Never the Twain... Research Methods in (some of) the Computing Disciplines

Stephen MacDonell
Auckland University of Technology, New Zealand
Context – a personal research journey (1)

Information Science:
• algorithms, formal methods

Information Systems:
• processes, organisations, value... users even!

Software Engineering:
• software, design, tools... and users again
Context – a personal research journey (2)

Information Science:
• field test, simulations, experiments, proofs

Meaning that:
• I have encountered a variety of methods
• I (therefore?) see value in them all

Software Engineering:
• prototypes, proofs of concept, metrics, ‘expert’ walkthroughs

Information Systems:
• case studies, qualitative analyses, the odd taxonomy

EE
CE
CS
SE
IT
IS
HARDWARE
SOFTWARE
ORGANIZATIONAL NEEDS

AUT
2007
ISRM
Auckland
In contrast – the ‘one true way’

• For some in computing, valid research can only be undertaken using one particular method
• So knowledge on that method alone is needed
• In software engineering, the ‘one true way’ is akin to atheism, or perhaps agnosticism – i.e. there is no research method in SE, or at least its place is doubtful
• Methodology – what methodology?

ISRM 2007
Auckland
What we say to SE research students...

What is your ontology wrt reality?
What is your epistemology? Is there more to software engineering than science? What methodology will you use?
What they hear...

blah blah blah ontology blah
blah blah blah blah blah
blah blah software engineering
science blah methodology
Incremental advances

• Glass et al. (2004) reported a review of the computing research literature across CS, SE and IS:
  – Research in SE has been dominated by formulative and descriptive work (86%) rather than evaluative, compared with 33% in IS
  – Research in SE has been dominated by conceptual analyses and concept implementations (71%); in IS, methods are more varied: conceptual analyses and implementations 28%, field studies 25%, lab experiments with human subjects 16%, case studies 13%
  – Towards ‘better’ research in software engineering...
Current initiatives

- Increasing attention being paid to design science methods
  - adds legitimacy to the notion of research through the production of artifacts
  - tends to treat the basis of production and evaluation as something of a given, something self-evident

- Significant effort to promote and adopt evidence-based software engineering
  - a more evaluative approach to SE research outcomes
  - dominated by the medical research model that relies primarily on the scientific method
Augmenting design science...

Research Activities

<table>
<thead>
<tr>
<th>Constructs</th>
<th>Build</th>
<th>Evaluate</th>
<th>Theorize</th>
<th>Justify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instantiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nunamaker et al. (1991)
March and Smith (1995)
Peffers et al. (2006)

System Development Research Process

Construct a Conceptual Framework
Develop a System Architecture
Analyze & Design the System
Build the (Prototype) System
Observe & Evaluate the System

Possible entry points for research

AUT UNIVERSITY
Auckland

ISRM 2007
Table 1 Design and design science process elements from IS other disciplines and synthesis objectives for a design science research process in IS.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Problem identification and motivation</td>
<td>Programming Data collection</td>
<td>Problem enumeration</td>
<td>Analysis</td>
<td>Construct a conceptual framework</td>
<td>Meta-requirements Kernel theories</td>
<td>Identify a need</td>
<td>Important and relevant problems</td>
</tr>
<tr>
<td>2. Objectives of a solution</td>
<td>Solution</td>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Implicit in “relevance”</td>
</tr>
<tr>
<td>3. Design and development</td>
<td>Analysis</td>
<td>Synthesis, Tentative design proposals</td>
<td>Develop a system</td>
<td>Design method</td>
<td>Build</td>
<td>Artifcet</td>
<td>Iterative search process</td>
</tr>
<tr>
<td>4. Demonstration</td>
<td>Simulation, Conditional prediction</td>
<td></td>
<td>Experiment, observe, and evaluate the system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Evaluation</td>
<td>Confirmatory evaluation</td>
<td>Evaluation, Decision, Definite design</td>
<td>Testable design process/product hypotheses</td>
<td>Evaluate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Communication</td>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Communication</td>
</tr>
</tbody>
</table>
...using grounded theory

© Petteri Kaskenpalo

ISRM 2007
Auckland
Closing comments

- Conviction with respect to method(ology) is vital, but fanaticism can be unhelpful.
- Absence of a theoretical framework does not mean absence of a theory.
- We can do better in software engineering – grounded design science is one option.
- Does the multi-method approach have a chance?
- Perhaps this is another opportunity for the two research communities to learn from one another.