August 7, 2019

Agile Requirements with User Stories

Trainer: Thy Vo

DXC.technology

Introduction

- Your name
- Your role
- Your background and experience in
  - Software Development
  - Software Requirements
- What do you want to get the most out of this course ?

Course Objectives

- At the end of the course, you will have acquired sufficient knowledge to:
  - Read and Understand requirements in Agile projects
  - Identify User Story/EPIC/Theme
  - Split user story
  - Differentiate between the requirement in Agile projects compared with non-agile projects

Agenda

- Requirement Overview
- User Stories
- Other Requirement Artifacts

Course Audience and Prerequisite

- Those who would like to understand more about how to read, analyze and document requirements in Agile projects
- To benefit most from the course, you should:
    - Have knowledge about Agile process

Duration and Course Timetable

- Course Duration: 6 hours
- Course Timetable:
  - 2 sections
  - Break 15 minutes in each section

Further References

- Agile Course(s) on DXC University:

  1. Planning an Agile Software Development Project

     Course ID: *“sd_agsd_a02_it_enus”*

- 2. JIRA AGILE - An Introduction for Agile Project Management

  - Course ID: *“_scorm12_csc_gbs153”*

Course Administration

- In order to complete the course you need to:
  - Sign in the Class Attendance List
  - Participate in the course
  - Pass Final Test: 7/10
  - Provide your feedback near the end of the class

Requirement Overview

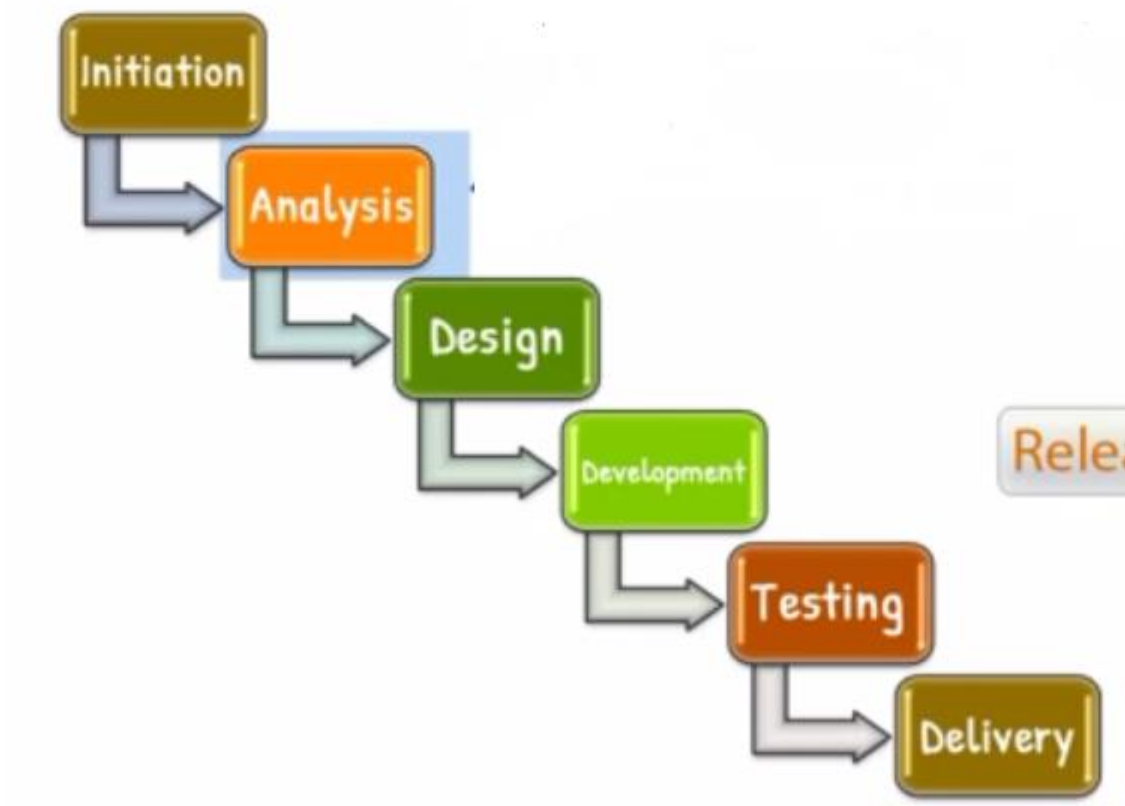
What is a “Requirement”?

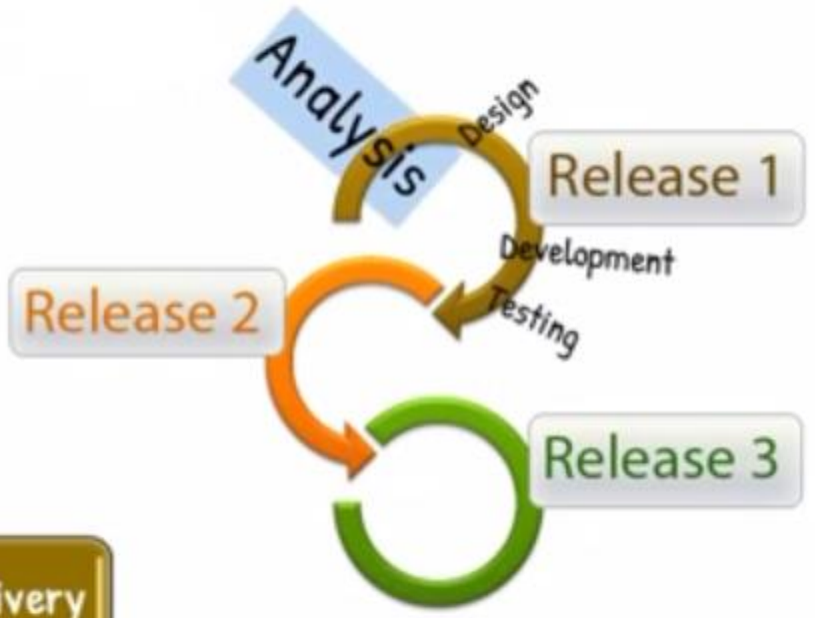
- A requirement is:
  - A condition or capability needed by a stakeholder to solve a problem or achieve an objective (1)
  - A condition or capability that must be met or possessed by a solution or solution component to satisfy a contract, standard, specification, or other formally imposed documents (2)
  - A documented representation of a condition or capability as in (1) or (2)

Requirements in Traditional and Agile Software Development Life Cycle (SDLC)

