Simulation of Systems Development Processes – Why, What and How?

LUCAS Breakfast Presentation – 20 April 2011

Dietmar Pfahl
1. Background

2. Current Research
   • SPS* – Why?
   • SPS – What?
   • SPS – How?

3. Future Research Goals
   • VSEL**

*SPS = Software Process Simulation
**VSEL = Virtual Software Engineering Laboratory
My Background – Industry and Academia

- Siemens AG (1987-90, 1993-96)
- German Aerospace Research Est. (1992)
- Fraunhofer IESE (1996-2005)
- University of Calgary (2005-2007, adjunct since 2008)
- Simula Research Laboratory (2008-09)
- University of Oslo (2008-10)
My Background – Applied Research

- Process Management
- Project Management
- Quality Management
- Product Management

System Development

- System Requirements Analysis
- System Design
- SW-/HW-Requirements Analysis
- SW Architecture
- SW Design
- Coding
- SW Integration
- System Integration
- Deployment

Project Management

Quality Management

Product Management

Process Management
My Background – Applied Research

Content

1. Background

2. Current Research
   - SPS* – Why?
   - SPS – What?
   - SPS – How?

3. Future Research Goals
   - VSEL**

*SPS = Software Process Simulation
**VSEL = Virtual Software Engineering Laboratory
SPS Models – Why?

It’s standard in other engineering disciplines:
• Industrial building processes
• Product engineering processes
• Chemical engineering processes
• Pharmaceutical engineering processes
• Hydrological engineering processes
• Manufacturing processes
• Biological/Ecological processes
• Flight simulators
• Aeronautics
• …
• **To understand**
  – …the interesting characteristics of an existing or desired (complex, socio-technical) development system (process, project)
  – SPS models as analysis tools (e.g., for risk analysis)

• **To predict**
  – …the interesting characteristics of the system (process, project) by analysing its model’s performance
  – SPS models as planning tools

• **To evaluate**
  – …the properties of alternative systems (process, project) by comparing its models’ performances
  – SPS models as planning tools

• **To communicate**
  – …the design and functioning of the system (process, project) to others
  – SPS models as documentation and education tools
1. Background

2. Current Research
   - SPS* – Why?
   - SPS – What?
   - SPS – How?

3. Future Research Goals
   - VSEL**

*SPS = Software Process Simulation
**VSEL = Virtual Software Engineering Laboratory
SPS Models – What?

- Operational Release Planning (ORP)
  - Example of using SPS for planning, risk analysis, and decision-making
  - Complements existing optimization models for Strategic Release Planning (⇒ DECIDE\textsubscript{Release})

Reasons for ORP re-planning:
- Drop in/out of developers
- Addition/deletion of features
- Adjustment of effort estimates
- Adjustment of productivity estimates
- Adjustment of task dependencies

All the above …
… in any combination
… at any point in time during release (repeatedly)
• ORP: Planning and Re-Planning in the context of SRP
• Re-Planning Example (enlarged):

Developer D4 becomes unavailable at the beginning of week 10
**SPS Models – What?**

- **Operational Release Planning**
  - Example of using SPS for planning, risk analysis, and decision-making
  - Complements existing optimization models for Strategic Release Planning (DECI\(\text{DE}_{\text{Release}}\))

---

**Sampling of input parameter values from probability distributions**

**Monte-Carlo & Process Simulation**

**Distributions of output parameter values**
SPS Models – What?

- Risk Analysis for Release Planning
  - To analyze impact of task effort and developer productivity over/under-estimation on
    - Release Duration
    - Developer allocation to tasks
    - Task scheduling:
      → Start and end times of tasks
## SPS Models – What?

