         tables => 'ZPRC_BKP_SCHED_SYS',
236         where => "CCMS_SID = '$ccms_sid'
237             AND QUERY_ID = '$query_id'
238             AND VALID_TO = '$yesterday'",
239     } );
240
241     if ($sc_rows > 0) {
242         # just update the row
243         my ($u_rows, $u_message, $u_content, $u_sql) =
244         GMP::DB->exec_sql( {
245             action => 'UPDATE',
246             tables => 'ZPRC_BKP_SCHED_SYS',
247             fields => [ [ [ 'VALID_TO', $today ] ] ],

```

```

248         where => "CCMS_SID = '$ccms_sid'
249                AND QUERY_ID = '$query_id'
250                AND VALID_TO = '$yesterday'"
251     } );
252 } else {
253     # insert a new row
254     my ($is_rows, $is_message, $is_content, $is_sql) =
255     GMP::DB->exec_sql( {
256         action => 'INSERT',
257         tables => 'ZPRC_BKP_SCHED_SYS',
258         fields => [[
259             [ "CCMS_SID",    $ccms_sid ],
260             [ "QUERY_ID",    $query_id ],
261             [ "VALID_FROM",  $today ],
262             [ "VALID_TO",    $today ],
263         ]],
264     } );
265 }
266 } #loop at systems in a query
267 } #loop at queries
268
269 # TODO: insert code here to send out automatic reports
270
271 } #sub normal_run
272
273 sub _error_handling {
274
275     ### UPDATE LIST OF SCHEDULED SYSTEMS ###
276
277     # get list of systems that are currently active (by query)
278     foreach my $query (@$q_content) {
279         my $query_id = $query->{QUERY_ID};
280         my $query_lstql = $query->{QUERY};
281
282         my @sys_list = GMP::Inventory->list( "SAP System",
283                                             { lstql => $query_lstql } );
284
285         # check if they were active on the last run date
286         foreach my $system (@sys_list) {
287             my $ccms_sid = $system->ccms_sid();
288             my ($s_rows, $s_content, $s_message, $s_sql) =
289             GMP::DB->exec_sql( {
290                 action => 'SELECT',
291                 fields => 'VALID_FROM',
292                 tables => 'ZPRC_BKP_SCHED_SYS',
293                 where => "CCMS_SID = '$ccms_sid'
294                        AND QUERY_ID = '$query_id'
295                        AND VALID_TO = '$last_run_date'",
296             } );
297
298             if ($s_rows > 0) {
299                 # this means the system is still online so update valid_to
300                 my ($u_rows, $u_message, $u_content, $u_sql) =
301                 GMP::DB->exec_sql( {
302                     action => 'UPDATE',
303                     tables => 'ZPRC_BKP_SCHED_SYS',
304                     fields => [ [ [ 'VALID_TO', $today ] ] ],
305                     where => "CCMS_SID = '$ccms_sid'
306                            AND QUERY_ID = '$query_id'
307                            AND VALID_TO = '$last_run_date'"
308                 } );
309             } else {
310                 # the system is in the new list but not in the old list ->
311                 # it went online in between

```

```

312     # since we don't need the time but only the date, rebuild
313     last change date
314     my $change_date = GMP::DateTime->mod_input_dt(
315         ($system->get_last_change_dt()->{'date'}." 00:00:00");
316     # we need the day before because the system went offline
317     that day so it doesn't count anymore
318     $change_date = GMP::DateTime->get_datetime(
319         $change_date->{epoch} - 24 * 3600, 1)->{'tst'};
320     # create new entry with valid_from = last change date
321     my ($i_rows, $i_message, $i_content, $i_sql) =
322     GMP::DB->exec_sql( {
323         action => 'INSERT',
324         tables => 'ZPRC_BKP_SCHED_SYS',
325         fields => [[
326             [ "CCMS_SID",    $ccms_sid ],
327             [ "QUERY_ID",    $query_id ],
328             [ "VALID_FROM",  $change_date ],
329             [ "VALID_TO",    $today ],
330         ]],
331     } );
332     }
333
334 } # loop at systems currently active in the query
335
336 # now get all systems that are still valid_to = last run date
337 (because these are not in the current list anymore so they
338 were set offline in between)
339 my ($s_rows, $s_content, $s_message, $s_sql) =
340 GMP::DB->exec_sql( {
341     action => 'SELECT',
342     fields => 'CCMS_SID',
343     tables => 'ZPRC_BKP_SCHED_SYS',
344     where => "QUERY_ID = '$query_id'
345             AND VALID_TO = '$last_run_date'",
346 } );
347
348 # and see when they went offline and update their valid_to
349 foreach my $entry (@$s_content) {
350     my $ccms_sid = $entry->{CCMS_SID};
351     my $lstsql = "[ FROM 'SAP System'
352                 WHERE 'CCMS SID' = '$ccms_sid' ]";
353     my @sys_list = GMP::Inventory->list( "SAP System",
354                                       { lstsql => $lstsql } );
355     my $count = @sys_list; #this should be either 0 or 1 because
356                          CCMS_SID is unique
357
358     if ($count == 0) {
359         # system was deleted so we can't update
360         next;
361     } else {
362         # since we don't need the time but only the date, rebuild
363         change date
364         my $change_date = GMP::DateTime->mod_input_dt(
365             ($sys_list[0]->get_last_change_dt()->
366              {'date'}." 00:00:00");
367         # we need the day before because the system went offline
368         that day so it doesn't count anymore
369         $change_date = GMP::DateTime->get_datetime( $change_date->
370             {epoch} - 24 * 3600, 1)->{'tst'};
371         my ($u_rows, $u_message, $u_content, $u_sql) =
372         GMP::DB->exec_sql( {
373             action => 'UPDATE',
374             tables => 'ZPRC_BKP_SCHED_SYS',
375             fields => [ [ [ 'VALID_TO', $change_date ] ] ],

```

