Manage Requirements

Creating High-Quality Deliverables

- Quality means different things to different people
- Quality is Conformance to Requirements AND Fitness for Use
- Understand what is important for the users (objectives)
- Understand Requirements - functional and non-functional, explicit and implicit

Defect Injection data – (normalized)

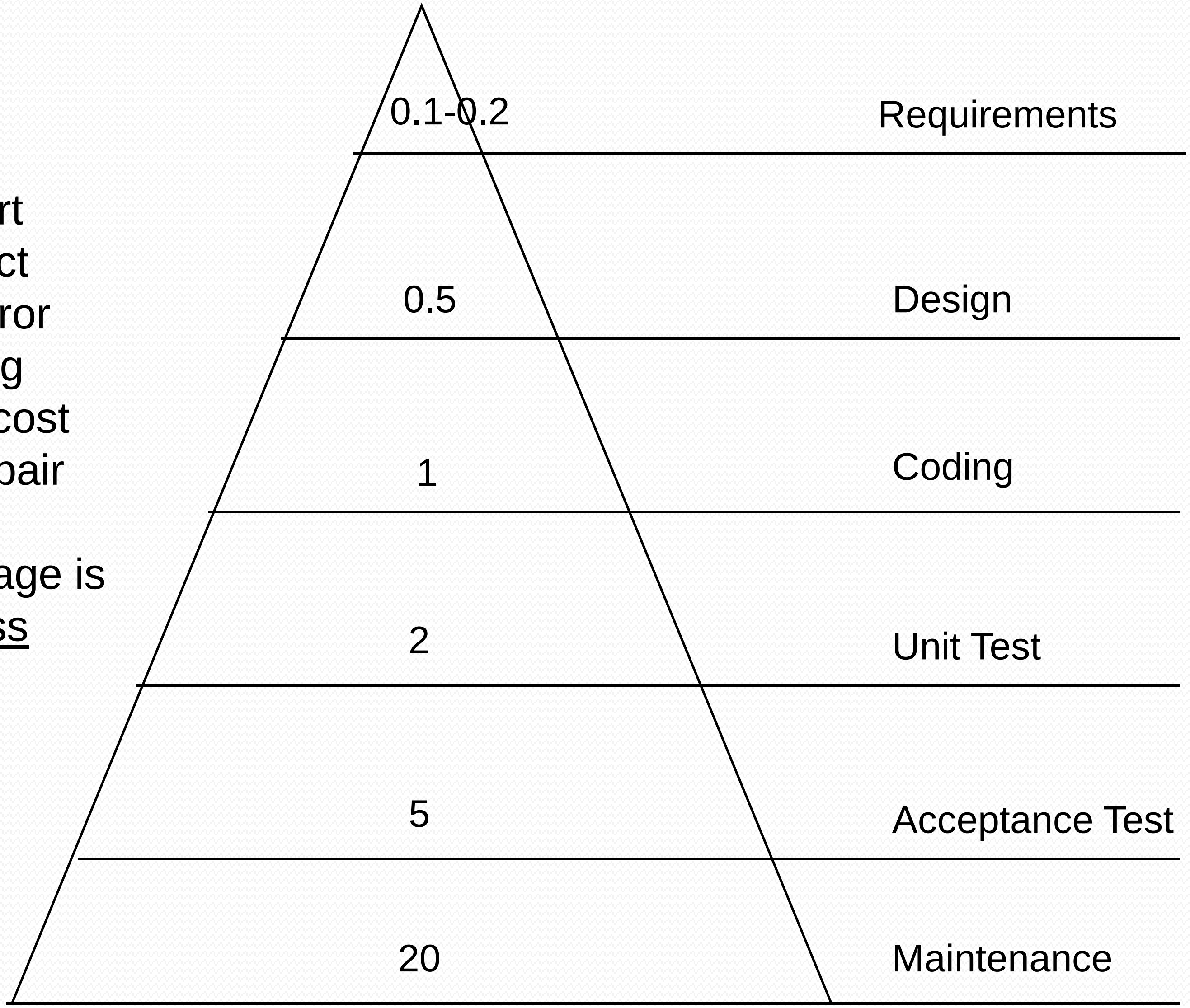
Defect Origins	Defect Potentials	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Documentation	0.60	80%	0.12
Bad Fixes	0.40	70%	0.12
TOTAL	5.00	85%	0.75

IT executive managers on their opinions about why projects are challenged

Project Challenged Factors	**% of Responses**
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%

Relative Cost to Repair a Defect at Different Life Cycle Phases

Unit cost of 1 is assigned to effort required to detect and repair an error during the coding stage, then the cost to detect and repair during the requirements stage is 5 to 10 times less



0.1-0.2	Requirements
0.5	Design
1	Coding
2	Unit Test
5	Acceptance Test
20	Maintenance

Conclusion

- If one third of the defects are Requirement Defects AND
- Requirement Defects are more costly to fix later THEN
- It makes a lot of sense to Manage Requirements better

- Lets look at why is managing requirements difficult!