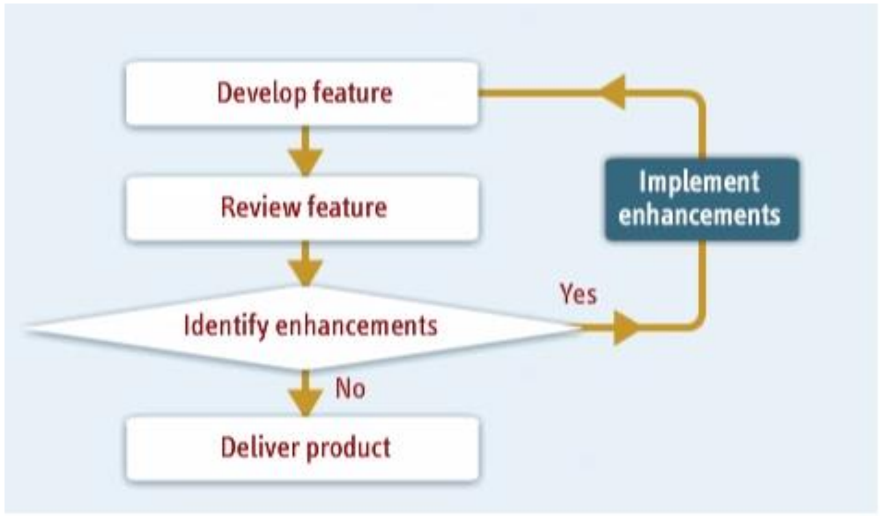
Traditional Process

Agile Process



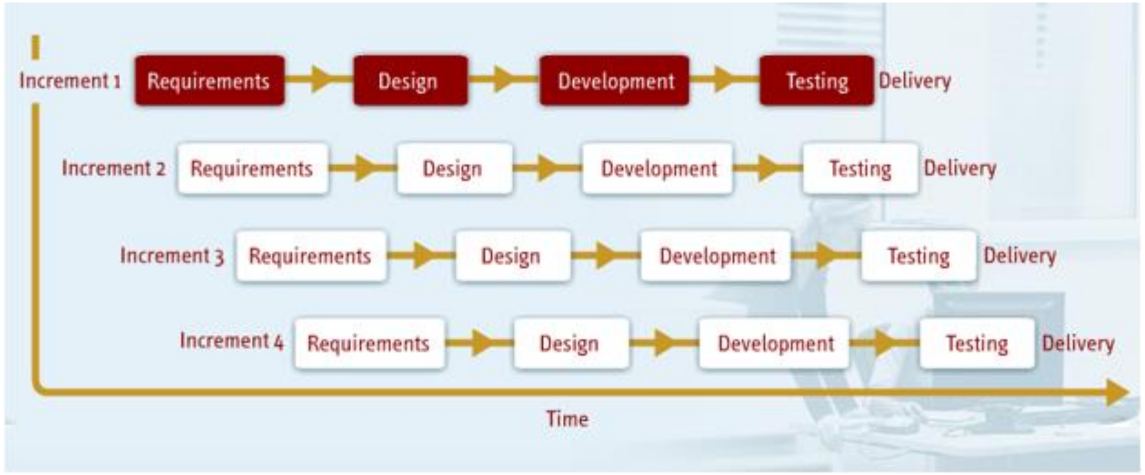


Requirements in Traditional and Agile Software Development Life Cycle (SDLC) (cont.)

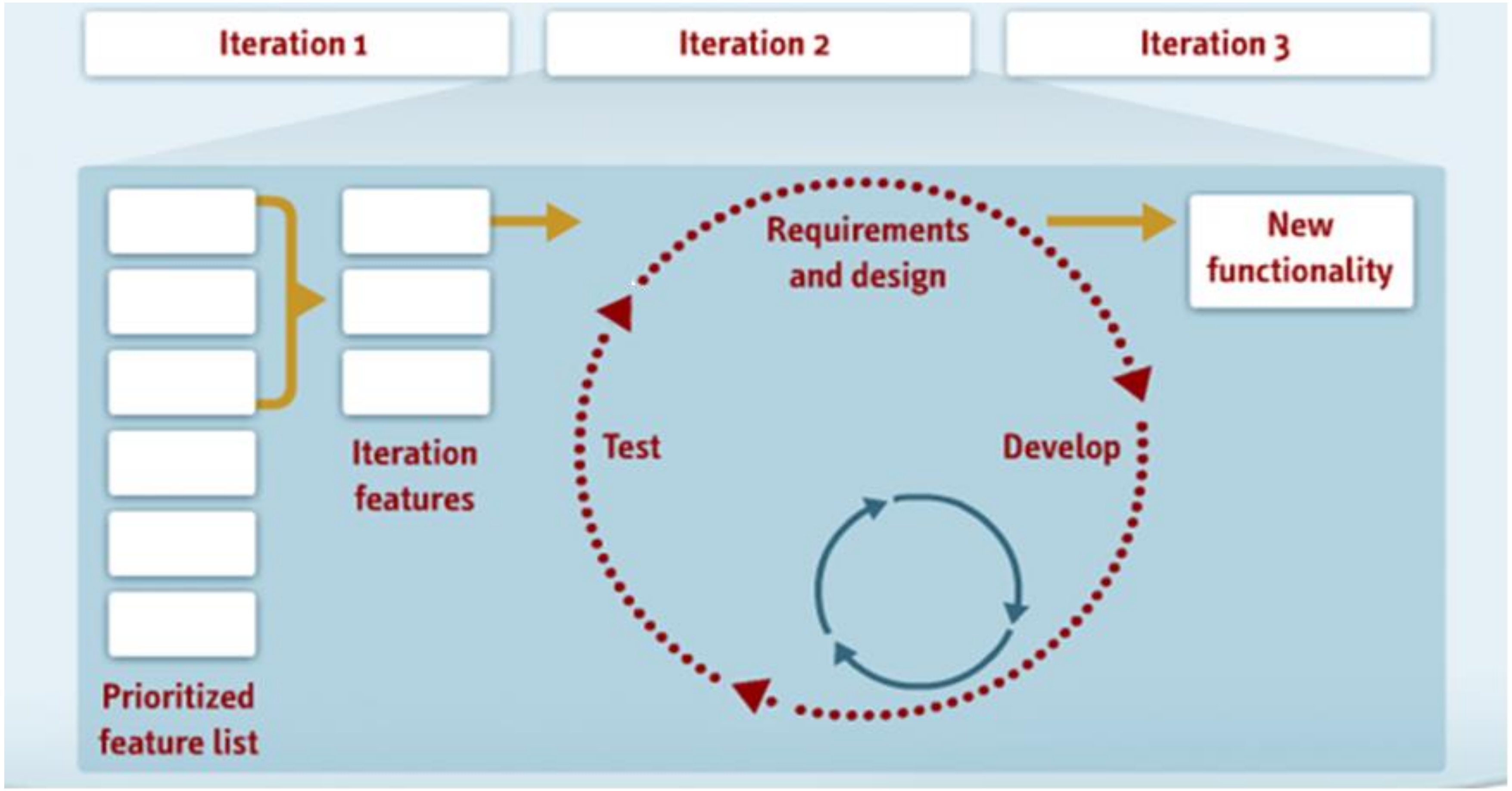


Change Request Workflow

Incremental Process



Requirements in Traditional and Agile Software Development Life Cycle (SDLC) (cont.)



Agile Model

Functional Requirements

- Define WHAT a system suppose to accomplish, a function of a system and its components
- Are supported by non-functional requirements (also known as quality requirements)
- Are expressed in the form "system must do <requirement>“
- As defined in requirements engineering, functional requirements specify particular results of a system
- Functional requirements and agile processes:
  - User Story and scenario

Non-Functional Requirements

- Define HOW a system suppose to do, specify “how well” the “What” must behave
- Quality expectation
- Represent system-level constraints that typically cut across functional requirement during the design or implementation (such as performance requirements, security, or reliability)
- Are expressed in the form "system shall be <requirement>",
- Affect the design and testing of most or all stories in the Product Backlog in Agile or the whole system in traditional projects
- Non-functional requirements and agile processes
  - Improving quality during construction
  - Improving quality during execution