```

376         where => "CCMS_SID = '$ccms_sid'
377         AND QUERY_ID = '$query_id'
378         AND VALID_TO = '$last_run_date'"
379     } );
380 }
381 } # loop at systems that went offline
382
383 } # loop at queries
384
385 ### CHECK BACKUPS FOR EACH MISSED DAY ###
386
387 # new loop at queries just to avoid errors
388 foreach my $query (@$q_content) {
389     my $query_id = $query->{QUERY_ID};
390
391     # we have to start ON $last_run_date because the script run on
392     last run date collected backups for the day before
393     # if $last_run_date = 3.1. then backups were checked for 2.1.
394     my $check_date = GMP::DateTime->get_datetime($last_run_date,1);
395
396     while ($check_date->{'tst'} <= $yesterday) { #has to be
397     yesterday because backups for today might not have finished yet
398     my $clear_date = $check_date->{'tst'};
399     # get surrounding dates in format YYYYMMDDhhmmss
400     my $next_date = GMP::DateTime->get_datetime(
401         $check_date->{epoch} + 24 * 3600, 1)->{'tst'};
402     my $day_before = GMP::DateTime->get_datetime(
403         $check_date->{epoch} - 24 * 3600, 1)->{'tst'};
404
405     # get systems that were live each day and count their number
406     my ($l_rows, $l_content, $l_message, $l_sql) =
407     GMP::DB->exec_sql( {
408         action => 'SELECT',
409         fields => "CCMS_SID",
410         tables => 'ZPRC_BKP_SCHED_SYS',
411         where => "QUERY_ID = '$query_id'
412             AND VALID_FROM <= '$clear_date'
413             AND VALID_TO >= '$clear_date'",
414     } );
415
416     next if $l_rows == 0; #if there were no active systems that
417     day we can just skip it
418
419     my $scheduled_systems = $l_rows;
420     my $failed_systems = 0;
421
422     # check their backups states
423     SYSTEM: foreach my $entry (@$l_content) {
424         my $ccms_sid = $entry->{CCMS_SID};
425         # get the DATA backups for that system from Inventory
426         my $lstql = "[ FROM 'Backup' WHERE 'Type' = 'DATA'
427             AND [ FROM 'Backup' WHERE 'SAP Application'
428             <=i2i=> [ FROM 'SAP System'
429             WHERE 'CCMS SID' = '$ccms_sid' ] ] ] ";
430         my @backups = GMP::Inventory->list( "Backup",
431             { lstql => $lstql } );
432         # check if this system has a backup at all
433         my $number_of_bkps = @backups;
434         if ($number_of_bkps == 0) {
435             $failed_systems++;
436             # check if system has failed the day before
437             my ($f_rows, $f_content, $f_message, $f_sql) =
438             GMP::DB->exec_sql( {
439                 action => 'SELECT',

```

```

440         fields => 'COUNT',
441         tables => 'ZPRC_BKP_FAILED',
442         where => "CCMS_SID = '$ccms_sid'
443                 AND QUERY_ID = '$query_id'
444                 AND M_DATE = '$day_before'"
445     } );
446
447     # add the system to the failed table (ZPRC_BKP_FAILED)
448     my $count = 1;
449     $count = $f_content->[0]->{COUNT} + 1 if $f_rows > 0;
450     my ($sif_rows, $sif_message, $sif_content, $sif_sql) =
451     GMP::DB->exec_sql( {
452         action => 'INSERT',
453         tables => 'ZPRC_BKP_FAILED',
454         fields => [[
455             [ "M_DATE",    $clear_date ],
456             [ "CCMS_SID",  $ccms_sid ],
457             [ "QUERY_ID",  $query_id ],
458             [ "COUNT",    $count ],
459         ]],
460     } );
461
462     next SYSTEM; #nothing else to check for this system if
463                 there was no backup at all
464 }
465
466 # says if a failed backup was found
467 my $has_system_failed = 0;
468
469 foreach my $backup (@backups) {
470     # get the very last backup run for the given day (we
471     # assume that the last one was the successful one)
472     my $bkb_id = $backup->id;
473     my ($rows, $r_content, $r_message, $sql) =
474     GMP::DB->exec_sql( {
475         action => "SELECT",
476         fields => "BKP_RET_CODE, RETURN_STATUS",
477         tables => "ZPRC_BKP_STATUS",
478         where => "APP_ID = '$bkb_id'
479                 AND BKP_START_TIME < '$next_date'
480                 AND BKP_START_TIME >= '$clear_date'",
481         options => "ORDER BY BKP_END_TIME DESC LIMIT 1",
482     } );
483
484     # if we found a good backup we can go directly to the
485     # next system (and thus ignore $has_system_failed)
486     if ($rows and $r_content->[0]->{BKP_RET_CODE} == 0
487         and $r_content->[0]->{RETURN_STATUS} == 0) {
488         $has_system_failed = 0;
489         next SYSTEM;
490     } else {
491         $has_system_failed = 1;
492     }
493 } #loop at system's backups
494
495 # $has system failed will only be set if there was no
496 # successful backup (and otherwise it would have started a
497 # new SYSTEM loop)
498 if ($has_system_failed) {
499     $failed_systems++;
500
501     # check if system has failed the day before
502     my ($f_rows, $f_content, $f_message, $f_sql) =
503     GMP::DB->exec_sql( {

```

