

ECE450 – Software Engineering II

Today: Requirements Engineering: Prioritization of Requirements

adapted from Steve Easterbrook's
material on Requirements Engineering

ECE450 - Software Engineering II

1

Prioritization - Overview

- Why is prioritization needed?
 - Basic trade-offs
- Cost-Value approach
 - Sorting requirements by cost/value
 - Estimating relative costs/values using AHP
- What if stakeholders disagree?
 - Visualizing differences in priority
 - Resolving disagreements

ECE450 - Software Engineering II

2

Basics of prioritization

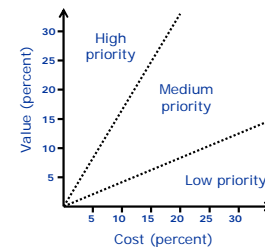
- Need to select what to implement
 - Customers (usually) ask for way too much
 - Balance time-to-market with amount of functionality
 - Decide which features go into the next release
- For each requirement/feature, ask:
 - How important is this to the customer?
 - How much will it cost to implement?
 - How risky will it be to attempt to build it?
- Perform Triage:
 - Some requirements **must** be included
 - Some requirements should definitely be excluded
 - That leaves a pool of "nice-to-haves", which we must select from.

ECE450 - Software Engineering II

3

A Cost-Value Approach

- Calculate return on investment
 - Assess each requirement's importance to the project as a whole
 - Assess the relative cost of each requirement
 - Compute the cost-value trade-off:



ECE450 - Software Engineering II

4

Estimating Cost and Value

- Two approaches:
 - Absolute scale (e.g. dollar values)
 - Requires much domain experience
 - Relative values (e.g. less/more; a little, somewhat, very)
 - Much easier to elicit
 - Prioritization becomes a sorting problem
- Ensure that estimates come from proper sources
 - Cost is best estimated by developers
 - Value is best estimated by customers
- Comparison Process - options
 - Basic sorting - for every pair of requirements (i,j), ask if i>j?
 - E.g. bubblesort - start in random order, and swap each pair if out of order
 - requires $n*(n-1)/2$ comparisons
 - Construct a Binary Sort Tree
 - Requires $O(n \log n)$ comparisons
 - Construct a Minimal Spanning Tree
 - for each pair (Ri, Ri+1) get the distance between them
 - Requires n-1 comparisons

ECE450 - Software Engineering II

5

Some complications

- Hard to *quantify* differences
 - easier to say "x is more important than y"...
 - ...than to estimate by how much.
- Not all requirements comparable
 - E.g. different level of abstraction
 - E.g. core functionality vs. customer enhancements
- Requirements may not be independent
 - No point selecting between X and Y if they are mutually dependent
- Stakeholders may not be consistent
 - E.g. If X > Y, and Y > Z, then presumably X > Z?
- Stakeholders might not agree
 - Different cost/value assessments for different types of stakeholder

ECE450 - Software Engineering II

6

Analytic Hierarchy Process (AHP)

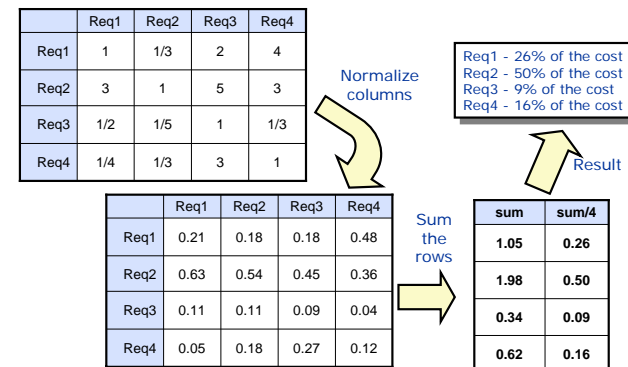
adapted from Karlsson & Ryan, 1997

- Create n x n matrix (for n requirements)
 - For element (x,y) in the matrix enter:
 - 1 - if x and y are of equal value
 - 3 - if x is slightly more preferred than y
 - 5 - if x is strongly more preferred than y
 - 7 - if x is very strongly more preferred than y
 - 9 - if x is extremely more preferred than y
 - (use the intermediate values, 2,4,6,8 if compromise needed)
 - ...and for (y,x) enter the reciprocal.
- Estimate the eigenvalues:
 - E.g. "averaging over normalized columns"
 - Calculate the sum of each column
 - Divide each element in the matrix by the sum of it's column
 - Calculate the sum of each row
 - Divide each row sum by the number of rows
- This gives a value for each requirement:
 - ...giving the estimated percentage of total value of the project

ECE450 - Software Engineering II

7

AHP Example – Estimating costs

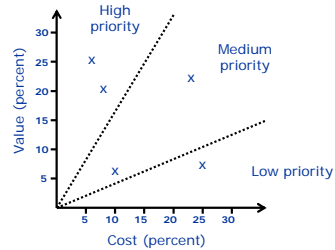


ECE450 - Software Engineering II

8

Plot ROI graph

- Do AHP process twice:
 - Once to estimate relative value
 - Once to estimate relative cost
- Use results to calculate ROI ratio:

