Requirements Engineering Software Engineering CS 130 Donald J. Patterson

Content adapted from Essentials of Software Engineering 3rd edition by Tsui, Karam, Bernal Jones and Bartlett Learning



Ideally Requirements Engineering is about the "What"

Preparation for Requirements Engineering



1. Prior to actually performing the requirements engineering activities, it is important to <u>plan for the resources, methodology and time needed</u> to perform this crucial step in software engineering.

2. Some organizations even perform requirements engineering <u>as a separate</u>, <u>stand-alone activity and price it separately</u>, with the option of folding the cost Into the whole project if they get the call to complete the software project.





Major Requirements Engineering Activities

- Elicitation
- Documentation and definitions
- Prototyping
- Analysis
- Specification(s)
- Review and Validations
- Gain Agreement and Acceptance

<u>Note</u>: This list does not address <u>"inventing"</u> new requirements



- Red color represents direct user/customer involvement

Why is this set of activities important and why should requirements be documented? (remember Chaos Report?)

- <u>Clear requirements are needed for design and implementation</u> activities.
- Requirements <u>documentation</u> is needed to <u>create test cases</u> and test scenarios - - - especially for large systems where the test team is a separate group of people from the developers.
- Requirements <u>document</u> is needed to <u>control potential scope-</u> <u>creep</u>.
- Requirements <u>document</u> is needed to create <u>user training</u> material, marketing material, and documents for support and maintenance.
- Requirements <u>document</u> provides a way to <u>segment a large</u> project, prioritize releases , and easier project management

Think about agile processes where this crucial step may sometimes be "compromised" by the novice software engineers.

Requirements Elicitation

- Requirements:
 - May be given to the software engineers
 - Initial product/system requirements
 - For second and/or third follow-on release of a "planned" sequences of software product where a preliminary set of requirements are already established
 - Requirements provided as a part of a request for price quotation for a software development project
 - Have to be <u>established</u> by software engineers
 - Users sometimes have an understanding of only the requirements related to their specific job tasks
 - The business rationale and goals are not always clear to individual user and needs to be established for prioritization reason
 - There may be contradicting and incomplete requirements stated by the users and customers

High Level Requirements Elicitation

- Need to seek out the business and management perceptions and goals for the software project
 - Business opportunity and business needs
 - Justification for the project
 - Scope
 - Major constraints
 - Major functionality
 - Success Factor
 - User characteristics

Software Engineers who have to interact with business management and handle requirements are sometimes called **Business Analysts**

6-Dimensions of Detailed Requirements Elicitation



Requirements Analysis

- Requirements "<u>analysis</u>" is composed of:
 - <u>Categorizing</u> the requirements (by some criteria)
 - Prioritizing the requirements
- Also "start to look" for <u>consistency &</u> <u>completeness</u> (see VORD Slide)

Requirements Classification/Categorization

- <u>Most High Level</u>:
 - Functional
 - Non-functional
- Other more detailed grouping also exist
 - 6 dimensions of requirements

Requirements <u>Categorization</u>

- By detailed 6 requirements areas:
 - 1. Individual functionality
 - 2. Business flow (usage 'scenarios')
 - 3. Data and information needs
 - 4. User interfaces
 - 5. Other interfaces to external systems/platforms
 - 6. Various constraints (non-functional)
 - Performance
 - Security
 - Reliability
 - etc.

Requirements "Analysis/Categorization" of <u>Multiple Views</u>

- <u>View Oriented Requirements Definition (VORD)</u> is based on the concept that requirements are viewed differently by different people:
 - Identify stakeholders and their viewpoints of the requirements
 - <u>Categorize the viewpoints of requirements</u> and eliminating any duplication (look for consistency & completeness)
 - Refine the identified viewpoints of requirements
 - Map the viewpoints of requirements to the system and the services that the system must provide

Requirements Prioritization

- Most of the time we have some limitations in developing software:
 - Time
 - Resources
 - Technical capabilities (existing)
- We need to prioritize the requirements to satisfy these limitations

Requirements <u>Priorities</u>

- Some Criteria for prioritization:
 - Current user/customer <u>demands</u> or <u>needs</u>
 - Competition and current <u>market condition</u>
 - Anticipated future and new customer needs
 - Sales <u>advantages</u>
 - Existing critical problems in current product
- These are often subjective and requirements should be prioritized with the help of many of the stakeholders (different viewpoints).

A Simple Requirements Prioritization List "sample"

Req. #	Brief Req. description	Req. source	Req. priority	Req. Status
# 1	One page query must respond in less than 1 second	A Major account Marketing Rep.	Priority 1*	Accepted for this release
# 2	Help text must be field sensitive	Large account users	Priority 2	Postponed For next release

* Priority may be 1, 2, 3, or 4, with 1 being the highest

Requirements Comparison and Prioritization

- Requirements prioritization is an activity of comparing the requirements and placing them in some order relative to each other.
 - Is it always performed with just <u>experience</u> and <u>gut feel ?</u> and / or
 - Done with a <u>little more rigor</u>:
 - Sort by priority groups (e.g. previous 2 charts) where the priority groups are based on some prioritization criteria list (e.g. current user needs has the highest priority)
 - <u>Pair-wise comparison</u>, normalize and compute relative value using the Analytical Hierarchical Process (AHP) – see pages