```

504         action => 'SELECT',
505         fields => 'COUNT',
506         tables => 'ZPRC_BKP_FAILED',
507         where => "CCMS_SID = '$ccms_sid'
508                 AND QUERY_ID = '$query_id'
509                 AND M_DATE = '$day_before'"
510     } );
511
512     # add the system to the failed table (ZPRC_BKP_FAILED)
513     my $count = 1;
514     $count = $f_content->[0]->{COUNT} + 1 if $f_rows > 0;
515     my ($if_rows, $if_message, $if_content, $if_sql) =
516     GMP::DB->exec_sql( {
517         action => 'INSERT',
518         tables => 'ZPRC_BKP_FAILED',
519         fields => [[
520             [ "M_DATE",    $clear_date ],
521             [ "CCMS_SID",  $ccms_sid ],
522             [ "QUERY_ID",  $query_id ],
523             [ "COUNT",    $count ],
524         ]],
525     } );
526 }
527
528 } #loop at systems
529
530 # add new row to ZPRC_BKP_RESULT with number of scheduled
531 # systems and number of failed systems
532 my ($i_rows, $i_message, $i_content, $i_sql) =
533 GMP::DB->exec_sql( {
534     action => 'INSERT',
535     tables => 'ZPRC_BKP_RESULT',
536     fields => [[
537         [ "M_DATE",    $clear_date ],
538         [ "QUERY_ID",  $query_id ],
539         [ "SCHEDULED", $scheduled_systems ],
540         [ "FAILED",    $failed_systems ],
541     ]],
542 } );
543
544 # increment day
545 $check_date = GMP::DateTime->get_datetime(
546     $check_date->{epoch} + 24 * 3600, 1);
547
548 } # while $clear_date <= $yesterday
549
550 } # loop at queries
551
552 } # error handling sub

```

MONITORING.PM

This is a Perl module which contains all functionality regarding the user interface.

```

1 package GMP::Appl::Backups::UI::Reports::Monitoring;
2 use base 'GMP::Appl::Base';
3
4 use strict;
5 use warnings;
6 use 5.010;

```

```

7
8 use GMP::Appl::Base::InputValidation qw(validate);
9 use GMP::DateTime;
10 use GMP::DB;
11 use GMP::GUI::Messages;
12 use GMP::Inventory;
13 use Date::Calc qw( Add_Delta_YM Monday_of_Week Add_Delta_Days );
14 use Data::Dumper;
15
16 __PACKAGE__->authenticate_as( 'bkp_reports.pl' );
17
18 # this method calls the sub that was requested from the user
19 sub routes {
20     [ qr#(?<sub>[^/]+)#x => __PACKAGE__ ],
21 }
22
23 sub show_input_fields {
24
25     my ($self) = @_;
26     my $gui = $self->widgets;
27     my $param = $self->param;
28
29     # set default dates to previous week
30     my $now = GMP::DateTime->get_datetime( time );
31     my ($year,$month,$day) = Monday_of_Week($now->{week},$now->{year});
32
33     $month = "0".$month if $month < 10;
34     $day = "0".$day if $day < 10;
35
36     # get previous Monday and previous day (Sunday) as to and from
37     dates
38     ($year,$month,$day) = Add_Delta_Days( $year, $month, $day, -1 );
39                                             #Sunday
40     $month = "0".$month if $month < 10;
41     $day = "0".$day if $day < 10;
42     my $to = GMP::DateTime->get_datetime(
43         $year.$month.$day.'000000', 1 )->{'date'};
44     ($year,$month,$day) = Add_Delta_Days( $year, $month, $day, -6 );
45                                             #Monday
46     $month = "0".$month if $month < 10;
47     $day = "0".$day if $day < 10;
48     my $from = GMP::DateTime->get_datetime(
49         $year.$month.$day.'000000', 1 )->{'date'};
50
51     $self->output->html_header_args( { title => 'Backup Reporting' } );
52
53
54     # build date fields
55     my $form = $gui->table(
56         $gui->tr(
57             $gui->td( { class => 'fname', align => 'right' },
58                 'Calculate report from' ),
59             $gui->td( { class => 'ffeld' },
60                 $gui->datepicker( {
61                     name => 'from',
62                     id => 'from',
63                     jquery => {dateFormat => 'yy-mm-dd', maxDate => -1},
64                 }, $from,
65             ),
66         ),
67         $gui->td( { class => 'fname', align => 'right' }, 'to' ),
68         $gui->td( { class => 'ffeld' },
69             $gui->datepicker( {
70                 name => 'to',

```