Requirements Management Issues

- The User-Developer Syndrome – the Language Barrier
- Users talk the language of business
- Developers talk the language of technology
- How can we cross the language barrier?
- **Develop domain expertise!**

Requirements Management Issues

- The “Yes, But” Syndrome
- Users know what they want, but are often unable to articulate it due to the language barrier
- When you show them something, they say “Yes, But…”

- **Use a process that seeks user feedback early**

Manage Requirements: Match steps to descriptions PRIORITIZE; CONTROL; VERIFY; DISCOVER; VALIDATE

1: What exactly is required to be done?

2: Are ALL Requirements equally important? You wont finish them all anyways!!

3: Are we doing it right? (Traceability)

4: Is this really what the customer wanted? The "Yes, But" syndrome

5: What is the IMPACT of this change?

Manage Requirements: Steps and descriptions

- DISCOVER : What exactly is required to be done?
- PRIORITIZE : Are ALL Requirements equally important?

  You wont finish them all anyways!!
- VERIFY : Are we doing it right? (Traceability)
- VALIDATE : Is this really what the customer wanted?

  The "Yes, But" syndrome
- CONTROL : What is the IMPACT of this change?

Manage Requirements

- DISCOVERY: What exactly is required to be done?
  Exercise: Flying device!

- Exercise 1: Take a rough piece of paper.
- You have to create a flying device using the paper. You have 30 seconds and your time starts NOW!

Manage Requirements

- Now draw a circle on the white board. Ask participants to hit the circle with their flying devices, without leaving their seat!
- Question: If you had known that the OBJECTIVE was to hit the circle drawn on the board, would you have chosen a SPECIFIC DESIGN for the flying device?

Manage Requirements - Verification

- Exercise 2:
- Pass the specified message by word of mouth from A to B, then B to C, then C to D and finally from D to E!
- Check the message coming out of E!

Manage Requirements - Verification

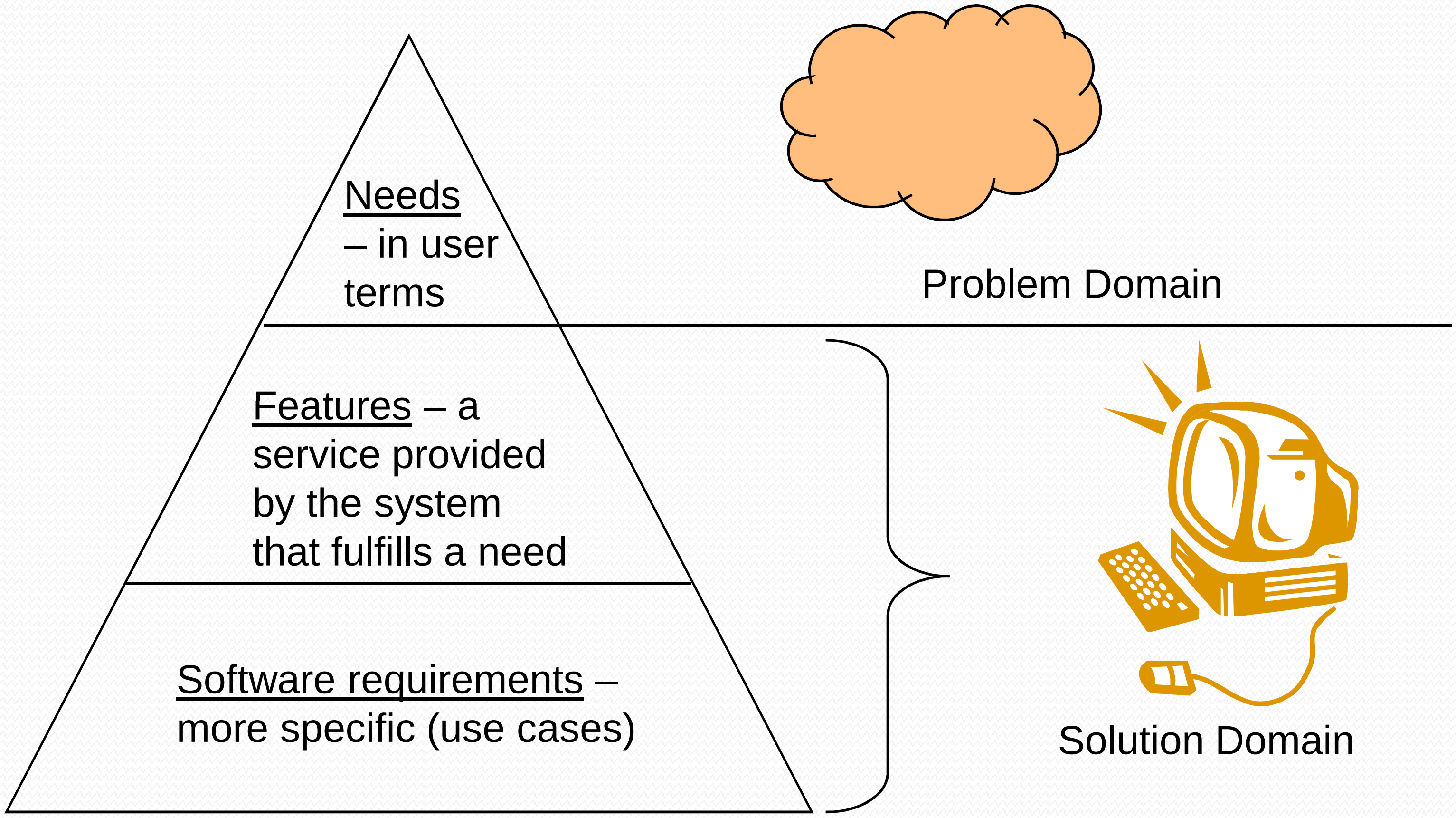
- Visitors from Guardian Insurance are coming on Monday morning for a presentation on Quality. Please ask Sunil to be ready with the presentation. Please book the conference room in SDF V and arrange for a good video projector and a PC. Please inform Admin to keep gate passes ready and make arrangements for tea/coffee.