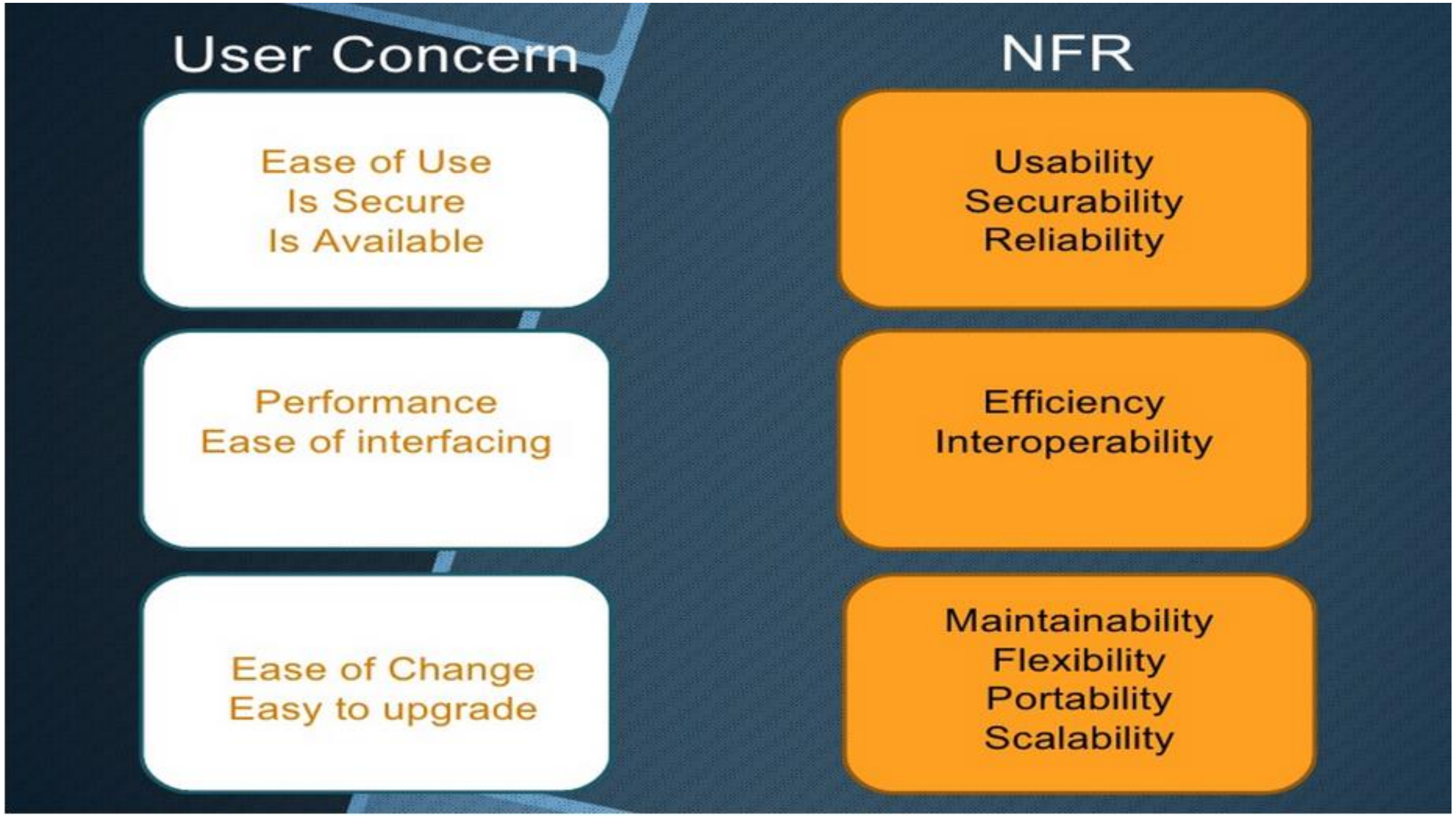
Non-Functional Requirements (cont.)

- If your nonfunctional requirement is not objectively measurable, you needs to revise, rewrite, or expand it
- Measurable objectives: 10,000 transactions per hour, 1 second response time, six packs of beer
- Subject quality: easy to maintain, high quality, good beer => not objective measurable => need to clarify

Non-Functional Requirements (cont.)

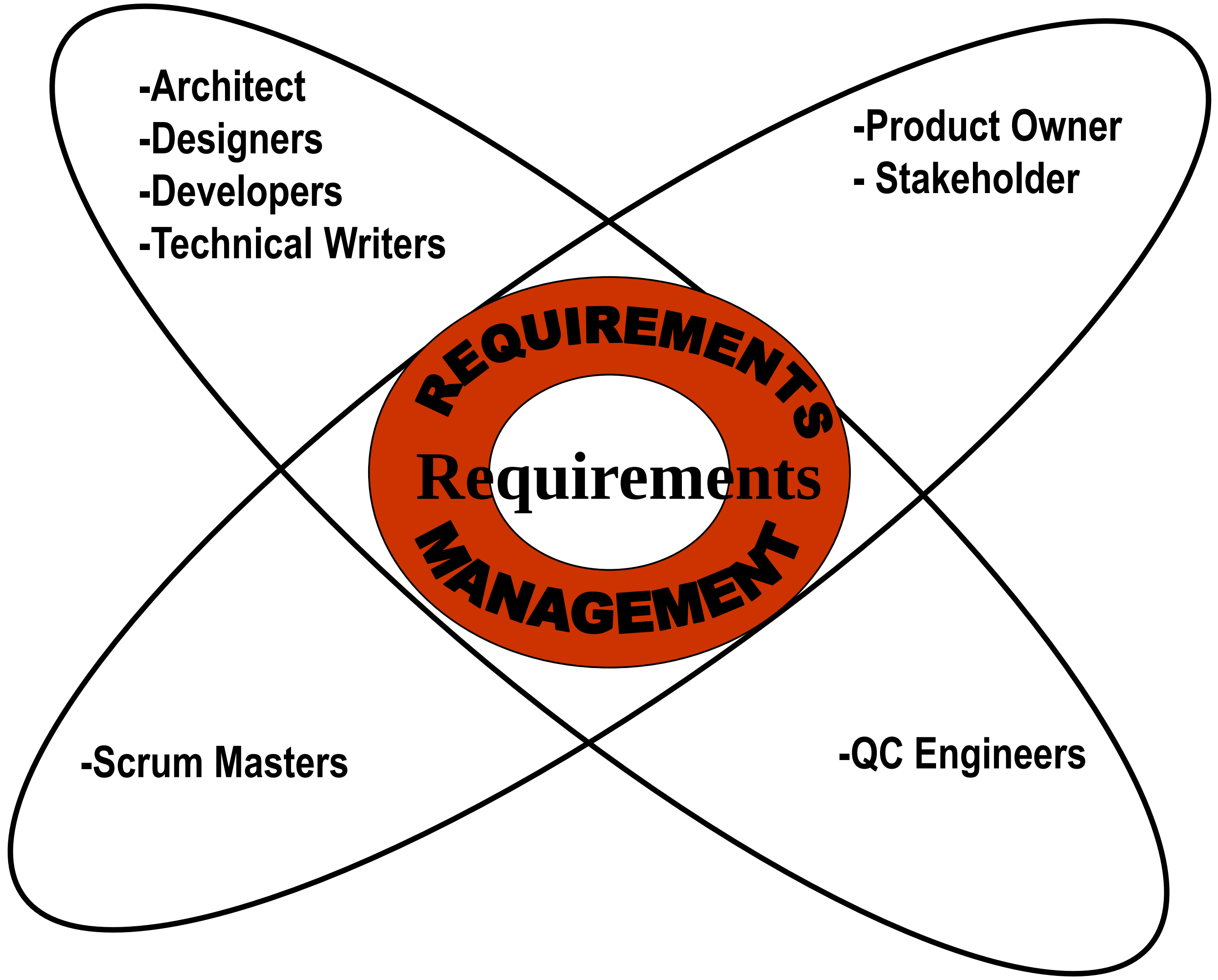
- Non-functional Features:
  - Frequency: how often?
  - Urgency: how quickly application response to user needs
  - Volume: how much data maintain?
  - Accuracy: how precise and timely for data?
  - Usability: what features easy to use by the role?
  - Learnability: how quickly the new user can learn to use application?
  - Flexibility/scalability: how volatile is usage?
  - Reliability: how critical that the app does not fail?

Non-Functional Requirements (cont.)



Mapping Business Concern to Non-Functional Requirements

Who needs to understand requirements?



All project participants need to understand requirements

User Stories

User Stories/ EPIC/ Theme – What are they?

