소프트웨어공학 원리

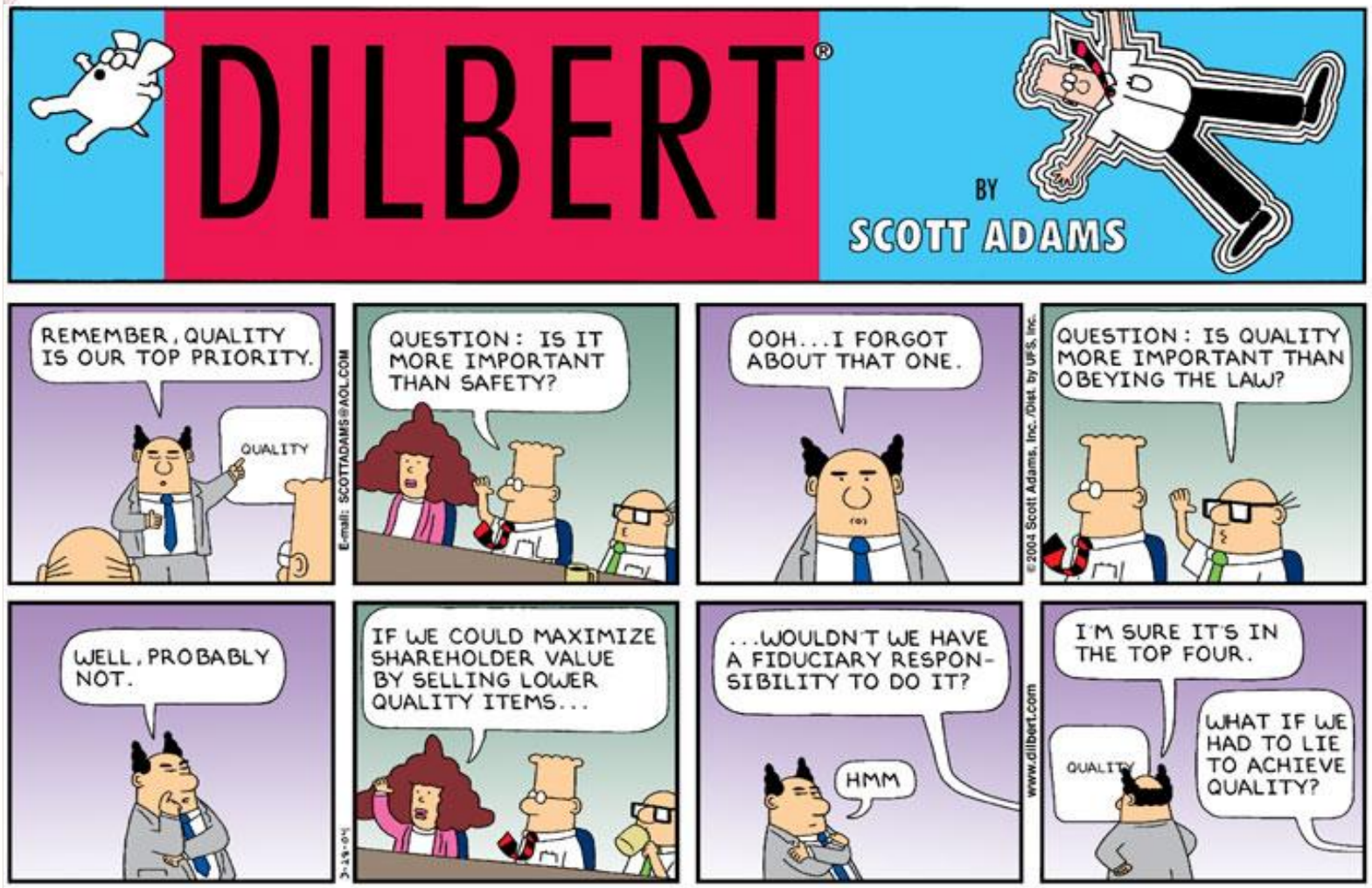
(SEP521)