- **Baseline Case**
- **Triangular Distributions**

### Table 1a. Feature/Task-specific efforts

<table>
<thead>
<tr>
<th>Task</th>
<th>eff(i, j) [person-week]</th>
</tr>
</thead>
<tbody>
<tr>
<td>t(1)</td>
<td>f(1) 3, f(2) 8, f(3) 6, f(4) 3, f(5) 5, f(6) 7, f(7) 10, f(8) 6</td>
</tr>
<tr>
<td>t(2)</td>
<td>f(1) 6, f(2) 3, f(3) 10, f(4) 3, f(5) 6, f(6) 5, f(7) 8</td>
</tr>
<tr>
<td>t(3)</td>
<td>f(1) 6, f(2) 2, f(3) 5, f(4) 6, f(5) 4, f(6) 3, f(7) 10</td>
</tr>
</tbody>
</table>

### Table 1b. Developer-specific productivities

<table>
<thead>
<tr>
<th>Task</th>
<th>p(k, j) [no unit]</th>
</tr>
</thead>
<tbody>
<tr>
<td>t(1)</td>
<td>d(1) 1.5, d(2) 1, d(3) 2, d(4) 0, d(5) 0.5, d(6) 2</td>
</tr>
<tr>
<td>t(2)</td>
<td>d(1) 2, d(2) 1.5, d(3) 1, d(4) 2, d(5) 1.5, d(6) 1</td>
</tr>
<tr>
<td>t(3)</td>
<td>d(1) 1, d(2) 2, d(3) 0, d(4) 1.5, d(5) 2, d(6) 1</td>
</tr>
</tbody>
</table>

### Table 2. ProSim/ORP risk factor variation

<table>
<thead>
<tr>
<th>Case</th>
<th>Variation Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Effort</td>
<td>-10%</td>
<td>20%</td>
</tr>
<tr>
<td>Case 2</td>
<td>Effort</td>
<td>-20%</td>
<td>20%</td>
</tr>
<tr>
<td>Case 3</td>
<td>Effort</td>
<td>-20%</td>
<td>40%</td>
</tr>
<tr>
<td>Case 4</td>
<td>Effort</td>
<td>-40%</td>
<td>40%</td>
</tr>
<tr>
<td>Case 5</td>
<td>Productivity</td>
<td>-20%</td>
<td>10%</td>
</tr>
<tr>
<td>Case 6</td>
<td>Productivity</td>
<td>-20%</td>
<td>20%</td>
</tr>
<tr>
<td>Case 7</td>
<td>Productivity</td>
<td>-40%</td>
<td>20%</td>
</tr>
<tr>
<td>Case 8</td>
<td>Productivity</td>
<td>-40%</td>
<td>40%</td>
</tr>
<tr>
<td>Case 9</td>
<td>Mixed</td>
<td>Case 1 + Case 5</td>
<td></td>
</tr>
<tr>
<td>Case 10</td>
<td>Mixed</td>
<td>Case 2 + Case 6</td>
<td></td>
</tr>
<tr>
<td>Case 11</td>
<td>Mixed</td>
<td>Case 3 + Case 7</td>
<td></td>
</tr>
<tr>
<td>Case 12</td>
<td>Mixed</td>
<td>Case 4 + Case 8</td>
<td></td>
</tr>
</tbody>
</table>
• Effect on Duration

