

## Discussion Slides

9/8/2007

Requirements

27

## Deeper ●

- Why are requirements errors so expensive to fix later?

*Because they are the basis for the architecture, design, specifications, and code. Changing the requirements could necessitate redoing all of that other work.*

- Why are they more important than other types of errors?

*Because they are almost sure to translate into customer satisfaction issues.*

9/8/2007

Requirements

28

## Deeper ●

- Is it possible to meaningfully explore customer requirements if the general business requirements are not yet clear?

*It is hard to develop user requirements if you don't know who the users are or what problem you are solving.*

*It is certainly possible, however, to explore user requirements before we settle on a price point, or understand who our partners might be.*

9/8/2007

Requirements

29

## Deeper ●

- Should all lower level requirements be traceable to higher level requirements?

*Lower level requirements are mostly expansions of higher level requirements. We should not "make up" new requirements as we elaborate our plans and designs.*

*But – many requirements do not come from the product concept (e.g. legal and standards compliance), and so may not have been mentioned in user level requirements gathering activities.*

9/8/2007

Requirements

30

## Deeper ●

- Why might an "interface specification" be considered a requirement, rather than a specification or design detail?

*Internal interfaces usually are considered to be design details.*

*External interfaces affect interoperability with other products and services, and interoperability constraints are clearly requirements.*

9/8/2007

Requirements

31

## Deeper ●

- Give examples of ambiguous or unbounded requirements.

*Response time to trivial request must be good.*

*System must be scalable to large loads.*

*System must be robust to user errors.*

*All code to be thoroughly tested.*

- How can we fix them?

*Often, by turning them into (non-arbitrary) quantitative statements.*

9/8/2007

Requirements

32

## Deeper ☺

- Why must requirements be traceable to an underlying problem?

*Requirements are often misunderstood. Being able to trace it back to the problem it is supposed to solve will help to ensure that we do solve the problem.*

Why traceable to a particular person?

*If we need to verify or seek clarification on a requirement, we need to know who to ask.*

9/8/2007

Requirements

33

## Deeper ☺

- Why must requirements be prioritized?

*When we are trying to reconcile conflicting goals, or impossible schedules, we need to be able to balance them against one another or to order our requirements so that we can still satisfy the most important ones.*

9/8/2007

Requirements

34

## Deeper ☺

- Why do we say that requirements should say what, but not how?

*This might over-constrain the problem, making it more difficult to reconcile competing requirements.*

*Product design is driven by many factors other than functional requirements.*

*Most users do not understand the implementation technology and framework and so are not competent to design a viable implementation.*

9/8/2007

Requirements

35

## Deeper ☺

- Why might it make sense to specify design or implementation details in a set of design requirements?

*The use (or reuse) of particular components or technology may be part of a strategic initiative or significantly lower engineering costs.*

*Organizational realities (like multi-site or out-sourced) development) could necessitate independence of selected components.*

*Such considerations often impose requirements on system and component designs.*

9/8/2007

Requirements

36

## Deeper ☺

- Why should all requirements be testable?

*If it is a requirement, we should test it as part of our product validation. If it is not testable, how will we know if we have satisfied it?*

*In most cases, requirements are non-testable because they are vague. It is often difficult to clarify these issues, but the reward is a much clearer understanding of our goal, and much greater chance of achieving it.*

9/8/2007

Requirements

37

## Deeper ☺

- Where could unachievable requirements come from?

*Somebody could easily want something, that turned out to be beyond our technological capabilities.*

*Some managers confuse requirements with stretch goals. The former must be achieved. The latter, we would like to achieve (or at least get closer to).*

9/8/2007

Requirements

38

## Deeper ☺

- Why is it important that the implications of each requirements change be studied and understood before it is accepted?

*An ill considered requirements change could greatly affect the amount of work involved, or even the feasibility of a project.*

*If the change is made late in the project, it could invalidate large amounts of work.*

*This doesn't mean no changes ... but it does mean we should weigh each carefully.*

9/8/2007

Requirements

39

## Deeper ☺

- Why is it important that we track the dependencies of requirements to specifications and other requirements?

*This cross referencing will help us to more quickly and completely understand the implications of proposed requirements changes.*

*When a requirement changes, we know what specifications must change (and which groups must be consulted)*

9/8/2007

Requirements

40

## Deeper ☺

- Why is it important to identify the full range of likely stake-holders?

*Any stake holder may be a source of key requirements. If we fail to talk to a stake holder, we may miss key requirements that they could have contributed.*

*Many stake holders will be only briefly involved in a project ... but they may still offer valuable input during that time.*

9/8/2007

Requirements

41

## Deeper ☺

- Why is the difference between a domain expert and a potential user?

*Domain experts might include consultants, sales people, and managers who know the domain but wouldn't actually use the product. They are often more technically savvy.*

- Why is it critical to talk to potential users?  
*These are the people we must ultimately satisfy.*

9/8/2007

Requirements

42

## Deeper ☺

- Why do we want to describing our product in too much detail?  
*We don't want to bias their thinking.*
- Why do we want to avoid "selling" our product?

*The purpose and value of this meeting is gathering input about what these users need. Other discussions would shift attention or take time away from that goal.*

9/8/2007

Requirements

43

## Deeper ☺

- Why is it important to start an interview with general and open-ended questions about what they do and how, and what problems they face?

*1. This is the context within which our product must operate.*

*2. We are giving them room to tell us things that we didn't know to ask about, and to describe problems we were not aware of.*

9/8/2007

Requirements

44

## Deeper

- What can we do when we seem to have conflicting or contradictory requirements?

*Start by understanding them, the underlying issues may not truly be in conflict.*

*It may be possible to define a reasonable trade-off or to offer different options for different situations.*

*Assess the likelihood of the respective situations and the importance of the associated requirements, and let priorities guide your decision.*

9/8/2007

Requirements

45

## Deeper

- When there is more work than we can possibly do, what can we do?

*Start by understanding your project goals and how the tradeoff between schedule, functionality, and quality factors in.*

*If functionality has to slip, use the priority (value, cost, risk) based ladder to decide which features to include. This is likely to deliver the greatest possible value for the cost.*

9/8/2007

Requirements

46