- **Theme** - A collection of features, epics, & stories that describe a broad business purpose

- A template often uses the following type of format for EPIC & User Story:
  - As a <role>, I want <feature> so that <reason>

- **EPIC** – A very large user story that will not fit into a single iteration, does not pass the test for inclusion in an iteration, and will need to be subdivided to be considered

Sample (Updated p34)

Master Data			Theme
As an Administrator, I want to be able to have list of Ships so that I can refer later	As an Administrator, I want to be able to have list of Countries so that I can refer later	As an Administrator, I want to be able to have list of Ship Types so that I can refer later	Epic
As an Administrator, I want to be able to create a Ship so that I can refer later	As an Administrator, I want to be able to create a Country so that I can refer later	As an Administrator, I want to be able to create a Ship Type so that I can refer later	User Story
As an Administrator, I want to be able to update a Ship so that I can refer later	As an Administrator, I want to be able to update a Country so that I can refer later	As an Administrator, I want to be able to update a Ship Type so that I can refer later	
As an Administrator, I want to be able to delete a Ship so that I don’t see the unused ship in all referenced list	As an Administrator, I want to be able to delete a ship so that I don’t see the unused Country in all referenced list	As an Administrator, I want to be able to delete a Ship Type so that I don’t see the unused Country in all referenced list	

User Stories/ EPIC/ Theme – What are they? (cont.)

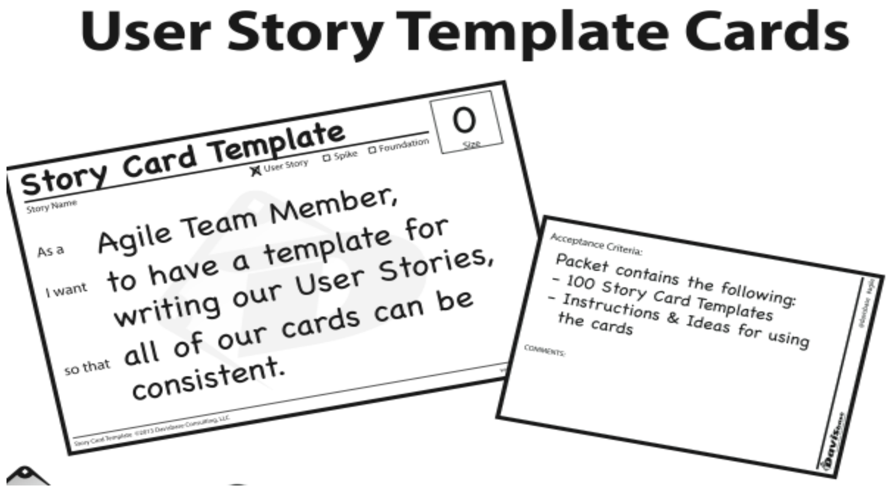
- **User Story** is**:**
  - a convenient format for expressing the desired business value
  - crafted in a way that makes them understandable to both business people and technical people
  - used to provide a great placeholder for a conversation
  - written at various levels of granularity and are easy to refine

User Story

As a..	I want/would like..	So that..
(Who)	(What)	(Why)
Functional User Story		
User	to upload photos to public	I can share photos with others.
User	To view the Trade Ticket Report on mobile devices	offline users can review the ticket
User	receive boarding pass confirmation email after check-in online	I can save time at the airport
Non-functional User Story		
the CTO	the system to use our existing orders database rather than create a new one	we don't have one more database to maintain.
User	the site to be available 99.999 percent of the time I try to access it	I don't get frustrated and find another site to use.

User Stories

- **3Cs**: Card, Conversation, Confirmation
- **Card**:
  - **Who** – specify User Role
  - **What** – what User Role wants to achieve (the goal)
  - **Why** – why User Role wants to achieve the goal (the benefit)
- **Acceptance Criteria** - a list of questions, scenarios that enable the User Role to sign off the story as “done”


User Story Template Cards
Story Card Template
User Story
Spike
Foundation
0
Size
Story Name
As a
Agile Team Member,
I want
to have a template for writing our User Stories,
so that
all of our cards can be consistent.
Acceptance Criteria:
Packet contains the following:
- 100 Story Card Templates
- Instructions & Ideas for using the cards
COMMENTS:

User Stories (cont.)

- **3Cs**: Card, Conversation, Confirmation
- **Conversation**:
  - Ongoing dialog among Product Owner, Stakeholders, and Development Team during Sprint
  - Enable richer form of exchanging information and collaborating to ensure that the correct requirements are expressed and understood by everyone
  - Supplemented by documents

User Stories

- **3Cs**: Card, Conversation, Confirmation
- **Confirmation**:
  - Confirmation information in the form of conditions of satisfaction or acceptance criteria
  - Used by the development team to between understand what to build and test
  - Used by Product Owner to confirm that a user story has been implemented

User Stories – Template and Sample

- User Story Template
- User Story Sample

Gathering Stories

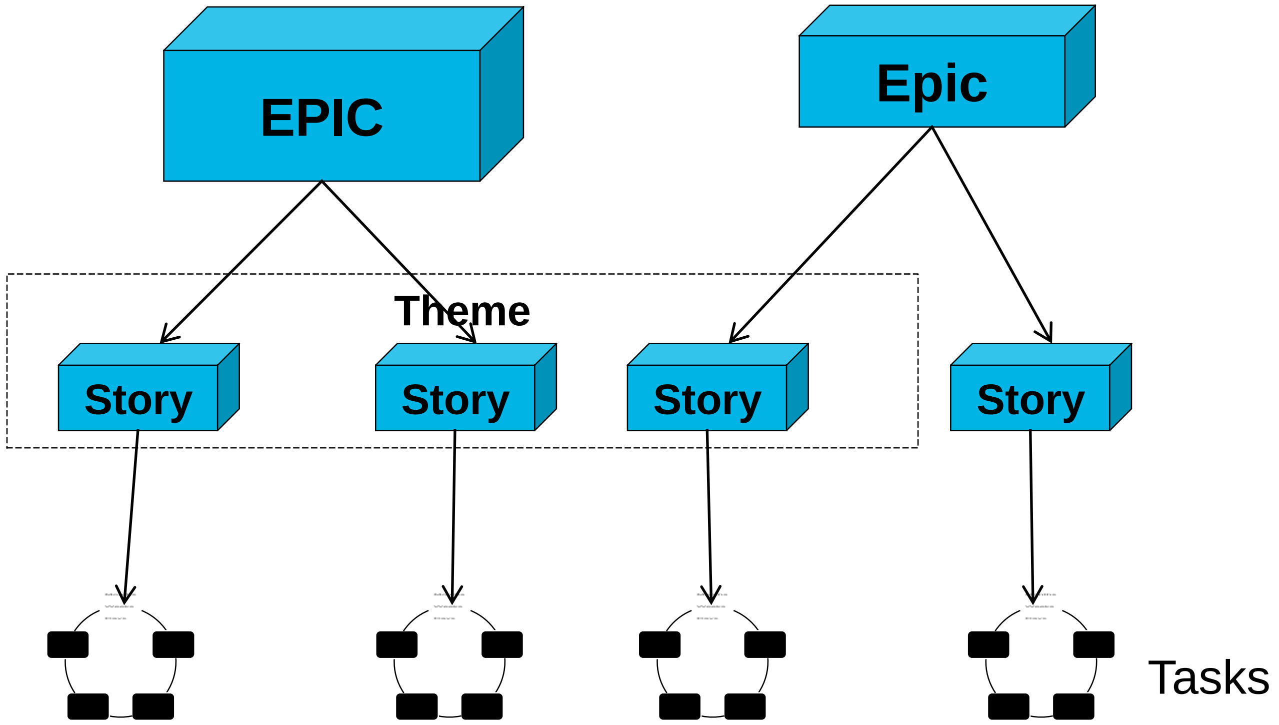
- Involve users as part of the team that is determining what to build and is constantly reviewing what is being built