Table 3a. Summary statistics of ORP performance parameter Dur_run

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>22.71875</td>
<td>23.811</td>
<td>1.611</td>
<td></td>
</tr>
<tr>
<td>Case 2</td>
<td>22.71875</td>
<td>23.146</td>
<td>1.608</td>
<td></td>
</tr>
<tr>
<td>Case 3</td>
<td>22.71875</td>
<td>24.755</td>
<td>2.210</td>
<td></td>
</tr>
<tr>
<td>Case 4</td>
<td>22.71875</td>
<td>23.526</td>
<td>2.428</td>
<td></td>
</tr>
<tr>
<td>Case 5</td>
<td>22.71875</td>
<td>23.780</td>
<td>1.012</td>
<td></td>
</tr>
<tr>
<td>Case 6</td>
<td>22.71875</td>
<td>23.075</td>
<td>1.250</td>
<td></td>
</tr>
<tr>
<td>Case 7</td>
<td>22.71875</td>
<td>24.849</td>
<td>2.034</td>
<td></td>
</tr>
<tr>
<td>Case 8</td>
<td>22.71875</td>
<td>23.508</td>
<td>2.141</td>
<td></td>
</tr>
<tr>
<td>Case 9</td>
<td>22.71875</td>
<td>24.541</td>
<td>1.434</td>
<td></td>
</tr>
<tr>
<td>Case 10</td>
<td>22.71875</td>
<td>23.038</td>
<td>2.080</td>
<td></td>
</tr>
<tr>
<td>Case 11</td>
<td>22.71875</td>
<td>26.367</td>
<td>2.495</td>
<td></td>
</tr>
<tr>
<td>Case 12</td>
<td>22.71875</td>
<td>23.950</td>
<td>3.331</td>
<td></td>
</tr>
</tbody>
</table>

• Case 9: Distribution of Durations

Histogram (Spreadsheet1 1v*50c)
Var1 = 50*0.5*normal(x, 24.5413, 1.4342)
N(24.54, 1.43)
### SPS Models – What?

- **Effect on Duration**
- **Single Sample T-Test**

#### Table 5. Results from the single sample t-tests comparing the distribution of the average durations in each case with the duration in the baseline case

<table>
<thead>
<tr>
<th>Case (n=50)</th>
<th>Baseline Duration</th>
<th>Duration Mean</th>
<th>Duration Std. Dev.</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>22.71875</td>
<td>23.811</td>
<td>1.611</td>
<td>0.000</td>
</tr>
<tr>
<td>Case 2</td>
<td>22.71875</td>
<td>23.146</td>
<td>1.608</td>
<td>0.066</td>
</tr>
<tr>
<td>Case 3</td>
<td>22.71875</td>
<td>24.755</td>
<td>2.210</td>
<td>0.000</td>
</tr>
<tr>
<td>Case 4</td>
<td>22.71875</td>
<td>23.526</td>
<td>2.428</td>
<td>0.023</td>
</tr>
<tr>
<td>Case 5</td>
<td>22.71875</td>
<td>23.780</td>
<td>1.012</td>
<td>0.000</td>
</tr>
<tr>
<td>Case 6</td>
<td>22.71875</td>
<td>23.075</td>
<td>1.250</td>
<td>0.049</td>
</tr>
<tr>
<td>Case 7</td>
<td>22.71875</td>
<td>24.849</td>
<td>2.034</td>
<td>0.000</td>
</tr>
<tr>
<td>Case 8</td>
<td>22.71875</td>
<td>23.508</td>
<td>2.141</td>
<td>0.012</td>
</tr>
<tr>
<td>Case 9</td>
<td>22.71875</td>
<td>24.541</td>
<td>1.434</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Case 10</strong></td>
<td><strong>22.71875</strong></td>
<td><strong>23.038</strong></td>
<td><strong>2.080</strong></td>
<td><strong>0.284</strong></td>
</tr>
<tr>
<td>Case 10</td>
<td>22.71875</td>
<td>26.367</td>
<td>2.495</td>
<td>0.000</td>
</tr>
<tr>
<td>Case 12</td>
<td>22.71875</td>
<td>23.950</td>
<td>3.331</td>
<td>0.012</td>
</tr>
</tbody>
</table>

**alpha = 0.05**

---

© Dietmar Pfahl, 2011
**SPS Models – What?**

- Effect on Developer Allocation
- Total number of allocations: \( 8 \times 3 = 24 \)
- With 6 developers:

### Table 3b. Summary statistics of ORP performance parameter Alloc_diff

<table>
<thead>
<tr>
<th>Case</th>
<th>Difference in developer allocation Alloc_diff [no unit]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Case 1</td>
<td>7.2</td>
</tr>
<tr>
<td>Case 2</td>
<td>9.04</td>
</tr>
<tr>
<td>Case 3</td>
<td>9.765</td>
</tr>
<tr>
<td>Case 4</td>
<td>10.14</td>
</tr>
<tr>
<td>Case 5</td>
<td>11.38</td>
</tr>
<tr>
<td>Case 6</td>
<td>12.26</td>
</tr>
<tr>
<td>Case 7</td>
<td>13.3</td>
</tr>
<tr>
<td>Case 8</td>
<td>14.06</td>
</tr>
<tr>
<td>Case 9</td>
<td>11.96</td>
</tr>
<tr>
<td>Case 10</td>
<td>12.16</td>
</tr>
<tr>
<td>Case 11</td>
<td>13.08</td>
</tr>
<tr>
<td>Case 12</td>
<td>13.3</td>
</tr>
</tbody>
</table>

**Box & Whisker Plot**

- Median: 29% change
- Median: 58% change
SPS Models – What?

- Effect on Task Scheduling

Effect on Task Scheduling

Data:

\[ Dv_{\text{diff}} \in [0, 1] \]

Table 3c. Summary statistics of ORP performance parameters ST_diff, ET_diff, Dv_diff

<table>
<thead>
<tr>
<th>Case</th>
<th>ST_diff Mean</th>
<th>ST_diff Std. Dev.</th>
<th>ET_diff Mean</th>
<th>ET_diff Std. Dev.</th>
<th>Dv_diff Mean</th>
<th>Dv_diff Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.26</td>
<td>11.34</td>
<td>25.44</td>
<td>11.9</td>
<td>0.4214</td>
<td>0.1503</td>
</tr>
<tr>
<td>2</td>
<td>27.67</td>
<td>16.67</td>
<td>35.28</td>
<td>17.79</td>
<td>0.5026</td>
<td>0.1533</td>
</tr>
<tr>
<td>3</td>
<td>32.72</td>
<td>17.02</td>
<td>42.93</td>
<td>17.76</td>
<td>0.5345</td>
<td>0.136</td>
</tr>
<tr>
<td>4</td>
<td>40.76</td>
<td>17.8</td>
<td>52.43</td>
<td>19.52</td>
<td>0.6002</td>
<td>0.1299</td>
</tr>
<tr>
<td>5</td>
<td>28.15</td>
<td>17.8</td>
<td>33.71</td>
<td>17.84</td>
<td>0.4958</td>
<td>0.1465</td>
</tr>
<tr>
<td>6</td>
<td>34.28</td>
<td>17.55</td>
<td>40.8</td>
<td>17.48</td>
<td>0.5536</td>
<td>0.1344</td>
</tr>
<tr>
<td>7</td>
<td>46.01</td>
<td>20.96</td>
<td>54.09</td>
<td>21.32</td>
<td>0.6096</td>
<td>0.1306</td>
</tr>
<tr>
<td>8</td>
<td>47.95</td>
<td>20.49</td>
<td>57.06</td>
<td>21.55</td>
<td>0.6368</td>
<td>0.112</td>
</tr>
<tr>
<td>9</td>
<td>38.29</td>
<td>18.62</td>
<td>47.41</td>
<td>19.19</td>
<td>0.5742</td>
<td>0.123</td>
</tr>
<tr>
<td>10</td>
<td>41.05</td>
<td>16.03</td>
<td>49.71</td>
<td>16.26</td>
<td>0.61</td>
<td>0.0987</td>
</tr>
<tr>
<td>11</td>
<td>60.34</td>
<td>22.13</td>
<td>75.17</td>
<td>26.19</td>
<td>0.6902</td>
<td>0.0817</td>
</tr>
<tr>
<td>12</td>
<td>62.53</td>
<td>32.56</td>
<td>75.81</td>
<td>30.18</td>
<td>0.7052</td>
<td>0.0717</td>
</tr>
</tbody>
</table>
1. Background