```

71         id => 'to',
72         jquery => { dateFormat => 'yy-mm-dd', maxDate => -1},
73     }, $to,
74 ),
75 ),
76 $gui->td(
77     $gui->button( { lnk => "javascript:show_result();", },
78         'Generate Report' ),
79 ),
80 ),
81 );
82
83 # add feedback link
84 $form .= $gui->table( { width => "100%" },
85     $gui->tr(
86         $gui->td( { align => 'right' },
87             $gui->link(
88                 "Send feedback".$gui->image( "/images/forward.gif",
89                     { border => '0', height => '16', width => '16',
90                         title => 'Send feedback' } ),
91                 { href => 'mailto:anne.wendt@sap.com?
92                     subject=Feedback%20on%20Backup%20Reporting' }
93             ),
94         ),
95     ),
96 );
97
98 # build expandables
99 $form .= $gui->table(
100     $gui->tr(
101         $gui->td(
102             $gui->expandable( {
103                 expandable_id => "prop_audit",
104                 header => "Audit-relevant systems",
105             }, _properties(1), # 1 = audit relevant
106         ),
107     ),
108 ),
109     $gui->tr(
110         $gui->td(
111             $gui->expandable( {
112                 expandable_id => "prop_non_audit",
113                 header => "Non-Audit-relevant systems",
114             }, _properties(0), # 0 = not audit relevant
115         ),
116     )
117 )
118 ).$gui->hr();
119
120 return $gui->p($form) . $gui->div( { id => 'results_div' } );
121
122 }
123
124 sub show_result {
125
126     my ($self) = @_;
127     my $gui = $self->widgets;
128     my $param = $self->param;
129     my $message = GMP::GUI::Messages->new();
130
131     # get the parameters
132     my $from = $param->{from};
133     my $to = $param->{to};
134

```

```

135 # remove all '-' of YY-MM-DD
136 $from =~ s/-//g;
137 $to    =~ s/-//g;
138
139 # convert it to yyyymmddhhmmss
140 $from .= "000000";
141 $to   .= "000000";
142
143 # check date selection
144 return $message->print_note( {icon => 'red'},
145                             "'FROM' may not be after 'TO!'" ) if $from > $to;
146
147 # check if/which queries were selected
148 my @queries = split ' ', $param->{queries};
149 return $message->print_note( {icon => 'red'},
150                             "Please select at least one query!" ) unless @queries;
151
152 # some default values
153 my $percentage = 100;
154 my $three_day_count = 0;
155 my $scheduled = 0;
156 my $failed = 0;
157 my $result = "";
158 my $text = "Results for ";
159 my @query_names;
160
161 # to avoid double counting all SIDs will be stored in an array
162 my @three_day_fails;
163
164 # get data to calculate success values
165 foreach my $query_id (@queries) {
166     # get data for percentage value
167     my ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
168         action => 'SELECT',
169         fields => 'SCHEDULED, FAILED',
170         tables => 'ZPRC_BKP_RESULT',
171         where => "QUERY_ID = '$query_id'
172                 AND ( M_DATE BETWEEN '$from' AND '$to' )",
173     },
174 );
175
176     # sum up for percentage value
177     foreach (@$r_content) {
178         $scheduled += $_->{SCHEDULED};
179         $failed += $_->{FAILED};
180     }
181 }
182
183 # check if there were active systems
184 return $message->print_note({icon => 'red'},
185                             "There were no active systems with this query between
186                             $param->{from} and $param->{to}!" ) if $scheduled == 0;
187
188 # collect all other data if there were active systems
189 foreach my $query_id (@queries) {
190
191     # get data for three day rule
192     my ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
193         action => 'SELECT',
194         fields => 'CCMS_SID, COUNT, M_DATE',
195         tables => 'ZPRC_BKP_FAILED',
196         where => "QUERY_ID = '$query_id'
197                 AND ( M_DATE BETWEEN '$from' AND '$to' )
198                 AND COUNT > 2",

```

```

199         options => "ORDER BY CCMS_SID, M_DATE", # important for
200                                                     correct calculation
201     },
202 );
203
204     # count how many systems violated the rule
205     foreach my $entry (@$r_content) {
206         # this also covers cases when the system fails three times
207         # again (and was running fine in between)
208         if ($entry->{COUNT} == 3) {
209             $three_day_count++;
210             push @three_day_fails, $entry->{SID};
211         } else { # case: system failed for the third time before time
212                 # frame and now starts with counting at > 4
213                 # make sure the system is not counted consecutively
214                 unless ( grep{$_ eq $entry->{SID}} @three_day_fails){
215                     $three_day_count++;
216                     push @three_day_fails, $entry->{SID};
217                 }
218             }
219     }
220
221     # add name of the query to list
222     ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
223         action => 'SELECT',
224         fields => 'Q_NAME',
225         tables => 'ZPRC_BKP_QUERIES',
226         where => "QUERY_ID = '$query_id'",
227     },
228 );
229     push @query_names, $r_content->[0]->{Q_NAME};
230
231 }
232
233 # calculate percentage value and round it
234 $percentage = (($scheduled - $failed) / $scheduled) * 100;
235 $percentage = sprintf("%.2f", $percentage);
236
237 # build the text
238 $text .= $query_names[0]; # first
239 for ( my $i = 1; $i < $#query_names; $i++) { # middle
240     $text .= ", ".$query_names[$i];
241 }
242 $text .= ", and ".$query_names[-1] if $#query_names > 0; #last
243
244 # set colour of boxes depending on values
245 my $perc_colour = "8dc100"; #green
246 $perc_colour = "dd0000" if $percentage < 98; #red
247 my $fail_colour = "8dc100"; #green
248 $fail_colour = "dd0000" if $three_day_count > 0; #red
249
250 my $perc_box = "border: 1px solid #d3d3d3; background: #e6e6e6
251               url(/images/jquery/ui/ui-bg_highlight-soft_75_".
252                   $perc_colour."_1x100.png) 50% 50% repeat-x;";
253 my $fail_box = "border: 1px solid #d3d3d3; background: #e6e6e6
254               url(/images/jquery/ui/ui-bg_highlight-soft_75_".
255                   $fail_colour."_1x100.png) 50% 50% repeat-x;";
256
257 # build the result with text and boxes
258 $result .= $gui->table( { width => "50%" },
259     $gui->tr(
260         $gui->td(
261             $gui->b($text." from ".$param->{from}.
262                 " to ".$param->{to}."."),

```