- **Techniques:**
- **User-Story-Writing Workshop:**
  - Brainstorm desired business value and create user story placeholders for what the product or service is supposed to do

- **Story Mapping**
  - Decompose high-level user activity into a workflow that can be further decomposed into a set of detailed tasks

User Story

- **Split Story/Task**



Split Stories

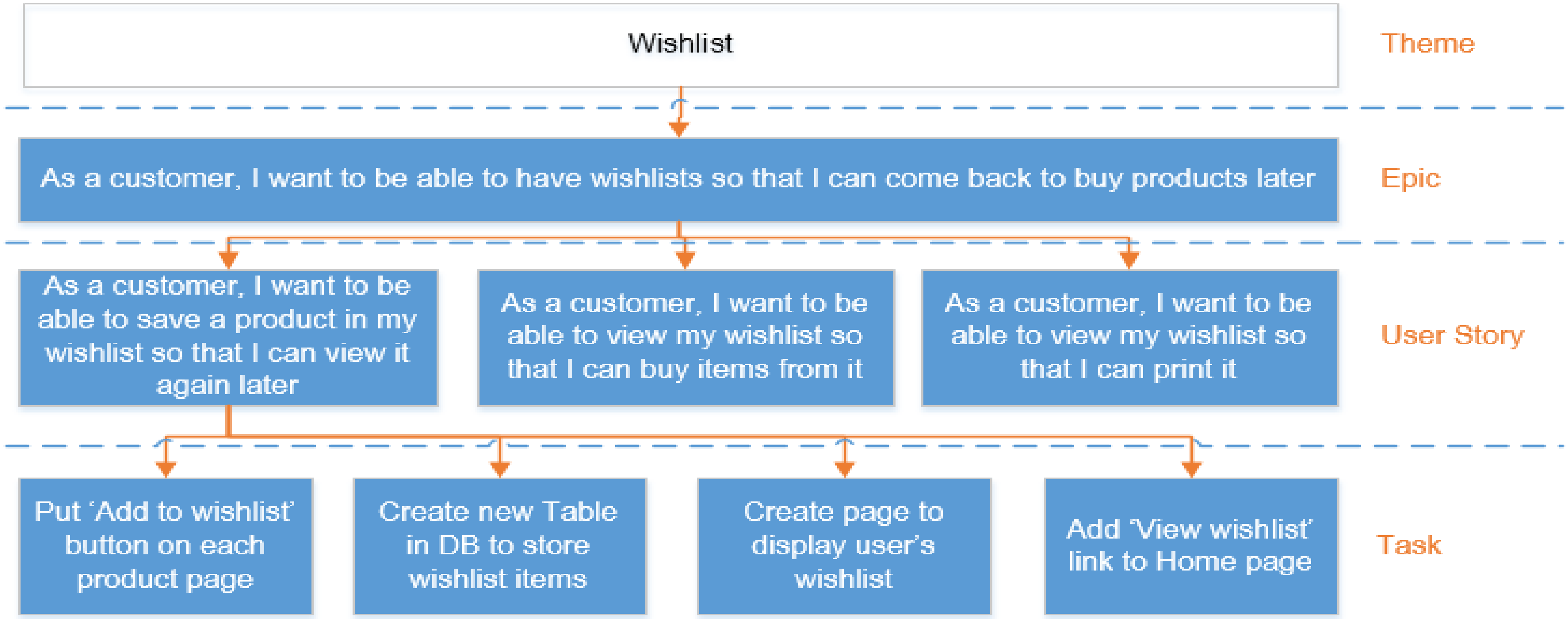
- When:
  - It is too large to fit into a single iteration or Sprint
  - If a more accurate estimate is necessary
  - If smaller stories have different priorities

- How:
  - Split large stories by the type of data that user could enter
  - Split large stories based on the operations that are performed within the story
  - Split large stories into separate CRUD operations
  - Remove cross-cutting concerns such as securities, logging, error handling, and so on and create two versions of the story: one with and one without support for the cross-cutting concern

Split Stories (cont.)

- How:
  - Split large stories by separating the functional and nonfunctional aspects into separate stories
  - Each of new spitted story should be well within the size the team could complete in a two-week Sprint
- Don’t:
  - Split a story into development tasks
- Combine Stories:
  - Combine related stories as that will make it easier to prioritize them, e.g. combine multiple bug reports and treat them as one item

Epic – Sample


Wishlist
Theme
As a customer, I want to be able to have wishlists so that I can come back to buy products later
Epic
As a customer, I want to be able to save a product in my wishlist so that I can view it again later
As a customer, I want to be able to view my wishlist so that I can buy items from it
As a customer, I want to be able to view my wishlist so that I can print it
User Story
Put 'Add to wishlist' button on each product page
Create new Table in DB to store wishlist items
Create page to display user's wishlist
Add 'View wishlist' link to Home page
Task

Assessing the Readiness of Stories for An Iteration

- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimable
- **S**mall (appropriately sized)
- **T**estable

User Stories

- **Independent:**
  - User stories should be deliverable independently of each other
  - Independent stories enable the team and customer to inject small stories into the backlog that can be delivered in timescales aligned to Sprint
  - User stories that exhibit a high degree of interdependence complicate estimating, prioritizing, and planning
  - Write stories in a way that minimizes dependencies
- **Negotiable**:
  - The details of user stories should be negotiable
  - A story will be refined over time and is negotiable up until the point that the story is planned within a sprint

User Stories

- **Valuable:**
  - User stories need to be valuable to a customer. This include technical stories
  - Treat technical stories like any other business-valuable story

- **Estimable**:
  - Stories should be estimable by the team that will design, build, and test them
  - Estimates provide an indication of the size and effort and cost of the stories
  - It is essential that the team is involved in the refinement of stories, in cooperation with the customer and stakeholders, to have solid understanding of the story and be able to create realistic and achievable estimates

User Stories

- **Small:**
  - Stories should be sized appropriately – each a few days in size to fit in Sprint

- **Testable**:
  - Being testable means having good acceptance criteria
  - Stories must include testable criteria

INVEST - Sample

- Not INVEST: TESTABLE

Sample:
As a user, I want the system to be fast, so I don’t have to waste my time to wait for the page loading

Improve:
As a user, I want the web pages should generally load within 2 or 3 seconds, so I can do what I want faster.

INVEST - Sample

- Not INVEST: Independent, Small

Sample:
As a product owner, I want to write game rules, so the player will follow the rule to play game

Improve:
As a newbie game player, I want to know who goes first so we can start the game

