Agile Methodologies and Scrum

Siddharth Gangadhar
Dr. Prasad Kulkarni
Department of Electrical Engineering & Computer Science
Lab Presentation

siddharth@ku.edu
prasadk@ku.edu

16 September 2015
Agile Methodologies and Scrum

Outline

• Introduction to SDLC
• Waterfall Model
• Agile Methodologies
• Scrum Methodology
• References
Agile Methodologies and Scrum

Introduction to SDLC

- Introduction to SDLC
- Waterfall Model
- Agile Methodologies
- Scrum Methodology
- References
Introduction to SDLC

SDLC Phases [I]

- Requirement analysis
  - Who will use the system?
  - inputs and outputs to the system
  - Creation of a requirement specification document

- System design
  - responsible for system requirements for the product
  - example resources required, OS, system level dependencies

- Program design
  - software architecture finalized in this phase
Introduction to SDLC

SDLC Phases [II]

• Coding
  – longest phase of SDLC

• Unit and Integration testing
  – unit testing
    • small piece of code is doing what is intended
    • intended for use by programmer
  – Integration testing
    • demonstrate different parts of system work together
    • intended to cover whole applications

• System testing
  – black box testing
Introduction to SDLC
SDLC Phases [III]

- Acceptance testing
  - also known as UAT (User Acceptance Testing)
  - tests to check if requirements are met
- User and maintenance testing
  - performance testing for further inspection or repair
  - stress testing an example
- Operation and maintenance
  - software system in production
  - stage includes periodic updates and bug fixes
Waterfall Model

• Introduction to SDLC
• Waterfall Model
• Agile Methodologies
• Scrum Methodology
• References
Waterfall Model

Waterfall Model Schematic

1. Requirements
2. Design
3. Implementation
4. Verification
5. Maintenance

Product requirements document
Software architecture
Software

[WM-wiki]
Waterfall Model

Waterfall Model Characteristics

- Requirements are well documented and fixed
- Product definition is stable
- Technology understood and not dynamic
- No ambiguous requirements
- Ample resources with expertise available
- Typically short projects
Waterfall Model
Pros and Cons

• Pros
  – simple and easy to understand and use
  – easy to manage
  – sequential processing of phases
  – process and results well documented

• Cons
  – high amount of risk and uncertainty
  – now working software produced until late cycles
  – not a good model for changing requirements
  – integration is done very late
Agile Methodologies and Scrum

Agile Methodologies

- Introduction to SDLC
- Waterfall Model
- Agile Methodologies
- Scrum Methodology
- References
Agile Methodologies

Historical SDLC Information

![Bar Chart: Project Success is Rare]

<table>
<thead>
<tr>
<th>Year</th>
<th>Failed</th>
<th>Challenged</th>
<th>Succeeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>24%</td>
<td>44%</td>
<td>32%</td>
</tr>
<tr>
<td>2006</td>
<td>19%</td>
<td>46%</td>
<td>35%</td>
</tr>
<tr>
<td>2004</td>
<td>15%</td>
<td>51%</td>
<td>34%</td>
</tr>
</tbody>
</table>


Average cost overrun: 45%
Time overrun: 63%
Functionality delivered on average: 67%
Agile Methodologies

Agile Principles

• Individuals and interactions
  – over processes and tools

• Working product
  – over comprehensive documentation

• Customer collaboration
  – over contract negotiation

• Responding to change
  – over following a plan
Agile Methodologies
Classical vs. Agile approach

- Timelines vs. no timelines
- Managed vs. unmanaged self disciplined teams
- Fixed vs. no fixed budget approach
- Predictable fixed deliverables vs. constantly evolved
- Multiple location groups vs. co-location
- non customer vs. customer inclusive team
Agile Methodologies

Agile Methods [I]

- **Scrum**
  - iterative and incremental agile development methodology
  - more later

- **XP**
  - frequent releases in short development cycles
  - pair programming
  - continuous integration

- **RUP**
  - develop iteratively; manage requirements
  - use components; model visually
  - verify quality; control changes
### Agile Methodologies

**Agile Methods [II]**

- **Kanban**
  - got inspiration from Toyota workline
  - focus only on one work at a time
  - continuous improvement through
    - control charts
    - cumulative flow diagrams

- **FDD**
  - develop overall model
  - build feature list
  - plan by feature, design by feature
  - build by feature
Agile Methodologies and Scrum

Scrum Methodology

- Introduction to SDLC
- Waterfall Model
- Agile Methodologies
- Scrum Methodology
- References
Scrum Methodology

Scrum Terminology

- **User stories**
  - requirements written from user perspective
- **Product backlog**
  - collection of user stories that need to be addressed
- **Sprint**
  - fixed length work period where backlog items are worked on
- **Sprint planning session**
  - review and prioritize each backlog item
- **Sprint review session**
  - work completed reviewed
Scrum Methodology

Scrum Roles

• Product Owner
  – develops and maintains product backlog
  – presents and explains product backlog to team

• Scrum Master
  – sets up and conducts meetings
  – representative to management and team

• Team
  – full authority during a sprint
  – responsible for the estimation of work
Scrum Methodology

Scrum Tools

• **www.icescrum.com**
  – platform for managing scrum for projects

• **Installation**
  – comes with an open source license
  – [https://www.icescrum.com/pricing/#standalone](https://www.icescrum.com/pricing/#standalone)

• **JIRA**
  – an Atlassian product
  – issue tracking and project management software
  – $10 for 10 users
Scrum Methodology

Scrum Demo

- Create a project
- Create user stories in Product Backlog
- Create sprint tasks in Release Plan
- Create actors for specific rules
- Create features to work on
  - As part of product backlog
Agile Methodologies and Scrum

References

- Introduction to SDLC
- Waterfall Model
- Agile Methodologies
- Scrum Methodology
- References
Agile Methodologies and Scrum

References

- WM-wiki
- agile-101
  http://agilerichmond.com/docs/agile101.ppt
END OF FOILS

QUESTIONS???