```

263     ),
264   ),
265   $gui->tr(
266     $gui->td( { class => "ui-corner-all", align => "center",
267               height => "30px", style => $perc_box},
268             "$percentage% of systems were successfully backed up",
269           ),
270   ),
271   $gui->tr(
272     $gui->td( {class => "ui-corner-all", align => "center",
273               height => "30px", style => $fail_box},
274             "$three_day_count systems failed to successfully back up
275             the third day in a row",
276           ),
277   ),
278 ).$gui->br();
279
280 # set second button disabled if there were no failures in selected
281 scope
282 my $btn2_status = "disabled" if $failed == 0;
283 # set third button disabled if there is no data to display
284 my $btn3_status = "disabled" if $three_day_count == 0;
285
286 # add the buttons
287 $result .= $gui->button( { lnk => "javascript:show_overview();" },
288                        "Display calculation overview" ).
289 $gui->button( { lnk => "javascript:show_scheduled();" },
290             "Display systems with scheduled backups
291             (basic population)" ).
292 $gui->button( { lnk => "javascript:show_failed();",
293               status => $btn2_status }, #set button enabled
294             depending on fails
295             "Display backup failures" ).
296 $gui->button( {lnk => "javascript:show_three_failed();",
297               status => $btn3_status }, #set button enabled
298             depending on fails
299             "Display systems that failed three or more
300             days in a row" );
301
302 return $gui->p($result) . $gui->div( { id => 'details_div' } );
303
304 }
305
306 sub show_overview {
307
308   my ($self) = @_;
309   my $gui = $self->widgets;
310   my $param = $self->param;
311   my $message = GMP::GUI::Messages->new();
312
313   # get the parameters
314   my $from = $param->{from};
315   my $to = $param->{to};
316
317   # remove all '-' of YY-MM-DD
318   $from =~ s/-//g;
319   $to =~ s/-//g;
320
321   # convert it to yyyyymmddhhmmss
322   $from .= "000000";
323   $to .= "000000";
324
325   # check date selection
326   return $message->print_note( {icon => 'red'},

```

```

327         "'FROM' may not be after 'TO!')" if $from > $to;
328
329 # check if/which queries were selected
330 my @queries = split ' ', $param->{queries};
331 return $message->print_note( {icon => 'red'}, "Please select at
332     least one query!") unless @queries;
333
334 my $result;
335
336 #build the result
337 foreach my $query_id (@queries) {
338     # get the query name
339     my ($q_rows, $q_content, $q_message, $q_sql) =
340     GMP::DB->exec_sql( {
341         action => 'SELECT',
342         fields => "Q_NAME",
343         tables => 'ZPRC_BKP_QUERIES',
344         where => "QUERY_ID = '$query_id'"
345     },
346     );
347
348     #get data to display
349     my ($r_rows, $r_content, $r_message, $r_sql) =
350     GMP::DB->exec_sql( {
351         action => 'SELECT',
352         fields => "M_DATE, SCHEDULED, FAILED",
353         tables => 'ZPRC_BKP_RESULT',
354         where => "QUERY_ID = '$query_id'
355             AND (M_DATE BETWEEN '$from' AND '$to')"
356     },
357     );
358
359     #combine it
360     foreach my $entry (@$r_content) {
361         push @$result, [
362             GMP::DateTime->get_datetime($entry->{M_DATE},1)->{date},
363             $q_content->[0]->{Q_NAME},
364             $entry->{SCHEDULED},
365             $entry->{FAILED},
366         ];
367     }
368 }
369
370 my @thead = ('Date', 'Query', 'Scheduled', 'Failed');
371
372 return $gui->grid( {
373     title => "Overview about calculation data from
374         $param->{from} to $param->{to} ",
375     table_id => 'overview',
376     head => \@thead,
377     data => $result,
378     export => { filename => "backup_report_".
379         GMP::DateTime::NOW(), columns => \@thead },
380     use_jquery => {
381         bPaginate => 'false',
382         bInfo => 'false',
383     },
384 },
385 );
386
387 }
388
389 sub show_scheduled {
390

```