Requirement Management

Jongmoon Baik

KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
KAIST
한국과학기술원
SINCE 1971

Review of Quality Attributes


DILBERT®
BY
SCOTT ADAMS
REMEMBER, QUALITY IS OUR TOP PRIORITY.
QUALITY
E-mail: SCOTTADAMS@AOL.COM
QUESTION: IS IT MORE IMPORTANT THAN SAFETY?
OOH...I FORGOT ABOUT THAT ONE.
© 2004 Scott Adams, Inc./Dist. by UFS, Inc.
QUESTION: IS QUALITY MORE IMPORTANT THAN OBEYING THE LAW?
WELL, PROBABLY NOT.
IF WE COULD MAXIMIZE SHAREHOLDER VALUE BY SELLING LOWER QUALITY ITEMS...
...WOULDN'T WE HAVE A FIDUCIARY RESPONSIBILITY TO DO IT?
HMM
www.dilbert.com
I'M SURE IT'S IN THE TOP FOUR.
QUALITY
WHAT IF WE HAD TO LIE TO ACHIEVE QUALITY?

~~Non Functional~~ Quality Attributes

- Performance
- Modifiability
- Reusability
- Reliability
- Stability
- Security
- Extendibility
- Portability
- Usability
  - User friendly?
- Scalability
- Data integrity
- & more

Recent Favorites: “Wowability” & “Buildability

Eg. Quality Attribute Scenarios

- Use case scenario

  Remote user requests a database report via the Web during peak period and receives it within 5 seconds.

- Growth scenario

  Add a new data server to reduce latency in scenario 1 to 2.5 seconds within 1 person-week.

- Exploratory scenario

  Half of the servers go down during normal operation without affecting overall system availability.

Stimulus, Environment, & Response

- Use case scenario

  Remote user requests a database report via the Web during peak period and receives it within 5 seconds.

- Growth scenario

  Add a new data server to reduce latency from 5 seconds to 1 to 2.5 seconds within 1 person-week.

- Exploratory scenario

  Half of the servers go down during normal operation without affecting overall system availability.

Quality Attribute – Utility Tree Entry Example

	Scenario(s)	Changing the radio propagation model used by the network simulator can be done without any changes to the ELzone system.
	Business Goals	Network simulator can be updated to maintain leading research/development position
	Relevant Quality Attributes	Modifiability
Scenario Components	Stimulus	Developer wants to modify network simulator behavior
	Stimulus Source	Network engineer
	Environment	Simulator development
	Artifact (if known)	
	Response	System responds as expected to network simulator change
	Response Measure	No changes to ELzone are needed
	Questions	
	Issues	
	Priority	High
	Difficulty Rating	Low

Functional Requirement Specification Table Entry Example

The requirements of the NIRC will be prioritized and handled in order of priority. These priority rankings were made with input from the client. For t he details and up-to-date information, see the requirement tracking sheets provided separately. The following table shows a snapshot of the priority at the date of this document. Both priority and difficulty are measured a 1- H (high), 2-M (medium), and 3-L (low).

REQ#	Category	Requirement	Attribute	
			Priority	Difficulty
REQ0	System	Implemented in the Eclipse development environment	2	
REQ1	System	Implemented using off-the-shelf SR engine	2	
REQ2	Recording Rationale	Place spoken comments as a rationale in a primary artifact	1	
REQ3-a	Recording Rationale	Provide the ability to place links to rationale in multiple secondary artifacts a) artifacts include multiple source documents (within Eclipse “project”)	2	
REQ3-b	Recording Rationale	Provide the ability to place links to rationale in multiple secondary artifacts b) artifacts include text documents	2	
REQ3-c	Recording Rationale	Provide the ability to place links to rationale in multiple secondary artifacts c) maintain links of rationale to multiple artifacts	2	

What should be in a SRS...

- Audience
- Definitions
- ???

Managing Software Development
Requirements: Capture and Change

Session Objectives

- Continuing overview of requirements from the management perspective
- Requirements organization
- Requirements artifacts
- Managing change

Overview

Raw requirements must be analyzed to set the exact specification of what we will build. But before construction begins, the exact specification of what we must build, but be written down.

"I just invent, then wait until man comes around to needing what I've invented. "

R. Buckminster Fuller (1895-1983)

inventor, architect, engineer, mathematician – inventor of the geodesic dome

Recall: Requirements Engineering

- Systematic way of getting from need to specification – *must be planned*
  - Elicit need
  - Analyze
    - validate and quantify functional, quality attribute requirements, and constraints
    - set completion criteria
    - establish working agreement (statement of work)
  - document

Requirements Organization – 1

- After gathering requirements, the next step is to document them.
  - Affinity Groups – collections of related things
    - items conceptually related
    - items related by theme (function, feature, etc.)
  - Hierarchies – ordered dependencies
    - lower items subordinate to higher items
    - lower items elaborate or further explain higher items

Requirements Organization – 2

- Requirements should be organized in three steps
  - divide requirements into affinity groups
  - structure each group hierarchically
  - identify gaps
- Gaps and inconsistencies may require further analysis and/or elicitation.

