Challenges and Working Solutions in Agile Adaptation: Experiences from the Industry

Özden Özcan-Top
Graduate School of Informatics,
Middle East Technical University

Co Authors: Onur Demirörs, Fergal McCaffery
Agile Software Development

- Agile approaches are characterized with
  - Providing fast feedback
  - Favoring adaptive and exploratory practices
  - Self-organization, collaboration, communication
  - Delivering working software to customer through short, time-boxed iterations
  - Minimizing bureaucracy
  - Balancing up-front work and just-in-time work
  - Embracing change
Motivation for agility assessment

- Main concerns or organizations
  - How far they are to be «agile»?
  - How can they become more agile?
- Agile mindset/practices were misinterpreted
What do most people get "wrong" about Agile

- Becoming agile requires significant changes

«A lot of people are still trying to "do" Agile instead of becoming Agile. It's not about getting trained or buying a fancy new board—it's about changing the way you think.»

John Hughes, Strategist & Agile Coach, Blackstone Technologies
Motivation for agility assessment

- Major concerns or organizations
  - How far they are to be «agile»?
  - How agile they can be?
- Agile concepts were misinterpreted
- Agile was used as an excuse for being undisciplined
- Fundamental need for organizations
  - to assist them in adopting agile methods/practices
  - to guide them for improving their agile capability
The Solution: AgilityMod

- The structure of AgilityMod conforms to ISO/IEC 15504
  - to create common basis for performing an assessment and
  - To present the results using a common rating scale

- The model is independent from any specific agile model
- It can be applicable in any domain
The Solution: AgilityMod

LEVEL 3: EFFECTIVE

LEVEL 2: LEAN

LEVEL 1: AD-HOC

LEVEL 0: NOT IMPLEMENTED

AGILITY REFERENCE MODEL
AgilityMOD

Agile Practices and Processes
Agility Levels

- Not Implemented
  - practices either are not achieved or partially achieved

- Ad-Hoc
  - practices are achieved
  - transition attempts to agile
  - inconsistency in agile implementation

- Lean
  - iterative and incremental
  - effective communication
  - balance is achieved
  - minimum ceremony

- Effective
  - agile engineering methods/practices
    - tools
  - learning and improvement
  - agile engineering methods/practices
Application of a software agility assessment model – AgilityMod in the field

Özden Özcan-Top a, Onur Demirors b, c
Case Study

- We performed a multiple case study with eight cases
- We observed the applicability of AgilityMod for the identification of agility gaps in software projects and also to identify strengths and the weaknesses of the Model.

Application of a software agility assessment model – AgilityMod in the field

Özden Özcan-Top, Onur Demirors
Focus of the paper

• Revealing the agile adaption challenges and lessons learnt from the most successful cases.

• We focus on two cases that had achieved the highest agility levels from the eight cases:
  • Case G
  • Case C
Case Descriptions

• Case G is an e-government project
  • providing solutions to 40 foundations which are located in different cities of Turkey and with approximately 25 million Turkish citizens.
  • Case G includes 21 employees divided into four teams which report to a project manager and an assistant project manager.

• Case C is a digital advertisement sharing platform Project
  • It is in use and new versions of the product are being deployed continuously.
  • The purpose of the project is to ensure the security of the advertisements and to deliver harmless and focused advertisements to end users.
  • The project includes 22 employees.
Challenges Faced and Working Solutions / Having no on-site customer representative

- In Case G, the Product Owner (PO) lives in the United States, while the rest of the team reside in Turkey.
  - However, the product owner communicates with the program managers regularly (3 to 5 times in a week) over teleconferencing, despite the 8 hours difference.
  - The PO does not only communicate with the program managers but also with the scrum masters and the developers when further clarification is required for the backlog items.

*distance is not an excuse for limited communication with customer/product owner*
Challenges Faced and Working Solutions / Varying levels of granularity for user stories

- In Case C, a well working process has been implemented for this challenge.
- The teams use two approaches to decide on the optimum granularity level for user stories.
  - Story points estimation
    - A user story is not included in a sprint, if its size is above a threshold
  - Acceptance criteria
    - Definability of the acceptance criteria is an indicator of a well-defined user story
Challenges Faced and Working Solutions / Growth of product backlog at a inconstant pace

• The product backlog had not grown in a constant pace.
• The issue arose due to communication problems among the PO and the program managers.
• Once they sensed the reason for the problem, they established a *communication matrix* that had to be updated whenever the PO and the program managers communicated with each other.
• It was shown that there is a correlation between the growth of the product backlog and the numbers in the communication matrix.
• Conduct of *regular product backlog grooming meetings*
Challenges Faced and Working Solutions / Nonfunctional retrospective meetings

• Retrospective meetings are one of the ways to transform good teams to great teams.
• They may easily turn into useless meetings.
• Solution:
  • Open action items for the issues and assign the items to team members using the Jira tool.
  • Specify a team quality criterion based upon the percentage of closed retrospective issues in Jira.
  • No new items can be suggested before closing the previous ones
Challenges Faced and Working Solutions / Problems on motivation and software quality

• The Case G team members had suffered from high personnel turnover in the testing team.
• They were in a continuous “fire-fighting” reactive state, because of the bugs found in released versions of the product G.
• The problem mentioned above was due to decrease in the motivation levels of the testers
  • successful, experienced and talented developers assigned to black-box manual testing roles.
• Solution:
  • quitting manual testing and abolishing the test team.
  • they were asked to code the automated unit tests
  • collaborative work and adopting shared responsibility
Challenges Faced and Working Solutions / Ability to manage technical debt

- Technical debt is evitable in software development, when team needs to develop quick solutions or hotfixes

- Solution
  - assigning the responsibility of recovery from technical debt to the person who created it
  - following the progress of such recoveries via a tracking system such as Jira
Unresolved Challenges -1

Identification of the dependencies among design elements for change management

- Knowing the relationship between design elements has a significant impact on identification of changes within an existing software system
- Teams mostly overlook and rely on personal experiences for change impact analyses until the system grows to an unmanageable size.

- Problem:
  - The impact of new requirements on modules and lower level module components were evaluated based on personal experiences.
  - Finding effective solutions for establishing traceability.
Unresolved Challenges -2

The efficiency of the code comments

- Code comments are significant especially for the living software systems where a policy of little documentation is applied.
- Source control systems do not allow developers to check-out code parts without comments.
- But the efficiency of the code comments is not evaluated.
- Problem:
  - There is a need for a mechanism to evaluate efficiency of code comments to increase the clarity of the code, especially at the maintenance phase of a software development life cycle.
Thanks for your attention

ozdenoz@metu.edu.tr