

Week 2

1. Plan Driven Requirements
 - Gather requirements is typically the first step in a plan driven SDLC
 - Business Analyst interviews the customer to extract features required from the system
 - The system **shall** allow the user to enter her name, address, and email
 - The system **shall** be available 24x7x365 with 0.99% reliability
 - Business Requirements Document (BRD)
 - Formal contract between the customer and the team delivering the product
2. BRD
 - All requirements are gathered and reviewed by the business analysts before handing over to development
 - Little or no discussion between customers and developers
 - Describes all the features needed by the customer
 - Functional/non-functional requirements
 - Limitations
 - Assumption of completeness of document
 - Difficult to change
 - Developers don't actually talk to the customer
3. Use Cases
 - Relationship between RUP and agile?
 - RUP is what led to agile (motivated it)
 - Part of UML and RUP
 - Each use case describes:
 - A scenario
 - The actors in the scenario
 - Actors may be people or system
 - A sequence of actions between the actor and the system
 - An observable result of value to a particular actor
 - Actor's intention/what does the actor want?
4. Use Cases vs. User Stories
 - Use Cases: RUP approach for capturing requirements
 - User Stories: Agile approach for capturing requirements
5. User Stories: Agile approach
 - Customers communicate their needs via short statements
 - Customers provide the user stories with help from developers
 - Each user story is a reminder for the customer and developer to discuss the issue
 - Each statement describes the goal of an actor:
 - “As a user, I want to ...”
 - Customer should decide priority of each user story
6. 3 C's of User Stories
 - *The Card*:
 - Best understood as an invitation to conversation, must address the “who”, “what”, and “why” of the story
 - The card usually follows the format similar to one below
 - *The Conversation*:
 - The collaborative conversation facilitated by the product owner which involves all stakeholders and the team
 - As much as possible, this is in-person
 - Conversation is where real value of story lies and the written card should be adjusted to reflect the current shared understanding of this conversation
 - * Customer resolves ambiguity

- * Dev estimates the effort
- *The Confirmation:*
 - Product owner must confirm that the story is complete before it can be considered done
 - Team and product owner check the doneness of each story in light of the teams current definition of done
 - Specific acceptance criteria that is different from the current definition of done can be established for individual stories, but the current criteria must be well understood and agreed by the team
 - All assoc. acceptance tests should be in a passing state
- 7. User Story components
 - Title
 - Acceptance test
 - Priority
 - Story points
 - Description
- 8. Estimating time for user stories
 - Use previous experience, similar stories on projects, or use a consensus process to compare estimates
- 9. Limitations of user stories
 - User stories come from XP with small teams and engaged customers
 - Lack of look ahead
 - Lack of context
 - Lack of completeness
- 10. When to use which?
 - User stories are more flexible and easier to write
 - Ideal for small projects:
 - * Where dedicated customer responds to questions
 - * Where a small team can deliver the entire solution
 - Use cases are more rigorous
 - May be better for large projects
 - Where risks are higher