Requirements Grouping

- To group requirements
  - scan the set of needs or ideas (yellow stickies work well)
  - identify independent themes – each theme defines a group
  - give each group a prose description
    - explain high level need
    - explain specific requirements

Requirements Hierarchy

- Organize each group hierarchically
  - priority
  - higher/lower levels of abstraction
  - identify constraints
  - work towards a tree structure
  - add new information if necessary

Annotating and Clarifying Information

- "Shall statements" can be annotated with clarifying information.
  - functional use cases
  - quality attribute scenarios
  - prototypes
  - equations
  - …
- This information may come from further elicitation, research, and/or analysis.

Identifying Gaps

- Gaps are places where the requirements are incomplete.
  - ungrouped (or un-groupable) ideas
  - lots of miscellaneous groups
  - lots of "and so on" or "etc" phases
  - inconsistent or unclear terminology
  - lots of derived requirements

Adding Information & Requirements

- Information and requirements should be added within the existing tree structure.
  - Add more information to existing general statement.
  - Add new detail to group or hierarchy descriptions.
- Regrouping should be done as a last resort.

Prioritization

- Prioritization provides guidance to designers regarding
  - importance, urgency, implementation dependencies, difficulty, availability of resources, …
- There are a variety of strategies for prioritization – *stakeholders must be involved in the prioritization*

Prioritization – Importance

- Importance of a requirement depends upon
  - the value a requirement delivers to a stakeholder
  - product differentiation
  - important of the requesting stakeholder
  - cost/benefit ratio
  - potential market share or profit

Prioritization – Urgency

- The urgency of a requirement depends upon
  - immediate operational needs deadlines
  - minimal required capability for market entry
  - incremental benefit
  - market windows
- Field first what will sell best, and design to support enhancements that will support market growth.

Prioritization – Implementation

Dependencies

- Prioritization of requirements may be dictated by the dependencies of need.
  - gives rise to derived requirements – a need to store information, often gives rise for the need for a database.
- Some requirements may seem deceptively simple, but in reality may depend upon the satisfying other unstated need (derived requirements)

Prioritization – Difficulty

- Some requirements are technically harder to satisfy than others.
  - harder requirements should be identified as risks and mitigated earlier rather than later
- Difficulty can based on unknowns and/or resource draining requirements.
- Technical difficulty should be determined by those equipped to do so.

Prioritization – Availability of Resources

- Sometimes resources are only available in windows of time.
  - domain experts
  - staff hours
  - funding

Requirements Artifacts – 1

- Requirements engineering can create all or some of the following artifacts
  - statement of work
  - requirement specification
  - concept of operation
  - initial software project plans
  - acceptance criteria

Requirements Artifacts – 2

- Do I always create all of this stuff?!
  - There are lots of rules,... so there are no rules!
  - Names are unimportant – information is.
- You must answer the following:

- ✓ what must the system do?
- ✓ what quality attributes must the system possess?
- ✓ what are the constraints?
- ✓ who responsible for what?
- ✓ how the system works in an operational environment?
- ✓ how will you build the system?
- ✓ what does it mean to be successful

Statement of Work (SOW)

- The SOW assigns responsibilities to the various stakeholders
  - who builds what, and when
  - who tests what, how, and when
  - who pays for what, and when
  - who reports to whom
  - who accepts/approves completion
  - who, how, and when changes are authorized
  - and so on...

Requirement Specification

- A requirement specification is created as a consequence of requirements analysis.
  - context
    - identify what is internal and external to the system
  - functions the system must perform
  - constraints
    - technical, cost, and schedule
  - quality attributes
  - prioritization

Concept of Operation (CONOPS)

- The concept of operation is a statement of how the system is to work or will be used in an organizational context.
  - which functions will be invoked by which users
  - how the functions will be invoked
  - under what conditions functions will be invoked
  - how information is input or output
  - how the system interfaces with other systems
- Sets the stage for use case development

Initial Project Plans

- This is a first attempt to lay out the products in the SOW along a time line
  - major reviews
  - major artifacts
  - decision points
  - deliverable artifacts
  - milestones
  - payment schedules
- It will be high level and approximate

Acceptance Criteria – 1

- The acceptance criteria is the standard to which the builders will be held to.
  - identify key requirements (again, prioritize)
  - functional acceptance criteria
  - quality (goodness) criteria
  - quality attribute criteria
  - acceptance test process
  - minimal acceptable deliverables
  - failure options

Acceptance Criteria – 2

If these criteria can be agreed to, documented, and baselined before design and development begins, everybody wins. This is called

expectation management.

