

Intro. to Requirements

Dave Levitt

CS 2000: Systems Analysis & Design

Agenda

- Team selection due by end of class!
- Questions on readings, RUP, SDLC, etc.
- Quick review of last week's discussion
- Discuss articles
- Introduction to requirements.

Requirements Overview

- Common Terms:
 - Stakeholder
 - Sponsor
 - Requirement
 - Functional Requirement
 - Non-functional (quality) requirement
 - Constraint
 - Hacking

Requirements Overview (cont'd)

- A Functional Requirement is a capability that is needed by a user to solve the problem.
 - If you don't know what you want, you're not likely to get it.
 - Knowing what you want is not as easy as it sounds.
 - We will capture these in use cases.

Requirements Overview (cont'd)

- A Non-Functional Requirement describe quality attributes that the system must possess. Example: Performance, reliability.
 - We'll capture these in a Supplemental Spec.
- A Constraint is a restriction on a solution set.
 - Are not implemented. Example: Time, cost, budget, development tools. Capture in Supp. Spec. or other documents.
- Class discussion: You are a homeowner talking to a builder about remodeling your kitchen:
 - What might some of the functional requirements be?
 - What might some of the non-functional requirements be?
 - What might some of the constraints be?

Requirements Overview (cont'd)

- Understanding your customer's requirements is important because:
 - They drive downstream development activities.
 - Fixing requirements errors later in the process is more costly to fix
 - although these studies are being challenged by XP, etc.
 - If it doesn't meet requirements, it doesn't matter if it works.
 - If you don't meet your customer's requirements, you're not likely to get paid, or get called back for more work.

Requirements Overview (cont'd)

- Weinberg Study (1974):
 - Contrary to popular belief, programmers do what they're instructed to do.
 - The root cause of many problems is programmers are often not told (clearly) what they should do.

Requirements Overview (cont'd)

- Standish Group Study (1994):
 - US spends over \$250 billion annually on application development.
 - The average cost for a project in a large company is over 2M: 1M for a medium sized company, .5M for a small company.
 - Over 31% of these projects will fail. Over 50% will incur cost overruns of over 180%.
 - Root Cause:
 - Lack of user input: 13%
 - Incomplete Requirements: 12%
 - Changing Requirements: 12%

Non-Functional Requirements

- Observable at runtime:
 - Performance
 - Security
 - Availability
 - Usability (learnability, efficiency, memorability, error avoidance, error handling, satisfaction).
- Not observable at runtime:
 - Modifiability
 - Portability
 - Reusability
 - Integrability
 - Testability
 - Legal, Packaging, etc.

Attributes of good requirements

- The prime directive: requirements must be written and organized so that stakeholders AND the development team AND the test team can understand them!!!!!!
- Requirements that are captured effectively are:
 - Complete
 - Correct
 - Feasible
 - Necessary
 - Prioritized
 - Unambiguous
 - Verifiable (Testable)

Ambiguity and Specificity

- What level of specificity is needed to avoid being misunderstood?
 - It depends on many factors: the system, the Stakeholders, the culture, product knowledge, etc.
 - The right level is called the “sweet spot”.
 - Finding the “sweet spot” is an acquired skill.
 - What works on one project, team, or client may not necessarily work somewhere else!
 - Acid tests that you’re “done”:
 - Do end users understand it?
 - Can testers write test cases?
 - Do Analysts and designer have enough information?
 - Can you start writing a user manual?

Requirements Elicitation

- A Vision Document helps “jump start” the requirements effort.
 - Short: less than 10 pages.
 - Captures at a high level the application, target market, target users, key stakeholders, application features, and major risks / constraints.
 - Helps define the initial system boundary
 - Used to gain consensus with the project sponsor, project team, and other stakeholders.
 - Usually written by a business person with the help of an Analyst.

Requirements Elicitation (cont'd)

- Vision Document Answers Context Free Questions:
 - Process:
 - Who is the client? How much time and money do we have?
What are other alternatives?
 - Product:
 - What does this problem solve? What problems does it create?
What is the target environment? What type of precision is required?

Requirements Elicitation (cont'd)

- Before gathering detailed requirements, it's helpful to ask your stakeholders some context free questions, such as:
 - What is your experience / background?
 - Are you the right person? Are you comfortable with this?
 - Are your answers official?
 - What are your communication preferences?
 - What days/times are you most available? How much time can you devote to this?
 - Is there a backup person?
 - Is there anybody else I should consider? Not consider? Why?
 - What questions to you have for me?
 - Do my questions seems relevant?
 - What other questions should I be asking?