- Requirements get changed similarly without our knowledge as they pass thru the stages of Analysis, Design, Code, Test and Delivery; We MUST VERIFY THE SCOPE at each stage using Phase-end Reviews!

Discovering Requirements

- Who are the Stakeholders?
- Who is the Customer?
- Who are the End-users?
- What is the Business problem or opportunity?
- What are the Objectives?
- What are the Constraints and Assumptions? ##

Discovering Requirements



<u>Needs</u> – in user terms

<u>Features</u> – a service provided by the system that fulfills a need

<u>Software requirements</u> – more specific (use cases)

Problem Domain

Solution Domain

Elicitation techniques; Domain Expertise; Early User Feedback

Discovering Requirements

- PROGRESSIVE ELABORATION
- NEEDS
  - Multi-user Multi-level Security
- FEATURES
  - Login Authentication; Transaction Password;
- SOFTWARE REQUIREMENTS
  - Use Case for Login Authentication; Password Encryption; Digital Signature, Digital certificate, Number of attempts ##

Discovering Requirements

- Requirements versus Specifications
- Requirements attributes: Complete, Clear and Concise
- Completeness: Explicit versus Implicit
- Clarity: Simplicity; No Ambiguity or Inconsistency
- Concise: Verbose versus Pictorial representations
- Issues: Customers do not know what they want?!
- Real Issues: Inadequate DOMAIN Knowledge!! Poor RM!! ##

Manage Requirements – PMI Tools
Match tools with descriptions

- WBS; Project Charter; Scope Statement or SOW

- 1. Describes Stakeholder needs, High-level product description, Project Justification
- 2. Describes Project Objectives, Product Requirements, Deliverables, Acceptance criteria, Constraints
- 3. Starts with Phase level Deliverables; decomposes to the work package level;

Manage Requirements – PMI Tools

- Project Charter
  - Stakeholder needs, High-level product description, Project Justification
- Scope Statement or SOW
  - Project Objectives, Product Requirements, Deliverables, Acceptance criteria, Constraints
- WBS
  - Starts with Phase level Deliverables; decomposes to the work package level;

CASE: Dealing with ***one-liner requirements***

- How would you handle the one-liner requests below?
  - 1. We need to make the application Y2K compliant!
  - 2. We have grown rapidly in the last couple of years. We need to add one more digit to the customer code! We must have it within 2 weeks, before the holiday season starts!!
  - 3. Can you quickly gives us a facility to view history data?

CASE: Dealing with one-liner requirements

- Facility to view history data:
  - First we need to figure out which data needs to be viewed – which tables, which fields? Is it local or distributed?
  - Currently does the application save the History data – where and how?
  - If not we need to worry about saving it in the first place!!
  - View means what? View on Screen? Also print? Or print to file?
  - Do we need to provide the facility to sort or filter the data?
  - How about security? Which users need to view which History data?
  - Is this facility required at one place in the application or it could be invoked from different points?

To summarize "Manage Requirements"!

- Discover, Prioritize, Verify, Validate & Control Requirements
- Understand the Stakeholders, Customers, End-Users, the Business problem or opportunity, the Objectives, the Constraints and Assumptions!!
- Use Progressive Elaboration to ensure requirements are Complete, Clear and Concise
- Use relevant tools like Project Charter, SOW and WBS
- Be extra careful with One-Liner Requirements and Change Requests if your project is Fixed Price!

- Introspect!!
- Any new ideas?
- Any Action Items?