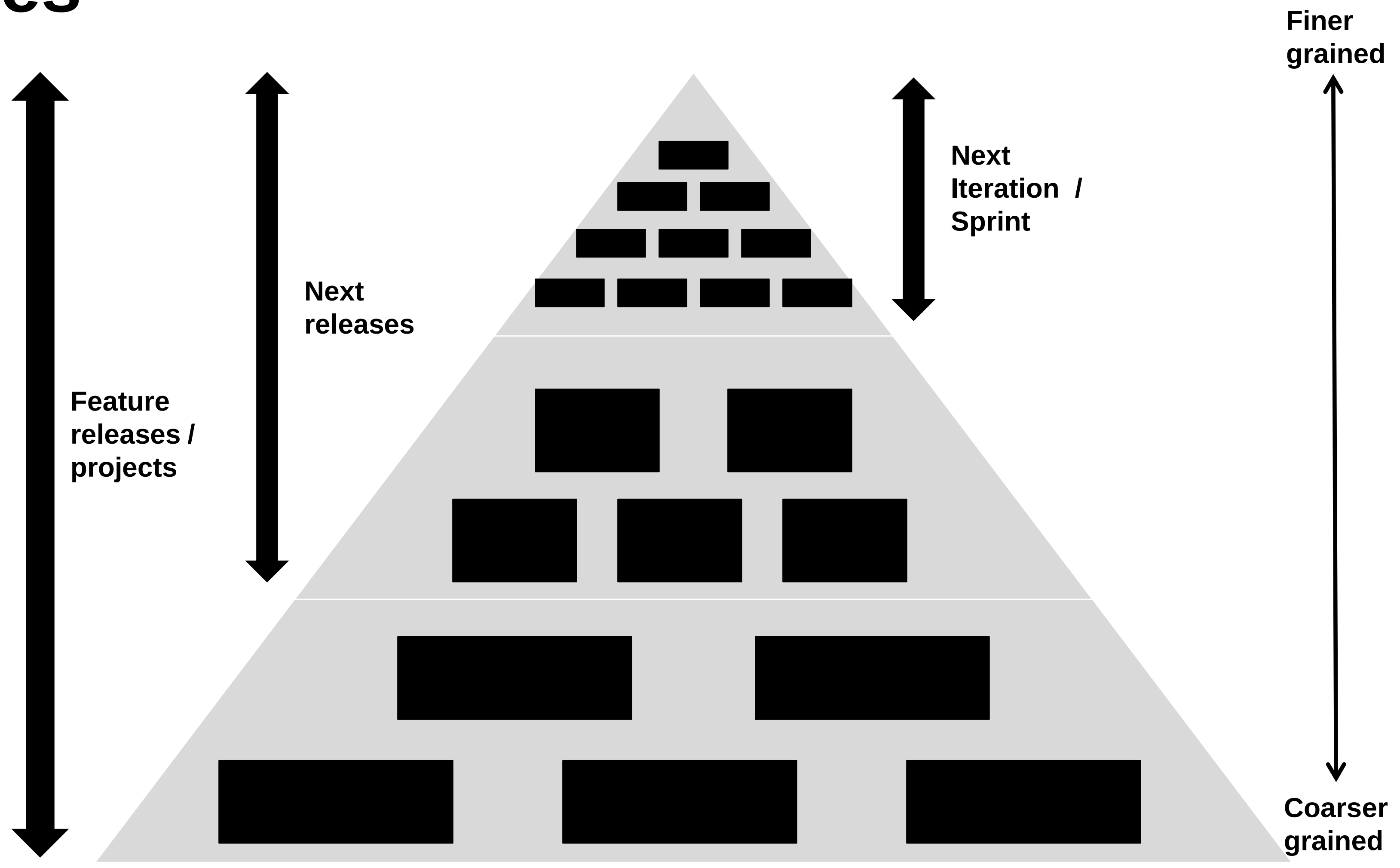
User Stories

- **Backlog refinement:**
  - Stories are continually refined within backlog throughout the whole lifetime of the product
  - Stories should be refined JIT basis for next sprint

- **‘Spike’ stories:**
  - Is a story that drives technical or functional research effort or investigating work
  - Story-driven activity to investigate something specific

- **Planning pyramid:**
  - A feature breakdown structure of parent-child stories may be required when delivering large, complex projects

User Stories



A planning Pyramid contains both coarse-grained and fine-grained stories; the coarse-grained stories are being refined to be fine-grained stories as delivery progress

User Stories

- **Prioritization – a MoSCoW acronym**
  - Arrange stories in a sequence within sprint time-box

- **M**ust have**:**
  - These are stories that must be delivered within sprint time-box

- **S**hould have**:**
  - A story that is very important within a time-box, that will cause significant problems to customer if not delivered

User Stories

- **C**ould have:
  - A story that is very important within a time-box, that may cause some problems to customer if not delivered

- **W**on’t have:
  - Agreed between customer and team that a particular story won’t be delivered “this time”. It might be added to a later time-box or removed completely from PB

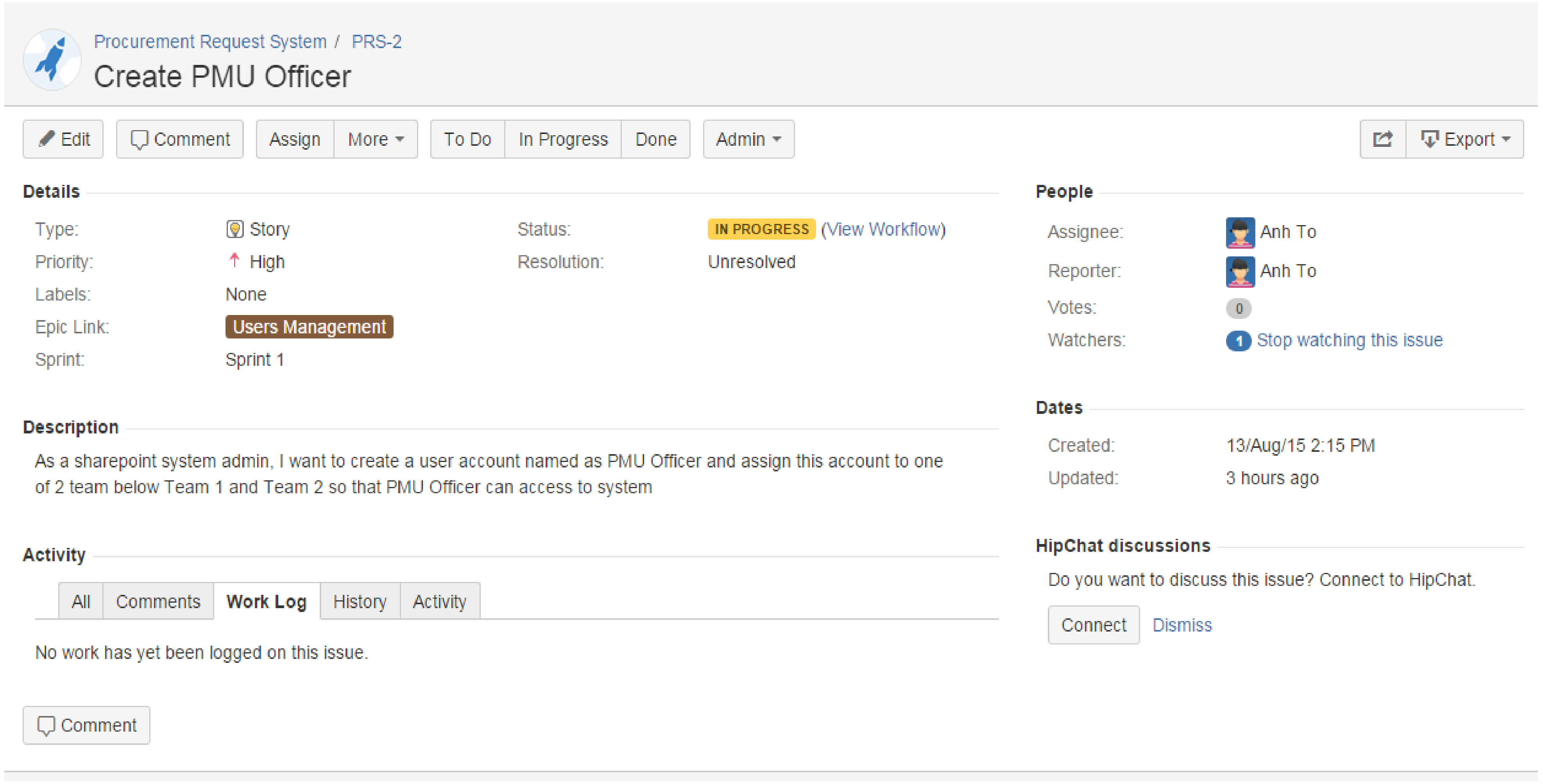
Product Backlog

- A placeholder of requirements and desires from all stakeholders
- A prioritized and emerging list of functional, nonfunctional, architectural, infrastructural, risks elements that required to fulfill the Product Vision
- More granular items kept towards the top, general epics at the bottom
- Product Backlog contents will change over time
- PO is ultimately responsible for the content and state of the Product Backlog, though anyone is able and encouraged to contribute to the Product Backlog
- Each PBI should be small enough to fit into a Sprint and must be clear by specifying the acceptance criteria

