Requirements Prioritization

Structure of the presentation

- Introduction
- Prioritization principles
- Prioritization methods
  - Prioritization scales
  - Wiegers` method
  - Pair-wise comparison technique
- Comparison between the methods

Introduction (1/3)

- There are usually more requirements than you can implement given stakeholder`s time and resource constraints... [Kar97],

Few resources

but...

Lot`s of requirements

Introduction (2/3)

- ... on the other hand, systems have *useless functions* for the users and customers!

> Large amount of the software functions are ”rarely” (19%) or ”never used” (45%) [Moi00]

Introduction (3/3)

- If we manage to implement just those requirements that meet customer and user needs
  - product **development time shortens**
  - product **development costs decrease**

”How to select a subset of the customers´ requirements and still produce a system that meets their needs?”

Requirements prioritization

Prioritization principles

"Prioritization means balancing the business benefit of each requirement against its cost and any implications it has for the architechtural foundation and future evolution of the product " [Wie99]

Why prioritize requirements? (1/2)

- Priorities help you
  - concentrate on the most important user and customer requirements
  - focus the development effort
  - manage projects more effectively
  - plan for staged deliveries
- It can also help you
  - make acceptable trade-offs among conflicting goals
  - allocate resources [Wie99], [Kar97]

Why prioritize requirements? (2/2)

”If the customers do not differentiate their requirements by importance and urgency, project managers must make these decisions on their own.” [Wie99]

”Most software organisations carry out this selection process informally and quite frequently produce systems that developers, customers and users view as suboptimal.” [Kar97]

Challenges of prioritization (1/2)

- Different stakeholders have usually different opinions about requirement´s importance and urgency.
  - People naturally have their own interest and they aren`t always willing to compromise their needs for someone else`s benefit.
- Many of the prioritization methods are either too complicated and time consuming or insufficient [Rya97].

Challenges of prioritization (2/2)

- Customers may try to avoid prioritization, because
  - they suspect that low priority requirements will never be implemented
- Developers may try to avoid prioritization, because
  - they feel bad to admit, that they can`t implement all requirements

Prioritization methods

Prioritization scales (1/3)

- Method
  - Grouping requirements in the categories
  - Usually three-level scale (e.g. Essential, Conditional, Optional [IEEE98])
- Participants
  - Different stakeholders
  - Conflicts are negotiated ”informally”

Prioritization scales (2/3)

- Pros
  - Cheap and easy to use
  - Clear technique, near common sense.

Prioritization scales (3/3)

- Cons
  - The results are in many cases just a rough estimate
  - Participant dependent method
  - Customers estimate 85% of requirements at high priority, 15% at medium and 5% at low priority
    - No desired flexibility for the project
  - In the real world low priority requirements have frequently been abandoned.

Wiegers´ method (1/4)

- Basic idea
  - Customer value depends on both
    - the benefit provided to the customer by spesific requirement
    - penalty paid by that feature [Par96]
  - Can be used only for negotiable requirements (those that are not top priority)

Wiegers´ method (2/4)

- Method
  - Estimate for each requirement using scale from 1 to 9
    - relative benefit that it provides to the customer
    - relative penalty the customer would suffer without it
    - relative cost for it
    - relative risk for it
  - Calculate the percentage of the benefit/penalty/ cost/risk that comes from each requirement

Wiegers´ method (3/4)

- Calculate priority for each requirement using the following formula

$$\text{priority} = \frac{value\%}{(cost\% \times cost, weight) + (risk\% \times risk, weight)}.$$

- Participants
  - Key customer representatives
  - Software developers

Wiegers´ method (4/4)

- Pros
  - Relative method
  - Estimation is based on several variables
  - Results are informative and clear
- Cons
  - Not much objective information is available about the method
  - Results are dependent on people´s ability to estimate value, cost and risk

Pair-wise comparison (1/6)

- Method
  - n requirements are setted up in the rows and columns of the n x n –matrix
  - Pair-wise comparison of all the requirements according the criterion from 1 to 9

Pair-wise comparison (2/6)

