Requirements Engineering in Agile Development

Presented by: Rafi Alam
Traditional Software Development

- Emphasis on gathering requirements in early phases
  - Eliciting all the requirements followed by high level architecture design
- Documentation Driven
  - Requirements document shall be complete before commencing project
- Inflexible
  - No changes welcomed after project starts
But requirements change...

• Agile development
  • Highly customer driven with involvement throughout the lifecycle
  • Low documentation compared to traditional methods
  • Code-oriented
  • Focus on people than process
  • Adaptation: Late changes are acceptable
  • Daily Scrums: Daily check-ins and cooperation among team
  • Collective responsibility of the team
Agile in nutshell

• Planning
• Frequent Small releases
• Continuous customer feedback loop
• Testing: Highly unit test driven
• Continuous Integration
• Established coding standards
Requirements Engineering in Agile
Requirements Engineering in Agile

• Requirements Elicitation
  • Formulation of backlog by product owner
  • All stakeholders actively participate

• Requirements Analysis
  • Customer meetings for requirements refinement
  • Prioritization in conjunction with stakeholders
  • Feasibility

• Requirements Documentation
  • User Stories are written

• Validation
Requirements Engineering in Agile

• Validation
  • Review meetings with the dev team

• Requirements Management
  • Sprint Planning Meeting.
  • Items in Product Backlog for tracking.
  • Change requirements are added/deleted to/from Product Backlog.
Requirement feasibility

• Discussion with customers
  • How the requirement fits with goals of organization?
  • Does it fall within budget and timeframe?
  • Can it be done with the current headcount?

• Org A
  • Was done by business manager/ program manager.
  • Developers were not involved. Unit architect may join.

• Org B
  • Was done by product owner.
  • Senior developers/ architect involved sometimes.
Requirements gathering

• Interviews
  • Customers being asked some specific questions

• Discussion and brainstorming
  • Open meetings with stakeholders to get the needs
  • Generate new and useful ideas. Translate to features

• Use Case Analysis
  • Discuss the flow/ interaction of user with the system

• Org A
  • Was done by business manager/ program manager

• Org B
  • Was done by product owner
Requirement Analysis

• Analyse the list of candidate requirements
  • Validate to check if the requirements are unclear, ambiguous, incomplete etc.
  • Prioritization
  • Whiteboard modelling
  • Joint application development

• Org A: Depends on size of application and timelines
• Org B: Mostly done in planning meeting. No JAD
Requirement Analysis

• Org A
  • In JAD, multiple teams sat together
  • Security team, UX team
  • Much robust security and UX requirement guidelines

• Org B
  • Self contained team with UX and security known developers
  • No organization wide guidelines but following the general standard practices
Validating Requirements

• Manual review
  • Very similar to analysis but read by both developers and stakeholders

• Unit testing
  • Requirements converted to unit tests. Test Driven Development

• Prototyping
  • Starts with minimum viable prototype

• Org A and Org B: Both adopted these methods
Managing requirements

- Creating a process flow/ framework for interaction between all stakeholders
- Using team software
- Using automated acceptance tests by customers
- Trace milestone artefacts and state of the project
Managing Requirements

- Change Control
  - Proposing changes
  - Analyzing impact
  - Making decisions
  - Incorporating
  - Measuring requirements stability

- Version Control
  - Identifying requirements document versions
  - Identifying individual requirement revisions

- Requirements Tracking
  - Defining links to other requirements
  - Defining links to other system elements

- Requirements Status Tracking
  - Defining requirement statuses
  - Tracking requirements that have each status
Scrum in RE

• Product owner/ Program manager has a continuous role in requirements gathering
• Weekly sprint demo to customers by PO/ PM
• Modest change conveyed during daily sprint stand-ups
Software Involved - TFS

![Fabricam Fiber Team Sprint 1 Board](image)

<table>
<thead>
<tr>
<th>Assigned To</th>
<th>Remaining Work</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jamal Hartnett</td>
<td>22</td>
<td>Design welcome screen</td>
</tr>
<tr>
<td>Jamal Hartnett</td>
<td>15</td>
<td>Change background</td>
</tr>
<tr>
<td>Christie Church</td>
<td>18</td>
<td>About screen</td>
</tr>
<tr>
<td>Raisa Pokrovskaya</td>
<td>8</td>
<td>Standardize on form</td>
</tr>
<tr>
<td>Christie Church</td>
<td>24</td>
<td>Slow response on welcome page</td>
</tr>
<tr>
<td>Christie Church</td>
<td>24</td>
<td>Fix performance issue</td>
</tr>
<tr>
<td>Raisa Pokrovskaya</td>
<td>57</td>
<td>Add an information form</td>
</tr>
<tr>
<td>Raisa Pokrovskaya</td>
<td>25</td>
<td>Auto-complete user's information</td>
</tr>
<tr>
<td>Raisa Pokrovskaya</td>
<td>32</td>
<td>Auto-save</td>
</tr>
</tbody>
</table>

**Work details**

- **Team**
  - (144 of 169 h)

- **Work By: Activity**
  - Development
    - (144 of 169 h)

- **Work By: Assigned To**
  - Christie Church (Fabrikam)
    - (42 of 39 h)
  - Jamal Hartnett (Fabrikam)
    - (37 of 78 h)
  - Raisa Pokrovskaya (Fabrikam)
    - (65 of 52 h)
Software Involved - Jira
Takeaways: Improving RE in Agile

• Customer involvement
• Frequent Releases
• Less documentation
• Minimum viable product
• Smaller focused agile teams
Takeaways: Improving RE in Agile

• Drawbacks
  • Traceability is poor
  • Non functional requirements are pushed back in priority
  • Late requirements can put pressure on developers
  • Keeping team “alive”
  • Not all agile are some – Some teams takes break due t organizational demand
Thanks! And Discussions