Software Change in the Solo Iterative Process: An Experience Report

Christopher Dorman         Václav Rajlich
Solo Iterative Process (SIP)

• Applicable to:
  • Small application development, i.e. smartphone applications
  • Maintenance of stable products
  • Development of open source
  • First step towards teaching of iterative and agile processes
Software Change (SC) and SIP

Initiation

Concept Location

Impact Analysis

Prefactoring

Actualization

Postfactoring

Conclusion

Change requests

Product backlog
Iteration backlog

Software changes

Commit

Baseline

Iteration/release

Delivery

Users

Verification

Wayne State
Exploratory Case Study

- Are the SC and SIP models complete?
  - Do they account for programmer’s activities and tasks?
- Is the SC model an effective process?
  - Does SC model support reasonable productivity and quality?
- Are tasks and phases defined sufficiently?
  - Do they give the programmer all necessary guidance?
Case Study Design

- Single programmer
  - Limited university programming experience
- Technologies portfolio
  - Java + Eclipse
  - JUnit
  - Clover Java Code Coverage & Test Optimization
  - Mylyn task management + Tasktop
  - Abbot Java GUI Test Framework
  - Subversion + TortoiseSVN
  - JRipples
Object Program

- 1070 classes – 76 KLOC
- 441 unit tests
- Swing libraries
- Advanced file explorer
- Open source
- [http://www.mucommander.com](http://www.mucommander.com)
Object Program GUI
<table>
<thead>
<tr>
<th>Title</th>
<th>Change Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Search</td>
<td>Add a basic search function that searches in the current directory for all or part of the title of a folder or file.</td>
</tr>
<tr>
<td>Recursive Search</td>
<td>Add search inside all directories.</td>
</tr>
<tr>
<td>Advanced Output</td>
<td>Change the output to a window similar to the main muCommander window.</td>
</tr>
<tr>
<td>Date Search</td>
<td>Add search by a date of file’s modification.</td>
</tr>
<tr>
<td>Case Sensitive Search</td>
<td>Add search by case sensitive search terms.</td>
</tr>
<tr>
<td>Extension Search</td>
<td>Add search for files with specific extensions.</td>
</tr>
<tr>
<td>Properties Search</td>
<td>Add search for files based on their properties.</td>
</tr>
<tr>
<td>Directory Chooser Bug</td>
<td>File chooser doesn't update the search directory.</td>
</tr>
<tr>
<td>Date Bug</td>
<td>DateOption is not removed when disabled.</td>
</tr>
</tbody>
</table>
SC 1 Basic Search

• User story
  Search in the current directory for all or part of the title and return a list of the matching files and subdirectories.

• Concepts
  • search
  • current directory
  • matching files and subdirectories
SC 1 Impact Analysis

Legend
- Concept Location
- Impacted
- Propagating
- Visited
- Not Visited

ToolBar

ToolBarAttributes

ActionBar

MainMenuBar

156 Classes

13 More Classes
SC 1 Other Phases

- No Prefactoring
- Postfactoring
  - Deleted several unused methods
  - Added comments
- Conclusion – New Baseline
New GUI after SC 1
Partial UML of All 9 SC of Iteration
<table>
<thead>
<tr>
<th>Change</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Total/AVG</th>
</tr>
</thead>
</table>

- **Small outliers SC 8 & 9**
  - Bug fixes – no Concept Location or Impact Analysis
- **Large outlier SC 3**
  - Epic – should have been divided into 2 or 3 SC
- **Range of the rest**
  - 13:41 – 22:37 hours:minutes, i.e. ~2 – 3 working days
Productivity SIP iteration
- 4563 LOC added / 144.5 hours = 31.6 LOC per hour

Comparable to Personal Software Process (PSP) [W. Humphrey]
- Range 12 to 42 LOC per hour [L. Prechelt and B. Unger]

Added business value
- Actualization 25.6 hours
- Actualization testing 24.1 hours
- \(\frac{25.4 + 24.1}{144.5} \approx 34\%\)
• 11 bugs introduced
• 9 fixed immediately
• 2 fixed later (changes 8 and 9)
• Added unit test class for each class
• Added functional test class for each feature
• Only done on 1\textsuperscript{st} and 3\textsuperscript{rd} SC

• Spent only 55 min in CL (less than 1%)

• Still indispensable key to SC
Impact Analysis Experience

- IA predicts Prefactoring and Actualization
  - Cannot predict postfactoring
  - Test class impact is difficult to predict
- Exit Criteria
  - Inspect 60% of probably impacted classes
    - Planning phase, no need for 100% accuracy
    - Better exit criterion needed
Impact Analysis Experience

- **Example of overestimate: SC 3**
  - Large prefactoring
  - Estimated that all 6 suppliers will be impacted
  - Only 4 were impacted

- **Example of underestimate: SC 7**
  - Impact to one class missed
  - Class interacted with 308 other classes
Prefactoring Experience

- Introduce base classes, patterns
  - Structure needed for actualization
- Exit Criteria
  - Code is ready for actualization
  - Significant concepts of the SC have their own class

Diagram:
- Initiation
- Concept Location
- Impact Analysis
- Prefactoring
- Actualization
- Postfactoring
- Conclusion

VE IF I ON
Actualization Experience

- Implement and incorporate new functionality
- Exit Criteria
  - Requirements of change request are met
  - All tests pass (unit, functional and regression)
  - Test coverage of 60% of new or modified code
Postfactoring Experience

• Clean-up the consequences of actualization
  Technical debt can build up from previous SCs

• Exit Criteria
  • “If it stinks, change it.” [Beck and Fowler]
  • “When it no longer stinks, stop”
    • Identifiers explain responsibility
    • Single significant concept per class
    • …
Conclusions and Future Work

• Threats to validity
  • Case study - generalize the results with caution
• SC and SIP models successful in this project
• Future work
  • How applicable is the SC model to team projects?
  • Is there a better Impact Analysis exit criterion?
  • Can postfactoring be predicted accurately?
  • Is better integration of technologies needed?
  • New additional tools? (Work Metric, Exit Criteria)
Contents a New Book

- Introduction to iterative processes
  - SIP
- Team iterative processes
  - Agile
  - Directed
  - SC is a key part of iter...
- Beginning and end of life span
- Available at Agile2012 Bookstore, Amazon