Software Project Management Anti-patterns in Students’ Projects

Raptopoulou Charikleia (raptopoc@csd.auth.gr)
Poranen Timo (timo.t.poranen@uta.fi)
Berki Eleni (eleni.berki@uta.fi)
Stamilos Ioannis (stamelos@csd.auth.gr)
Presentation Outline

- Introduction to Anti-patterns
- Data Gathering
- Survey’s Questionnaire
- Students and anti-patterns
- Anti-patterns met by students
- Most frequent anti-patterns
- Anti-patterns Consequences
- Conclusions
What is an Anti-pattern

• An anti-pattern is a commonly occurring solution to a problematic situation that generates overwhelming negative results.

• An anti-pattern may be the result of:
  ▫ A manager or a developer not knowing any better,
  ▫ Or not having sufficient knowledge or experience in solving a particular type of problem,
  ▫ Or having applied a perfectly good pattern in the wrong context.
Anti-pattern Template

• Name
  ▫ A short name that conveys the anti-pattern’s meaning.

• Central Concept
  ▫ A short synopsis of the anti-pattern

• Causes
  ▫ Causes which lead to the anti-pattern

• Symptoms
  ▫ Describing how to recognize the anti-pattern

• Consequences

• Refactored Solution
  ▫ Describing how to change the anti-pattern into a healthier situation
Data Gathering

- Academic year 2010-2011
- SPM students were managing 12 software development projects
- Every project team consisted of 2 project managers (from SPM course) and 4-6 developers (from software engineering course)
- 4 out of 12 projects the daily working language was English
- Average amount of projects’ work was 1200 hours
- Scrum development
Students and Anti-patterns

- Students selected an anti-pattern and wrote the description to the course’s wiki.
- Managers were asked to describe their own remarks on the anti-patterns in the context of students’ projects.
- Students familiarized themselves with the concept of anti-patterns by studying a relatively large amount of them.
Survey’s Questionnaire

- During the course, the managers were asked about the anti-patterns they encountered during the initiation, development and project finishing phase.
- Initiation phase
  - 24 answers from 12 projects
- Development phase
  - 18 managers from 11 projects
- Project Finishing
  - 15 managers from 11 projects
## Anti-patterns in Students’ Projects

<table>
<thead>
<tr>
<th></th>
<th>Initiation phase</th>
<th>Development phase</th>
<th>Project finishing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project 1:</strong></td>
<td>Unbearable Lightness of Being</td>
<td>Cut and Paste programming</td>
<td>Fear of Success</td>
</tr>
<tr>
<td><strong>Project 2:</strong></td>
<td>Hidden Requirements, Reinvent the Wheel</td>
<td>Train the Trainer</td>
<td>Hero Culture, Gilding the Lily</td>
</tr>
<tr>
<td><strong>Project 3:</strong></td>
<td>Hidden Requirements</td>
<td>Corncob, firedrill,, DLL Hell</td>
<td></td>
</tr>
<tr>
<td><strong>Project 4:</strong></td>
<td></td>
<td>We are Idiots</td>
<td></td>
</tr>
<tr>
<td><strong>Project 5:</strong></td>
<td>E-mail Is Dangerous, Project Mismanagement,</td>
<td>Train the Trainer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architects don’t Code, Idiot Proof Process,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Train the Trainer, Hero Culture</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project 6:</strong></td>
<td></td>
<td>Appointed Team</td>
<td>Untested but finished, Yet Another</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Meeting Will Solve it</td>
</tr>
</tbody>
</table>
### Anti-patterns in Students’ Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Initiation phase</th>
<th>Development phase</th>
<th>Project finishing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 7:</td>
<td>Fire Drill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project 8:</td>
<td>Unbearable Lightness of Being, The Feud, Hidden Requirements</td>
<td>Train the Trainer, Mushroom Management</td>
<td></td>
</tr>
<tr>
<td>Project 9:</td>
<td>Emperor’s New Clothes, They understood me</td>
<td>Gilding the Lily, Project Mismanagement</td>
<td>Death by Planning</td>
</tr>
<tr>
<td>Project 10:</td>
<td>Ambiguous Viewpoint Hero Culture</td>
<td></td>
<td>If It Is Working Don’t change It, The Customers Are Idiots, Myopic Delivery, Untested but finished, Mushroom Management</td>
</tr>
<tr>
<td>Project 11:</td>
<td>Hidden Requirements</td>
<td></td>
<td>The Customer</td>
</tr>
<tr>
<td>Project 12:</td>
<td></td>
<td></td>
<td>Untested but Finished</td>
</tr>
</tbody>
</table>
A brand new Anti-pattern!! (by Johanna Huhtala)

- Unbearable Lightness of Being
  - A project group is formed.
  - It is assumed that every member of the project group has (more than) enough to do.
  - In reality, some group members have not much to do.
  - Their tasks are either sort of which cannot be proceeded until some other part of the project is done first, or there just actually isn't so much to do with the task(s) as it was assumed beforehand.
Most frequent anti-patterns

- **Train the Trainer**
  - Training an entire team in some technology can be more expensive, time-consuming and less effective than training just a few individuals who can afterwards train the rest of the team.
  - **Cause**
    - students used technologies and programming languages that they were not familiar with.
Most frequent anti-patterns

- Hero culture
  - Hero Culture anti-pattern arises when the development process relies upon the heroic efforts of a small number of capable individuals striving to achieve a successful outcome
  - Cause
    - Several students either did not dedicate an adequate amount of time to the project or quite the course.
Most frequent anti-patterns

• Untested but Finished
  ▫ Untested but finished occurs when the project is lagging in implementation, and the apparent cause is usually a programming bottleneck - too few or too slow programmers.
  ▫ Cause
    • Managers need to demonstrate progress or enlarge their span of influence; at worse, they want to be able to pretend to make progress.
Generic Anti-patterns Causes

• Several anti-patterns share the same generic causes due to the nature of students’ projects.
  ▫ Overall lack of experience of students
  ▫ Difficulties in using tools
  ▫ Low motivation of team members
Anti-patterns Consequences

- Anti-patterns had unfavourable effects on students’ projects and, by extension, on their deliveries.
  - Almost all projects suffered serious delays
  - Not able to follow their initial schedule.
  - Several teams changed multiple times the scope of their system due to disagreements.
  - In many projects, members spent time working on something that was drop in the end.
  - The quality of the end product was poor and the system did not have the expected functionality.
Difficulties with Anti-patterns

- Anti-patterns, documented in literature and the Web, are numerous
- Lack of homogeneity among different anti-patterns’ templates
- In many cases the severity of each anti-pattern is not explicitly stated
- Anti-patterns rarely appear in isolation. They are inter-related.
Conclusion and Future Work

• The use of SPARSE in students’ projects and the evaluation of its efficacy and effectiveness

• We should train future project managers appropriately so as to be able
  ▫ to identify anti-patterns
  ▫ to avoid them and
  ▫ to resolve them

• SPARSE is a specialized knowledge-based expert system for the anti-pattern detection process.

• SPARSE will assist students by proposing directly related, but also semantically retrieved anti-patterns, according to a list of visible symptoms that may exist in their projects.
Thank you for your attention!