```

391 my ($self) = @_;
392 my $gui = $self->widgets;
393 my $param = $self->param;
394 my $message = GMP::GUI::Messages->new();
395
396 # get the parameters
397 my $from = $param->{from};
398 my $to = $param->{to};
399
400 # remove all '-' of YY-MM-DD
401 $from =~ s/-//g;
402 $to =~ s/-//g;
403
404 # convert it to yyyyymmddhhmmss
405 $from .= "000000";
406 $to .= "000000";
407
408 # check date selection
409 return $message->print_note( {icon => 'red'},
410     "'FROM' may not be after 'TO!'" ) if $from > $to;
411
412 # check if/which queries were selected
413 my @queries = split ',', $param->{queries};
414 return $message->print_note( {icon => 'red'}, "Please select at
415     least one query!" ) unless @queries;
416
417 my $result;
418
419 #build the result
420 foreach my $query_id (@queries) {
421     # get the query name
422     my ($q_rows, $q_content, $q_message, $q_sql) =
423     GMP::DB->exec_sql( {
424         action => 'SELECT',
425         fields => "Q_NAME",
426         tables => 'ZPRC_BKP_QUERIES',
427         where => "QUERY_ID = '$query_id'"
428     },
429     );
430
431     #get data to display
432     my ($r_rows, $r_content, $r_message, $r_sql) =
433     GMP::DB->exec_sql( {
434         action => 'SELECT',
435         fields => "CCMS_SID, VALID_FROM, VALID_TO",
436         tables => 'ZPRC_BKP_SCHED_SYS',
437         where => "QUERY_ID = '$query_id'
438             AND VALID_FROM <= '$to'
439             AND VALID_TO >= '$from'",
440         options => "ORDER BY VALID_FROM",
441     },
442     );
443
444     #combine it
445     foreach my $entry (@$r_content) {
446         push @$result, [
447             _prepare_CCMS_SID($gui, $entry->{CCMS_SID}, $query_id),
448             $q_content->[0]->{Q_NAME},
449             GMP::DateTime->get_datetime(
450                 $entry->{VALID_FROM}, 1->{date},
451             GMP::DateTime->get_datetime($entry->{VALID_TO}, 1->{date},
452         ],
453     }
454 }

```

```

455
456 my @thead = ('CCMS_SID', 'Query', 'Valid from', 'Valid to');
457
458 return $gui->grid( {
459     title          => "Scheduled systems from
460                     $param->{from} to $param->{to} ",
461     table_id       => 'sched_sys',
462     head           => \@thead,
463     data           => $result,
464     export         => { filename => "bkp_rep_scheduled_systems".
465                     GMP::DateTime::NOW(), columns => \@thead },
466     use_jquery     => {
467         bPaginate  => 'false',
468         bInfo      => 'false',
469     },
470 },
471 );
472
473 }
474
475 sub show_failed {
476
477     my ($self) = @_;
478     my $gui = $self->widgets;
479     my $param = $self->param;
480     my $min_count = $param->{count}-1; # to distinguish between
481                                       show_failed and show_three_failed
482     my $message = GMP::GUI::Messages->new();
483
484     # get the parameters
485     my $from = $param->{from};
486     my $to = $param->{to};
487
488     # remove all '-' of YY-MM-DD
489     $from =~ s/-//g;
490     $to =~ s/-//g;
491
492     # convert it to yyyyymmddhhmmss
493     $from .= "000000";
494     $to .= "000000";
495
496     # check date selection
497     return $message->print_note( {icon => 'red'},
498         "'FROM' may not be after 'TO!'" ) if $from > $to;
499
500     # check if/which queries were selected
501     my @queries = split ' ', $param->{queries};
502     return $message->print_note( {icon => 'red'},
503         "Please select at least one query!" ) unless @queries;
504
505     my $result;
506
507     #build the result
508     foreach my $query_id (@queries) {
509         # get the query name
510         my ($q_rows, $q_content, $q_message, $q_sql) =
511             GMP::DB->exec_sql( {
512                 action => 'SELECT',
513                 fields => "Q_NAME",
514                 tables => 'ZPRC_BKP_QUERIES',
515                 where => "QUERY_ID = '$query_id'"
516             },
517         );
518

```

```

519     #get data to display
520     my ($rows, $r_content, $r_message, $sql) =
521     GMP::DB->exec_sql( {
522         action => 'SELECT',
523         fields => "M_DATE, CCMS_SID, COUNT",
524         tables => 'ZPRC_BKP_FAILED',
525         where => "QUERY_ID = '$query_id'
526                 AND (M_DATE BETWEEN '$from' AND '$to')
527                 AND COUNT > $min_count",
528         options => "ORDER BY M_DATE, CCMS_SID"
529     },
530     );
531
532     #combine it
533     foreach my $entry (@$r_content) {
534         push @$result, [
535             GMP::DateTime->get_datetime($entry->{M_DATE},1)->{date},
536             _prepare_CCMS_SID($gui, $entry->{CCMS_SID}, $query_id),
537             $q_content->[0]->{Q_NAME},
538             $entry->{COUNT},
539         ];
540     }
541 }
542
543 my @thead = ('Date', 'CCMS_SID', 'Query', 'Count');
544
545 return $gui->grid( {
546     title => "Systems with failed backups from
547             $param->{from} to $param->{to} ",
548     table_id => 'sched_sys',
549     head => \@thead,
550     data => $result,
551     export => { filename => "bkp_rep_failed_systems".
552                 GMP::DateTime::NOW(), columns => \@thead },
553     use_jquery => {
554         bPaginate => 'false',
555         bInfo => 'false',
556     },
557 },
558 );
559
560 }
561
562 # this method checks the current system's backup's status and displays
563 it as a little coloured dot
564 sub _prepare_CCMS_SID {
565     my $gui = shift;
566     my $ccms_sid = shift;
567     my $query_id = shift;
568     my $status_btn;
569
570     # get the object id to link it
571     my @sys_list = GMP::Inventory->list( "SAP System", {lstsql => "
572         [ FROM 'SAP System' WHERE 'CCMS SID' = '$ccms_sid' ]" } );
573     my $system_id = $sys_list[0]->{'ITEM'}->{'ID'};
574
575     ### 1 ### check if the system still exists
576     return $gui->link( $ccms_sid, { href => "javascript:alert(
577         'The system has been deleted from the inventory.'" },).
578         "\t".$gui->image( "/images/16x16/dot_gray.png",
579             { border => '0', height => '16', width => '16',
580               title => 'System was decommissioned' } )
581     unless $system_id;
582

```