"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."

Bill Gates (1955-)
Microsoft Incorporated

Change and Evolution

Never fool yourself into thinking that you can *freeze* requirements. They will change anyway. The best you can do is take a *snapshot* of the current need, anticipate, and manage changes to requirements.

"Things alter for the worse spontaneously, if they be not altered for the better designedly."

Francis Bacon (1561-1626)
Renaissance author, courtier, and father of deductive reasoning.

Am I a Pessimist?

- No, I am a realist.
  - the glass is neither half full, nor is it half empty,.. it is poorly engineered!
- Change ≠ Evolution
  - change is violent and abrupt, evolution is gradual and, if planned for, goes unnoticed.
- Once you accept that requirements will evolve and change, you can begin to analyze how volatile they really are.

Reasons For Evolution and Change

- New and/or different
  - stakeholders
    - folks come and go, new folks have different ideas
  - procedure, missions, and/or businesses
  - scope and scale
    - often we are victims of our own success
  - technologies

Outline

- Management principles
- Assessing Volatility
- Evaluating the Source of Volatility

Management Principles

- Managers must ask three tough questions:
  - How do we identify and quantify volatility?
  - How do we respond to change and evolution?
  - How do we deal with unreasonable changes?

- The answers to questions will form the basis of your requirements change and evolution strategy.

Assessing Requirements

Volatility – 1

- Most managers deal with requirements evolution and change in a *passive-reactive* way.
- In most cases this is not appropriate.
  - reluctance to fund designs that support anticipated volatility
  - what would have evolutionary becomes disruptive change
- Change should be anticipated.

Volatility – 2

- To assess these sources of volatility, we need to consider
  - the organizational environments
    - structure, policies, business/mission
  - stakeholders
    - personnel turnover, new communities
  - technological trends
- This will involve further elicitation with stakeholders – *and they may not care!*

Responses to Volatility

- There are two extremes
  - "Sorry, no can do. The requirements are frozen."
  - "OK. The customer is always right."
- Neither position is tenable.
  - Offer new features and qualities in subsequent versions.
  - Be prepared to describe the cost and impact of changes.
- Evaluate the source of volatility

Evaluating Sources

- Likely sources of requirements volatility include
  - vague requirements
  - new stakeholders
  - paradigm shifts

Vague Requirements

- Symptom: *Same stakeholders, but requirements changing with no convergence or obvious pattern.*
- Potential Sources
  - initial requirements elicitation may have been flawed
  - real users/stakeholders not identified
  - real need not identified

New Stakeholders

- Symptoms: *Radical new opinions, directions, and needs for the product. Conflict, Hostility.*
- Potential Sources
  - new requirements
  - new priorities
  - new relationships must be established
  - no historical context
  - immense learning curves to overcome for all stakeholders

Paradigm Shifts

- Symptoms: *The original requirements are viewed as irrelevant, dated, or antiquated.*
- Potential Sources
  - new design, construction philosophies (i.e. structured, verses OO)
  - new business model and/or mission objectives
  - new tools and support technologies
  - new software and/or hardware technologies

Crystal Ball

- No one has a reliable crystal ball, but we should plan for change.
  - Anticipate volatility during requirements elicitation.
  - Actively monitor key volatility factors.
  - Institute a change request evaluation and approval process – *and stick to it!*
  - Use iterative, planned, product releases in volatile requirements environment.

Summary

- We discussed
  - requirements organization
  - prioritization
  - requirements artifacts
  - managing volatile requirements

References

- Davis, A. *Software Requirements: Objects, Functions, and States*. Upper Saddle River, NJ: Prentice Hall, 1993
- Gause, D.; Weinberg, G. *Exploring Requirements: Quality Before Design*, New York, NY: Dorset House Publishing
- Pressman, R. *Software Engineering: A Practitioners Approach 5th edition*, New York, NY: McGraw Hill
- Sommerville, I. *Software Engineering 7th edition*, New York, NY: Addison Wesley

Q & A