

**Requirements and Specification, ESOF 328, Spring 2020**

**“Agile Projects” (Chapter 20)**

**“Tools for requirements engineering” (Chapter 30)**

**Jan. 13**

**Agile Projects, Chapter 30**

Agile development - adaptive and change-driven

Non-agile - predictive and plan-driven

Waterfall software development process model:

Specification

Design

Implementation

Test

Maintenance

Other process models:

Incremental

Spiral

Prototyping (evolutionary)

Throw-away prototyping

Agile process models:

Scrum – becoming a de facto standard

Extreme Programming (XP)

Lean Software Development

Agile Unified Process

Crystal

Agile process:

Break development into short cycles called iterations

Scrum calls these “sprints”

1 week to 1 month

Customers are involved during the entire process, not just for the requirements

Less documentation, “the less documentation the better”

Often use “user stories” which are less detailed than use cases or functional requirements

Backlog contains a list of requests which are constantly being prioritized

Expect change

Glossary (back of text)

User story – A format to capture user requirements on agile projects in the form of one or two sentences that articulate a user need or describe a unit of desired functionality, as well as stating the benefit of the functionality to the user

Epic – A user story on an agile project that is too large to implement in one development iteration. It is subdivided into smaller stories that each can be fully implemented in a single iteration

Use case – A description of a set of logically related possible interactions between an actor and a system that results in an outcome that provides value to the actor. Can encompass multiple scenarios

Feature – One or more logically related system capabilities that provide value to a user and are described by a set of functional requirements

Goal: Identify the lowest level story that still aligns with the business requirements, yet the team can deliver within an iteration, and it provides value to the customer. Called “Minimum Marketable Feature” (MMF).

### **Tools for requirements engineering, Chapter 30**

“Use a tool when you already have an approach that works but that requires greater efficiency. Don’t expect a tool to compensate for a lack of business analysis and requirements engineering process, training, discipline, or experience” Page 504

“... no tool will be able to tell you if a requirement or a model element is missing, logically incorrect, or unnecessary... tools enable BAs to represent information in multiple ways and to spot certain types of errors and omissions, but they don’t eliminate the need for thinking and peer review.” Page 506

Requirement development tool categories:

- Elicitation – mind map tools, audio pens, etc.
- Prototyping – many tools, can just use PowerPoint
- Modeling – many tools, Visio is one

Requirement management tools are more developed than requirements development tools.

Benefits of requirement management tools (Pages 507-508):

- Manage versions and changes
- Store requirements attributes
- Facilitate impact analysis
- Identify missing and extraneous requirements
- Track requirements status
- Control access
- Communicate with stakeholders
- Reuse requirements
- Track issue status

- Generate tailored subsets