Tools in Agile Projects

- JIRA Agile

Jira Agile



Create PMU Officer

Details

Type:	Story	Status:	IN PROGRESS (View Workflow)
Priority:	High	Resolution:	Unresolved
Labels:	None		
Epic Link:	Users Management		
Sprint:	Sprint 1		

People

Assignee:	Anh To
Reporter:	Anh To
Votes:	0
Watchers:	1 Stop watching this issue

Dates

Created:	13/Aug/15 2:15 PM
Updated:	3 hours ago

Description

As a sharepoint system admin, I want to create a user account named as PMU Officer and assign this account to one of 2 team below Team 1 and Team 2 so that PMU Officer can access to system

Activity

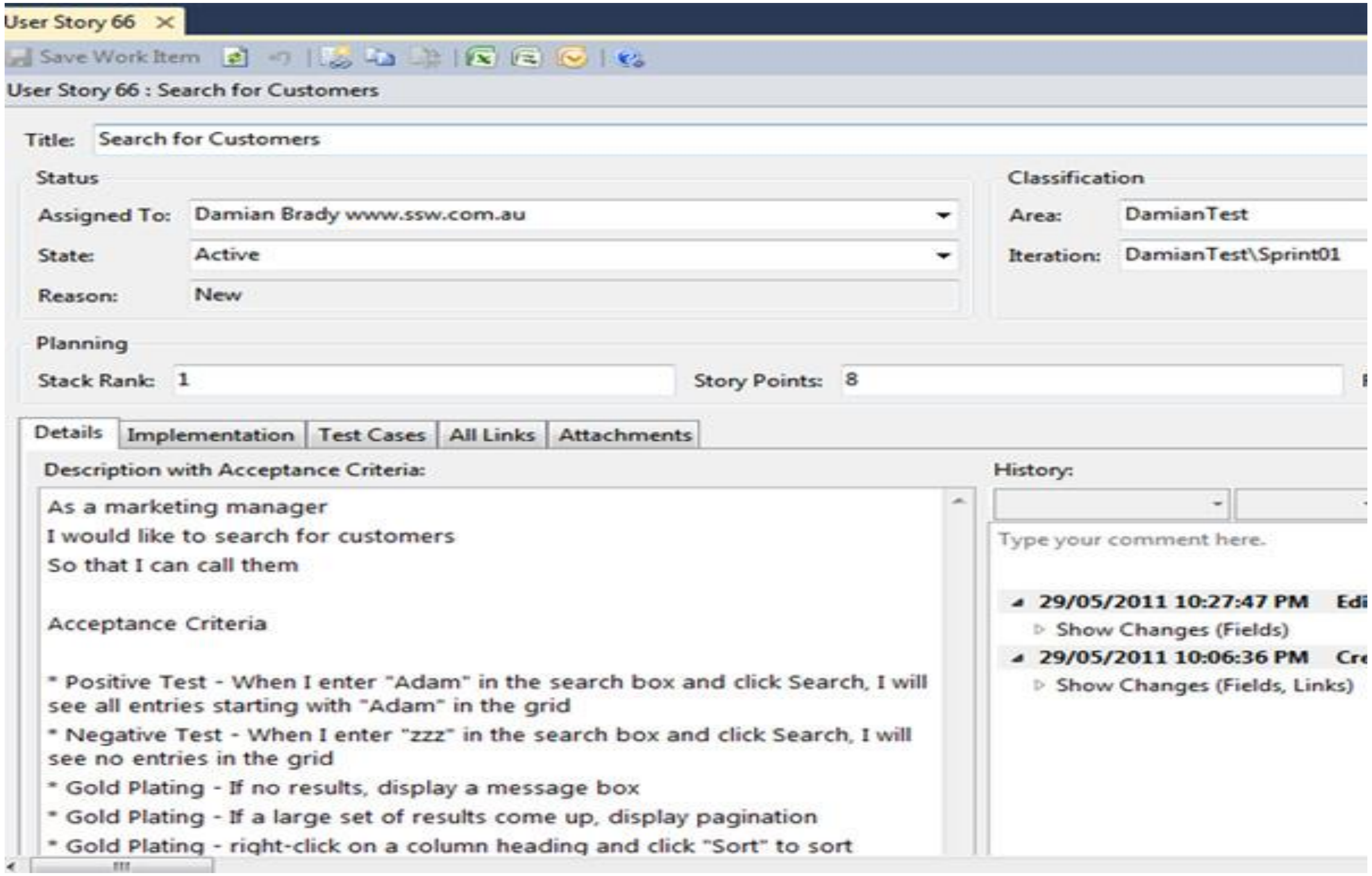
No work has yet been logged on this issue.

HipChat discussions

Do you want to discuss this issue? Connect to HipChat.

Screenshot of Jira Agile Tool with User story: Create PMU Officer

Team Foundation Server (TFS)



User Story 66 : Search for Customers

Title: Search for Customers

Status

Assigned To: Damian Brady www.ssw.com.au

State: Active

Reason: New

Classification

Area: DamianTest

Iteration: DamianTest\Sprint01

Planning

Stack Rank: 1

Story Points: 8

Description with Acceptance Criteria:

As a marketing manager
I would like to search for customers
So that I can call them

Acceptance Criteria

* Positive Test - When I enter "Adam" in the search box and click Search, I will see all entries starting with "Adam" in the grid
* Negative Test - When I enter "zzz" in the search box and click Search, I will see no entries in the grid
* Gold Plating - If no results, display a message box
* Gold Plating - If a large set of results come up, display pagination
* Gold Plating - right-click on a column heading and click "Sort" to sort

History:

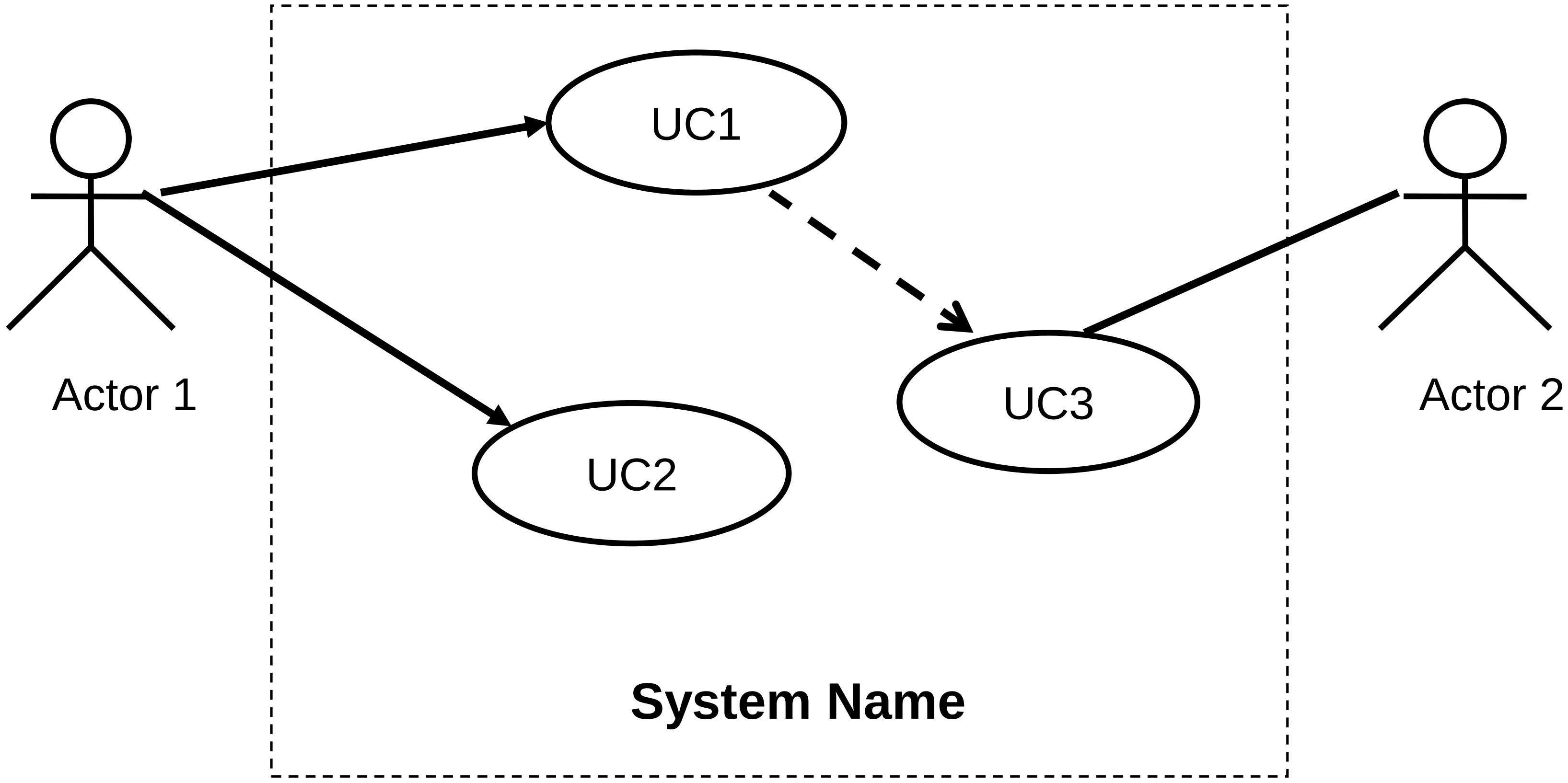
Type your comment here.

29/05/2011 10:27:47 PM Edi
Show Changes (Fields)
29/05/2011 10:06:36 PM Cre
Show Changes (Fields, Links)

Screenshot of TFS Tool with User story: Search for Customers

Other Requirement Artifacts

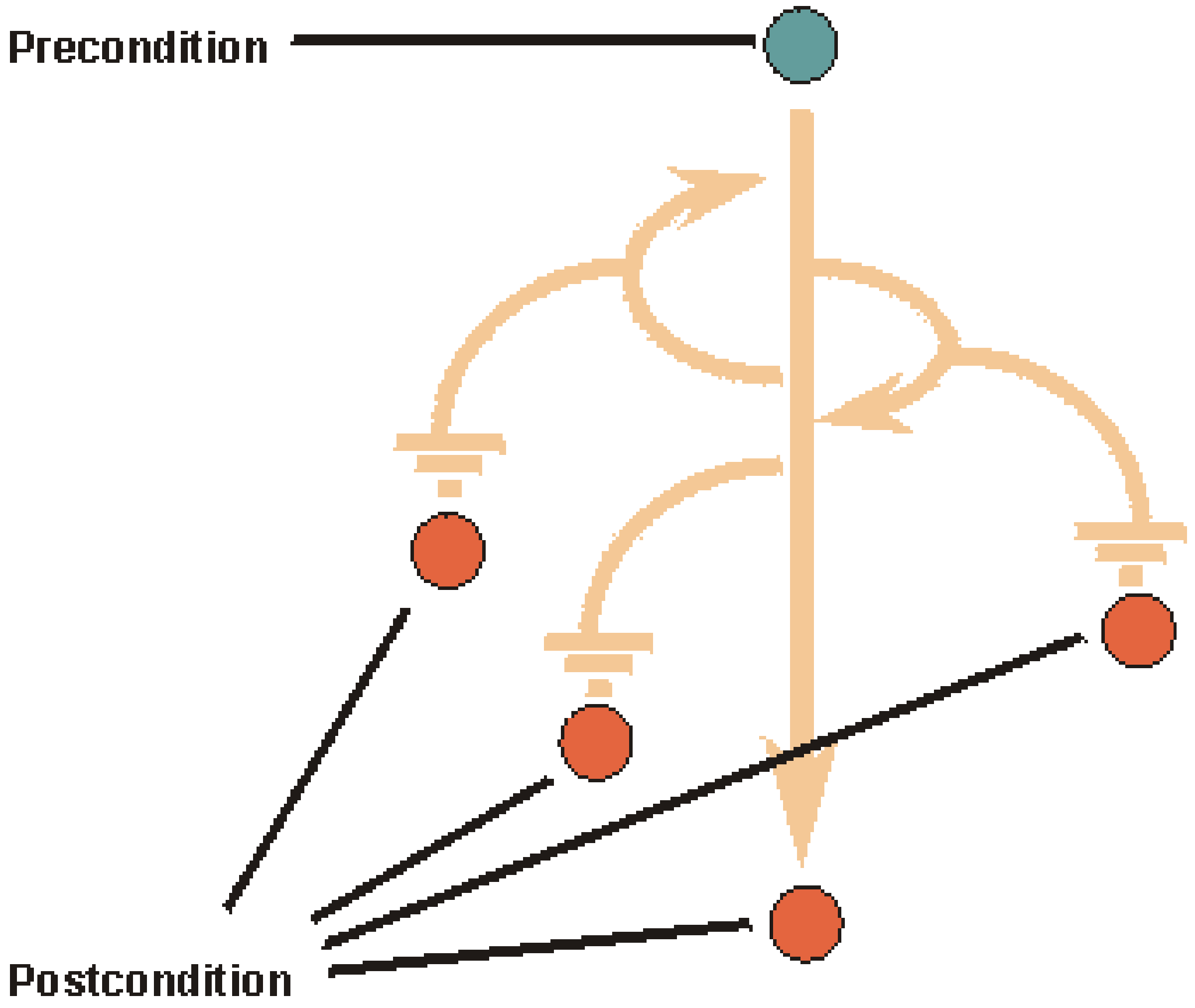
Use Case Model (cont.)



Use Case Model Diagram

Use Case

- Main Sections:
  - Pre-Conditions
  - Post-Conditions
  - Flow of Events
    - Basic Flow (only one)
    - Alternative Flows (one or many)
    - Exception Flows (one or many)
- Business Rules
- Special Requirements
- Supplementary Information



Functional Specifications – Sample

- Functional Specifications usually contains following information:
- GUI screen
- Description about behaviors of screen elements

Requirement Specifications – What is it?

- Requirement Specifications so- called Software Requirement Specifications (SRS)
- SRS captures complete software requirements for the system, or a portion of the system.
- SRS fully describes the external behavior of the application or subsystem identified.
- SRS also contains nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software

Points to Remember

Summary

- What is Functional Requirements? Non-Functional Requirements?
- What is User Story?
- What does INVEST stand for? Meaning of each?
- When to split a User Story?
- What does it mean by splitting a User Story across Data and Operational boundaries?
- What are cross-cutting concerns?
- List non-functional requirements?

Q&A

Thank You

Revision History

Date	Version	Description	Updated by	Reviewed and Approved By
30-Sep-2015	1.0	Initial	Anh Truong Thy Vo Anh To	Khanh Lam Quang Tran
09-May-2019	2.0	Update: - Update Example - Update Agile Model Picture - Remove “Write User Story Effectively” section - Remove unnecessary page under Use Case section	Dao Nguyen Anh To Thy Vo	Quang Tran