Analytical Hierarchical Process (AHP) <u>example</u>

	Req 1	Req 2	Req 3	
Req 1	1	2	3	
Req 2	1/2	1	2	
Req 3	1/3	1/2	1	Requirements Prioritiza
♦	1.83	3.5	6.0	Req 1 = $1.62/3 = 54\%$
	Req 1	Req 2	Req 3	Req $2 = .89 / 3 = 30\%$ Req $3 = .49 / 3 = 16\%$
Req 1	.55	.57	.5	1.62
Req 2	.27	.29	.33	.89
Req 3	.18	.14	.17	.49

Requirements <u>Definition/Prototyping/Review</u>

- Once the requirements are solicited, analyzed and prioritized, more details may/ must be spelled out. Three major activities which may be intertwined must be performed:
 - Requirements definition
 - Requirements prototyping
 - Requirements reviewing

Requirements Definitions/Documentation

- Requirements definitions may be written in different forms:
 - 1. Simple Input/Process/Output (I-P-U) descriptions in English
 - 2. Dataflow diagrams (DFD)
 - 3. Entity Relations diagram (ERD)
 - 4. Use Case Diagram from Unified Modeling Language (UML)
- Once the requirements are defined in detail using any of the above forms, they still need to be <u>reviewed</u> (see chapter 10 of your textbook) by the users/customers and other stakeholders.

Formal Review by Others

Requirement Definition using English and Input-Process-Output Diagram Form

Req. #	Input	Process	Output
# 12: customer Order Entry	 <i>Items</i> by type and quantity <i>Submit</i> request of items 	- Accept the items and respective quantities/ include error and rejection of items	- <i>Display</i> "acceptance" message and - <i>Ask</i> for confirmation message

English I/P/O : functionality, business and data flow

Syntax of Data Flow Diagram (DFD)



Requirements Definition using DFD

Captures – functionality, business flow, data



Requirements Definition using Entity- Relation-Diagram (ERD)



Cardinality: specifies the <u>number of occurrences</u> of entities



Modality: specifies the <u>necessities of relationship</u> to exist

Captures - relations among data

Requirements Definition specifying <u>Entity and Attributes</u>

(a) Graphical form

(a) Tabular form



Captures - relations among data

Requirements <u>Analysis & Definition Methodology</u> <u>using UML</u>

- Using <u>OO's Use Case</u> Methodology and Notation, which identifies:
 - Basic/Main functionalities
 - Pre-conditions of functionality
 - Flow of events or scenarios
 - Post-conditions for the functionality
 - Error conditions and alternative flows
- Using OO Use Case Diagram which identifies:
 - <u>Actors</u> (or all external interfaces with the system, including the users)
 - Related "<u>use cases</u>" (major functionalities)
 - Boundary conditions

Use Cases may be further refined during Design

Requirements Definition using <u>Use Case Diagram</u> (for --- major functionalities)



Requirements Definition using <u>Use Case Description</u>

- Use Case Diagrams are <u>mostly for functionalities and</u> are not detailed enough ---- so we <u>need to use</u> <u>"English" text</u> for further descriptions of the requirements:
 - Functional details
 - Pre conditions
 - Post conditions
 - Non-functional characteristics about the functionalities
 - "alternative paths" (e.g. error processing)
 - UI sample (e.g. "looks")
 - etc.

Requirements <u>Traceability</u>

- Capability to <u>trace a requirements</u> is needed to ensure that the product has <u>fully implemented</u> <u>the requirements</u>. (while not part of requirements process activities – it needs to be started early)
- We need to trace requirements:
 - Requirements <u>from</u> source (people and documents)
 - Requirements <u>to</u> later steps (design, implementation, test, & release)
- We also need to *link* requirements to other <u>"pre-requisite</u>" requirements.

Partially filled Traceability Table

Requirement	Design	Code	Test	Other related requirements
1	compon. X	Module 3	Test Case 32	2,3
2	compon. w	Module 5	Test Case 16	1
3	compon. X	Module 2	Test Case 27	1
4				
5				
6				
7				

Requirements Prototyping

- Requirements prototyping mostly address the <u>User Interface (UI)</u> part of the requirement in terms of:
 - Visual looks (size, shape, position, color)
 - Flow (control and logical flow; depth of flow)
- The prototyping may be performed in one of the two modes:
 - Low fidelity : using paper/cardboard to represent screens and human to move the boards
 - High fidelity : using automated tools such as Visual Basic to code the screens and direct the logical flow of these screens

Requirements Specification

- Once the requirements are <u>defined</u>, <u>prototyped</u> and <u>reviewed</u>, it must be placed into a <u>Requirements</u> <u>Specification document</u>
- IEEE /EIA Standard 12207.1-1997 may be purchased from IEEE. It outlines the guideline for 3 major sections of a requirements specification document.
 - Introduction
 - High Level description
 - Detailed descriptions
 - Details of each functionality (input-out-process)
 - Interfaces, including user interfaces and network interfaces
 - Performance requirements (response time, throughput, etc.)
 - Design constraints, standards, etc.
 - Additional attributes such as security, reliability, etc.
 - Any additional unique requirements

Finally --- Requirements "Sign Off"

- Having a requirements specification agreed to and signed off is important:
 - Serves as a milestone marker and <u>formally exits</u> a phase of software of engineering
 - Serves as <u>baseline</u> from which any future changes can be monitored and controlled

Remember "Agile" ? -- which wants less "legalese" like sign-off

WESTMONT COMPUTER SCIENCE

