

Requirements Engineering

What?

- Applying engineering processes to understanding what a software system should do
- Requirements ARE what the client wants the software to do

Why?

- Don't waste resources
- You have to know “what” to build

Tasks for Requirements Engineering

- Inception
- Elicitation
- Elaboration
- Negotiation
- Specification
- Validation
- Management

Inception

- Target market (identify stakeholders)
- Recognize multiple points of view
- Work to collaborate with the client
 - Identify problem cases early
- Ask context free questions

Questions

- What do you want?
- Who wants this product?
- Who will use the product?
- What are you expecting from a “successful” product?
- What economic benefit will come from the project?
- What other options do you have?

More Questions

- Are you the right person to be asking?
- Are your answers official?
- Are my questions relevant to the project?
- Am I asking too many questions?
- Who else should I be talking with?
- What questions should I ask that I haven't?

Elicitation

- Collaborative Requirements Gathering
 - Team approach
 - Set up meetings
 - Establish rules
 - Semi-formal agenda (flexible)
 - Facilitator before the meeting
 - Get requirements

Quality Function Deployment

- Translate customer needs into technical requirements
- Normal: those items specified in customer meetings, if they are met, customer is satisfied
- Expected: “obvious” things, not explicit, if missing, customer unhappy
- Exciting: “extras”, if included, customer is very satisfied

Use-Cases (scenarios)

- Many forms
 - Stories
 - Task lists
 - Outlines
 - Drawings, sketches
- Represent the USER'S point of view

Caveats

- Must identify ACTORS
 - Actors can be people or devices
- Identify actor's goals with system
- What is the main task?
- Any exceptions illustrated in this case?
- Are there variations in how to accomplish the goal?
- What information is shared?

Analysis Modeling

- Just a snapshot
- Provides models of the system
- Focus on one method or use a mixture of methods
 - Use cases
 - Diagrams
 - UML
 - Flow charts
 - Sketches
 - Drawings
 - Storyboards

Common Elements

- Scenario-based elements
 - Stakeholders, behavior, information flow
 - (early, will serve as input to other elements)
- Class-based elements
 - (objects that the actors use/interact with/manipulate, etc)
- Behavioral elements
 - How the system moves from state to state
INTERNALLY
 - State transition diagrams

Common Elements

- Flow-oriented elements
 - Information will be transformed as it moves through the system

Negotiation

- Ideally, we would like to NOT have any negotiation
- Reality says you WILL go through negotiation
- Balance functionality, performance with cost and time to deliver (time to market)
- Strive to meet customer's needs, but must reflect real-world

Art of Negotiation

- Not a competition
- Map out a strategy (anticipate problem areas)
- Listen actively
- Focus on other parties' interests
- Don't get personal
- Be creative (be open minded)
- Be ready to commit

Validation

- Are requirements consistent with the project?
- Are the requirements abstract at this stage?
- Is this requirement really necessary? (is this an add-on)
- Are the requirements bounded?
- Is there a source for each requirement?
- Is this requirement achievable in the environment?
- Is this requirement testable?
- Do the requirements properly reflect the information, function, and behavior of the system?
- Did we partition the analysis model to expose progressive detail?