```

583 # if it does link to the inventory entry
584 my $link = $gui->link( $ccms_sid,
585     { href => "objects.pl?sub=maintain&obj_item_id=$system_id",
586       target => "_blank" }
587 );
588
589 ### 2 ### check if the system is still in scope of the query
590 my ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
591     action => 'SELECT',
592     fields => "VALID_TO",
593     tables => 'ZPRC_BKP_SCHED_SYS',
594     where => "QUERY_ID = '$query_id' AND CCMS_SID = '$ccms_sid'",
595     options => "ORDER BY VALID_TO DESC LIMIT 1"
596 },
597 );
598 return $link."\t".
599     $gui->image( "/images/16x16/dot_turq.png",
600     { border => '0', height => '16', width => '16',
601       title => 'System is out of query scope' } )
602 if $r_content->[0]->{VALID_TO} < GMP::DateTime->mod_input_dt(
603     GMP::DateTime->get_datetime()->{'date'}." 00:00:00")->{'tst'};
604
605 # get backup information
606 my $lstsql = "[ FROM 'Backup' WHERE 'SAP Application' <=i2i=>
607     [ FROM 'SAP System' WHERE 'CCMS SID' = '$ccms_sid' ] ] ";
608 my @backups = GMP::Inventory->list( "Backup", { lstsql => $lstsql });
609
610 ### 3 ### check if backup exists
611 return $link."\t".
612     $gui->image( "/images/16x16/dot_red.png",
613     { border => '0', height => '16', width => '16',
614       title => 'No backup defined for this system!' } )
615 unless @backups;
616
617 my $no_of_run_bkp = 0; #number of backups that had at least one run
618 my @data_backups; #out of all backups for this system, this list
619     will store all DATA backups
620
621 foreach (@backups) {
622     # get the very last run status of that backup
623     my $bkp_id = $_->id;
624     my ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
625         action => "SELECT",
626         fields => "BKP_RET_CODE, RETURN_STATUS, BKP_TYPE,
627             BKP_STATUS",
628         tables => "ZPRC_BKP_STATUS",
629         where => "APP_ID = '$bkp_id'",
630         options => "ORDER BY BKP_END_TIME DESC LIMIT 1",
631     } );
632
633     $no_of_run_bkp += $rows;
634
635     push @data_backups, {
636         bkp_ret_code => $r_content->[0]->{BKP_RET_CODE},
637         return_status => $r_content->[0]->{RETURN_STATUS},
638         bkp_status => $r_content->[0]->{BKP_STATUS}
639     } if $r_content->[0]->{BKP_TYPE} eq 'DATA';
640
641     }
642
643 ### 4 ### check if system has a DATA backup
644 return $link."\t".
645     $gui->image( "/images/16x16/dot_red.png",
646     { border => '0', height => '16', width => '16',

```

```

647         title => 'No DATA backup defined for this system!' } )
648     unless (@data_backups);
649
650     ### 5 ### check if backup ran at all
651     return $link."\t".
652         $gui->image( "/images/16x16/dot_red.png",
653             { border => '0', height => '16', width => '16',
654               title => 'No backup (DATA or LOG) ran for this system!' } )
655     if $no_of_run_bkp == 0;
656
657     # for more than one DATA backups find the best one
658     unless (scalar(@data_backups) == 1) {
659         ### 6 ### check for number of DATA backups
660         $status_btn .= $gui->image( "/images/16x16/dot_yellow.png",
661             { border => '0', height => '16', width => '16',
662               title => "System has scalar(@data_backups) DATA backups!" } )
663     }
664
665     ### 7 ### check if one backup worked - that'd be all we need so we
666     can exit after that
667     foreach my $data_backup (@data_backups) {
668         return $link."\t".$status_btn.
669             $gui->image( "/images/16x16/dot_green.png",
670                 { border => '0', height => '16', width => '16',
671                   title => 'Backup is running fine again.' } )
672         if $data_backup->{bkp_ret_code} == 0 &&
673             $data_backup->{return_code} == 0;
674     }
675
676     # still here? oh well looks like we need to check which error(s)
677     occured
678     # this loop will add up all errors so one system can have several
679     dots next to it
680     foreach my $data_backup (@data_backups) {
681         ### 8 ### backup is running
682         $status_btn .= $gui->image( "/images/16x16/dot_blue.png",
683             { border => '0', height => '16', width => '16',
684               title => 'Backup is currently running.' } )
685         if $data_backup->{return_status} == 1;
686
687         ### 9 ### backup is inactive
688         $status_btn .= $gui->image( "/images/16x16/dot_red.png",
689             { border => '0', height => '16', width => '16',
690               title => 'Error because backup is inactive!' } )
691         if $data_backup->{return_status} == 2;
692
693         ### 10 ### backup finished with job exec error
694         $status_btn .= $gui->image( "/images/16x16/dot_orange.png",
695             { border => '0', height => '16', width => '16',
696               title => 'Finished with job execution error!' } )
697         if $data_backup->{return_status} == 4;
698
699         ### 11 ### backup finished from within GMP (probably running for
700         too long)
701         $status_btn .= $gui->image( "/images/16x16/dot_orange.png",
702             { border => '0', height => '16', width => '16',
703               title => 'Backup was finished from within GMP
704                 (probably running for too long)!' } )
705         if $data_backup->{return_status} == 8;
706
707         ### 12 ### backup is in status 'setup'
708         $status_btn .= $gui->image( "/images/16x16/dot_orange.png",
709             { border => '0', height => '16', width => '16',
710               title => "Backup is in status 'Setup!'" } )

```