2. Current Research
   - SPS* – Why?
   - SPS – What?
   - SPS – How?

3. Future Research Goals
   - VSEL**

*SPS = Software Process Simulation
**VSEL = Virtual Software Engineering Laboratory
• A Virtual Software Engineering Laboratory (VSEL)
  – Exploring the idea of SPS building blocks (macro-patterns)
  – Currently development of an open source MATLAB/SIMULINK library under way (→ Th. Birkhoelzer, Konstanz)
  – Example: Composition of SPS representing the V-model
    → E.g., used to investigate the effects of various factors (e.g., quantity and quality of personnel resources, effectiveness of inspection and test activities) on project performance (effort consumption, duration, defect density)
Macropattern 1: Development and Verification

- Development Activity (Maturity)
  - Artifact to develop > 0
  - Artifact to do = 0 AND Total V&V status = Complete
  - Artifact to rework > 0
  - Artifact to do = 0 AND Total V&V status <> Complete

- Verification Activity (Maturity)
  - Artifact > 0 AND Artifact dev status = Complete
  - Artifact = 0 AND Detected artifact faults < Quality threshold for artifact
  - Artifact = 0 AND Detected artifact faults > Quality threshold for artifact
SPS Models – How?

- Example Process

V-Model Process Structure:
- Requirements
- Design
- Code
- Unit Test
- Integration Test
- System Test

V-Model:
Development, Verification (Inspection), Validation (Test)

Macro-Pattern 1

V-Model Process Structure:
- System Level
- Subsystem Level
- Module Level
SPS Models – How?

• Implementation (System Dynamics, MATLAB/SIMULINK)
**SPS Models – How?**

- **GENSIM 2.0 – Generic Simulator**
  - Requirements Analysis and Specification
  - Design
  - Implementation

**Skills**

**Learning**

**Policies (Rules)**

**Tools**

**Communication**

**Techniques**
Possible Analyses (list not complete):

- What combinations (and intensity levels) of development, verification and validation techniques should be applied in a given context to achieve defined time, quality or cost goals?
- How do workforce size and skill variations impact project performance (project duration, effort consumption, code quality)?
- Does investment in training pay off for specific development contexts and goals?
- Do investments in improving development, verification, and validation techniques pay off for specific development contexts and goals?
- What are the promising areas of research for improving development, verification, and validation techniques?
1. Background

2. Current Research
   • SPS* – Why?
   • SPS – What?
   • SPS – How?

3. Future Research Goals
   • VSEL**

*SPS = Software Process Simulation
**VSEL = Virtual Software Engineering Laboratory
Virtual Software Engineering Laboratory (VSEL)

• Why
  – To provide a test-bed for evaluating issues or new ideas in a variety of contexts
  – To support decision-making

• What
  – To analyse/visualise cost-effectiveness of new development approaches, e.g., model-based development (i.e., model-based testing)
  – To analyse effects of local improvements (e.g., better skills for a specific tasks, better effectiveness of a specific inspection or test technique) or local problems (e.g., unclear or incomplete requirements, requirements volatility) on overall project/process/organisational performance

• How
  – Provision of a library of customizable model components macro-patterns)
  – Calibration of customised process simulator based on empirical data and expert knowledge
Tack så mycket!

Questions?
Publication Summary (since 1994)

- **Books/Proceedings + Book Chapters:** 8 + 8
- **Journals:** 17
  - Software Process Improvement & Practice: 5
  - Information and Software Technology: 3
  - The Journal of Systems and Software: 2
  - IJSEKE: 2
  - Empirical Software Engineering: 1
  - IEEE Transactions on Reliability: 1
  - Others: 3
- **Conference and Workshop Proceedings:** 51
  - SEKE (incl. SEDECS): 6
  - ProSim (ICSE-Workshop): 6
  - PROFES: 5
  - ICSP: 4
  - ISESE: 3
  - METRICS: 3
  - ESCOM: 3
  - LSO: 3
  - Others: 18