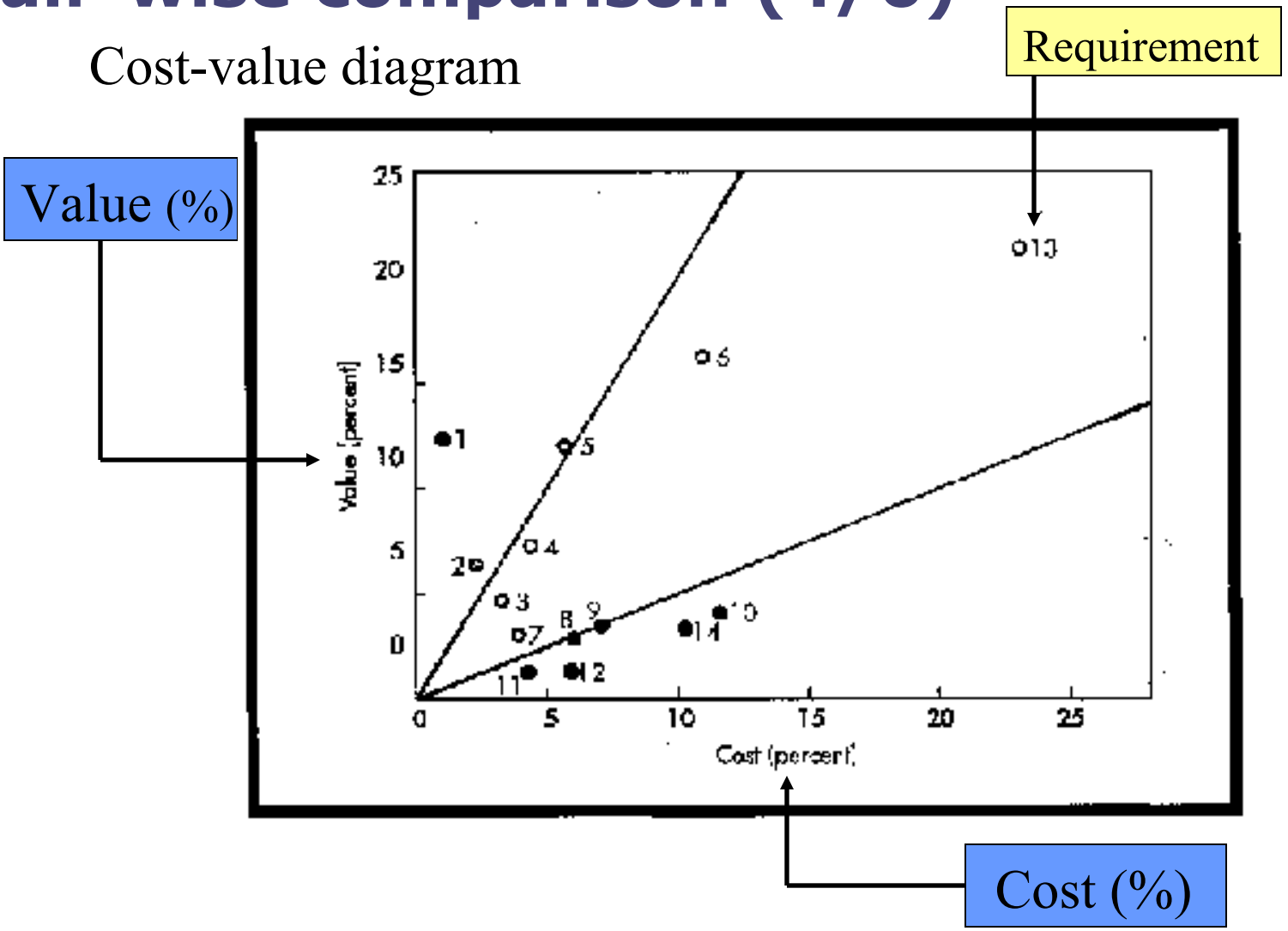
1	Of equal value
3	Slightly more value
5	Essential or strong value
7	Extreme value
9	Intermediate value

Pair-wise comparison (3/6)

- Participants
  - Requirements engineers
    - Review candidate requirements
  - Customers and users
    - Apply p-w comparison to estimate the relative value of the candidate requirements
  - Software engineers
    - Apply p-w comparison to estimate the relative cost of implementing each requirement

Pair-wise comparison (4/6)

Cost-value diagram



Pair-wise comparison (5/6)

- Pros
  - Reliable method
    - Redundancy
  - Informative results
  - Estimating relative values is found to be easier and quicker than estimating absolute values

Pair-wise comparison (6/6)

- Cons
  - Suitable only for a small number of requirements (<20)
    - Number of comparisons ½ n(n-1)
  - Does not take dependencies between the requirements in to account
  - One needs to get acquainted with the method before use

Comparison of the methods

Comparison of the methods

	Prioritization scales	**Wieger`s method**	**Pair-wise comparison**
Difficulty	Easy	Medium	Difficult
Work needed	Little	Medium	A lot
Results	Rough	Clear	Clear

References

- [Kar97] Karlsson Joachim, Ryan Kevin, “A Cost-Value Approach for Prioritizing Requirements”, IEEE Software, pp. 67-74, September/October 1997
- [Moi00] Moisiadis Frank, Prioritising Use Cases and Scenarios, IEEE 2001
- [Wie99] Wiegers, K., Software Requirements, Microsoft Press, Redmond, Washington, 1999.