```

711         if $data_backup->{bkp_status} eq 'Setup';
712
713     ### 13 ### backup is in status 'decommission' ('decommissi' in
714     status table)
715     $status_btn .= $gui->image( "/images/16x16/dot_red.png",
716         { border => '0', height => '16', width => '16',
717           title => "Backup is in status 'Decommission!'" } )
718     if $data_backup->{bkp_status} eq 'Decommissi';
719
720     ### 14 ### backup is in status 'inactive'
721     $status_btn .= $gui->image( "/images/16x16/dot_red.png",
722         { border => '0', height => '16', width => '16',
723           title => "Backup is in status 'Inactive!'" } )
724     if $data_backup->{bkp_status} eq 'Inactive';
725
726     ### 15 ### backup finished with warning
727     $status_btn .= $gui->image( "/images/16x16/dot_orange.png",
728         { border => '0', height => '16', width => '16',
729           title => 'Backup finished with warning!' } )
730     if $data_backup->{bkp_ret_code} < 32;
731
732     ### 16 ### backup finished with error
733     $status_btn .= $gui->image( "/images/16x16/dot_orange.png",
734         { border => '0', height => '16', width => '16',
735           title => 'Backup finished with error!' } )
736     if $data_backup->{bkp_ret_code} > 31;
737 }
738
739 return($link."\t".$status_btn);
740 }
741
742 # this method creates a list of checkboxes
743 sub _properties {
744
745     my $audit_relevant = shift;
746     my @queries;
747     my %labels;
748
749     # get queries
750     my ($rows, $r_content, $r_message, $sql) = GMP::DB->exec_sql( {
751         action => 'SELECT',
752         fields => 'QUERY_ID, Q_NAME',
753         tables => "ZPRC_BKP_QUERIES",
754         where => "IS_AUDIT_RELEVANT = '$audit_relevant'
755                 AND DISPLAY = '1'",
756         options => 'ORDER BY QUERY_ID'
757     } );
758
759     foreach my $entry (@$r_content) {
760         push @queries, $entry->{QUERY_ID};
761         $labels{$entry->{QUERY_ID}} = $entry->{Q_NAME};
762     }
763
764     # create list of checkboxes
765     my $output = "<div id=checkboxes>";
766     foreach (@queries) {
767         $output .= '<input type="checkbox"
768                 name="queries"
769                 value="' . $_ . '">' . $labels{$_} . ' (ID ' . $_ . ') <br>';
770     }
771     return $output."</div>";
772
773 }
774 1

```

APPENDIX K FOLD-OUT LIST OF REQUIREMENTS

1	Filter by technical landscape	<i>high</i>
2	Show list of affected customers	<i>high</i>
3	Filer storage reporting	<i>high</i>
4	Add more/new systems	<i>high</i>
5	Take only “best backup” into consideration	<i>high</i>
6	Report over not audit-relevant systems	<i>high</i>
7	Update information on upgraded systems	<i>high</i>
8	Take obsolete system categories out of scope	<i>high</i>
9	Display coloured dot next to system to show its current status	<i>medium</i>
10	Let users configure automatic reports	<i>high</i>
11	Send out reports automatically	<i>medium</i>
12	Display detailed failure reason	<i>medium</i>
13	Show more system details	<i>medium</i>
14	Send script error notifications to the developer	<i>medium</i>
15	Find employees who are responsible for each of the applications covered	<i>medium</i>
16	Retrieve the basic population	<i>low</i>
17	Automatically include policies in the reporting	<i>low</i>
18	Display total number of alerts for given scope	<i>low</i>
19	Feedback to be sent to the developer	<i>low</i>
20	Save sets of checkboxes	<i>low</i>
21	Schedule data collection script early	<i>low</i>
22	Create a portable concept	<i>high</i>
23	Use CCMS_SID instead of SID	<i>persistent</i>
24	Offer scope flexibility	<i>high</i>
25	Write readable and maintainable code	<i>persistent</i>
26	Usability	<i>persistent</i>
27	Historicisation of data	<i>persistent</i>
28	Data source manipulation only by automatic scripts	<i>persistent</i>
29	Add landscape as a column to the result table	<i>medium</i>
30	Filter per filer storage	<i>medium</i>
31	Change display text	<i>low</i>
32	Automatically add new ByD versions	<i>medium</i>
33	Follow development guidelines	<i>persistent</i>
34	Include alert reporting	<i>low</i>
35	Good performance	<i>persistent</i>
36	Mark/unmark all checkboxes	<i>low</i>