Requirements Elicitation (cont'd)

- Requirements Workshops
 - Benefits:
 - Gain consensus on requirements
 - Transfer of knowledge
 - Builds team jell
 - Roles:
 - Facilitator
 - Recorder
 - White board, system with monitor, post-it pads
 - As an Analyst, you will spend a lot of time in workshops.

Requirements Elicitation (cont'd)

- Preparing for a workshop
 - Start with an introductory memo
 - Usually written by a PM or related individual
 - Provide guidelines of how long meetings will last, what's expected, etc.
 - Training / Level Setting
 - Use case training
 - Business training
 - Team Selection
 - Business, Technical, QA, Tech. Writers, others.
 - No dead weight! Everybody needs a purpose.

Requirements Elicitation (cont'd)

- Before each meeting: Facilitator
 - Set an agenda
 - Date, Time, Location, Participants, Purpose, Preparation Requirements, Discussion Topics.
 - Insure room has proper facilitates (white board, markers, erasers, other equipment).
 - Arrange for lunch / beverages
- After each meeting: Recorder
 - Prepare minutes
 - Recap of agenda, highlights of meeting, action items, whom, when

Requirements Elicitation (cont'd)

- The Facilitator:
 - Runs the meeting in an orderly manner.
 - Builds trust of team members.
 - Manages conflict (dialog)
 - Stays neutral
 - Requires excellent negotiating and problem solving skills.
 - Usually a primary role of the Analyst, but sometimes a special facilitator is needed.
 - Probably one of the most important skills of an Analyst is to be a good facilitator.

Requirements Elicitation (cont'd)

- Facilitating with problem people:
 - Filibusters Repeaters / Broken Records
 - Latecomers Headshakers
 - Dropouts Whisperers
 - Attackers Gossipers
 - Interpreters Know-it-alls
 - Backseat drivers Busybodies
 - Teachers Pet Doubters
- Remain neutral, but in control
- Praise publicly, criticize privately

Requirements Elicitation (cont'd)

- Facilitating in the face of conflict
 - Conflict vs. dialog
 - All teams will face conflict.
 - One of the signs of a healthy team is how well it resolves conflicts.
 - The art of being fully present:
 - Notice everything you can.
 - Ask about things you don't understand.
 - Feel free to comment on anything

Requirements Elicitation (cont'd)

- Facilitating in the face of conflict (cont'd)
 - Most negotiating is positional bargaining
 - Win/lose
 - Dig in
 - Search for a single answer
 - Apply pressure
 - Threats
 - Strong willed

Requirements Elicitation (cont'd)

- Change the game – principled negotiations
 - Separate people from the problem
 - Be soft on people, hard on the problem
 - Seek to understand, then be understood
 - Focus on interests
 - Explore interests / avoid having a bottom line
 - Invent options for mutual gain
 - Use problem solving skills to develop mutual options – decide later
 - Insist on using objective criteria
 - Yield to principle, not pressure

Requirements Elicitation (cont'd)

- Facilitating – interviewing
 - Watch for verbal and non-verbal signals.
 - Be prepared, but realize that you won't know the next question until the answer to the previous one is provided.
 - Learn the language of the Stakeholders.
 - Frame questions so that the reply can be validated.
 - Be prepared. See if the information exists elsewhere before taking up someone's time

Requirements Elicitation (cont'd)

- Interviewing (cont'd)
 - Leading Questions:
 - You're not going to do 'x' are you?
 - I'm concerned about 'x' because....
 - Loaded Questions:
 - Are we going to keep this clumsy file?
 - I have a hard time understanding this and I'm afraid others may too.
 - Self Answering Questions:
 - How many users? 50 right?
 - I would like to set a limit on the number of users so we can build the right sized system

Requirements Elicitation (cont'd)

- Interviewing (cont'd)
 - Parting shots:
 - Ask just when everybody is about to leave.
 - I just thought of something. I'll call you later.
 - Controlling Questions:
 - Jot down the item and move onto the next.
- *You want to be in control, but not in absolute control. That is impossible.*

Requirements Elicitation (cont'd)

- The user is not always right, but they do always have a point!
- You're not taking orders for hamburgers.
- Be patient.
- *Your goal is not perfection – just good enough.*