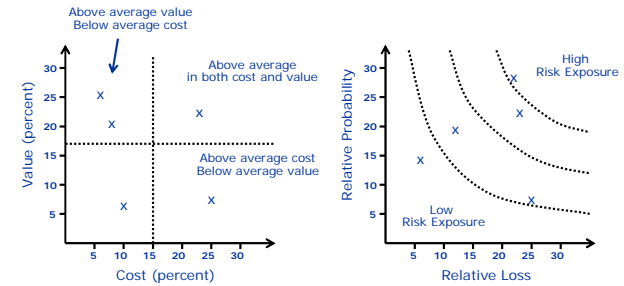


ECE450 - Software Engineering II

9

Other selection criteria

- ROI ratio is not the only way to group requirements

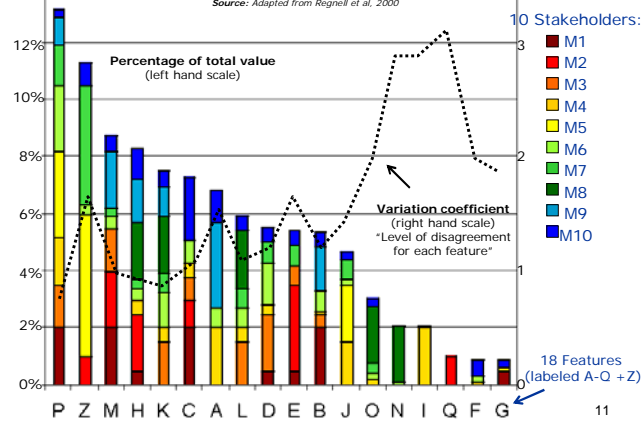


ECE450 - Software Engineering II

10

Visualizing "Value by Stakeholder"

Source: Adapted from Regnell et al., 2000

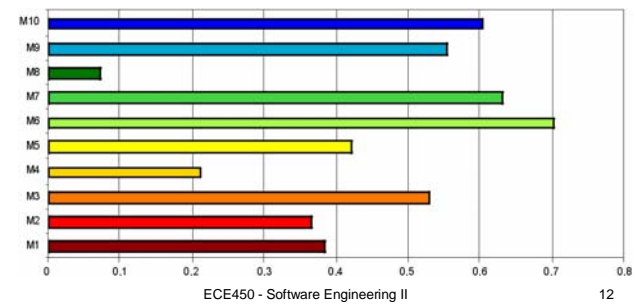


11

Visualizing stakeholder satisfaction

Adapted from Regnell et al., 2000

- Graph showing correlation between stakeholder's priorities and the group's priorities
 - Can also be thought of as "influence of each stakeholder on the group"



ECE450 - Software Engineering II

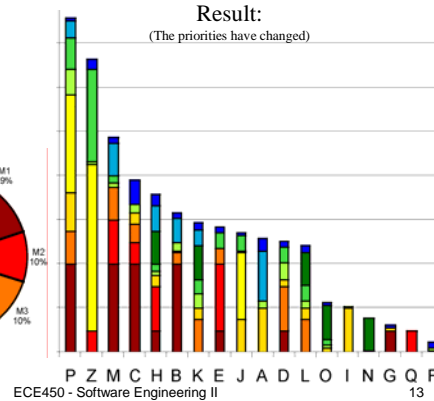
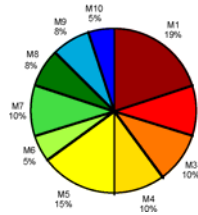
12

Assigning weight to stakeholders

Adapted from Regnell et al., 2000

- Weight each stakeholder
 - E.g. to reflect credibility?
 - E.g. to reflect size of constituency represented?

- Example:



ECE450 - Software Engineering II 13

Resolving stakeholder conflict

- Causes of Conflict
 - Deutsch (1973):
 - control over resources
 - preferences and nuisances (tastes or activities of one party impinge upon another)
 - values (a claim that a value or set of values should dominate)
 - beliefs (dispute over facts, information, reality, etc.)
 - the nature of the relationship between the parties.
 - Robbins (1989):
 - communicational (insufficient exchange of information, noise, selective perception)
 - structural (goal compatibility, jurisdictional clarity, leadership style)
 - personal factors, (individual value systems, personality characteristics).
- Interesting Results
 - deviant behaviour & conflict are normal in small group decision making
 - more aggression and less co-operation when communication is restricted
 - a decrease in communication tends to intensify a conflict (the contact hypothesis)
 - heterogeneous teams experience more conflict;
 - homogeneous groups are more likely to make high risk decisions (groupthink)
 - effect of personality is overshadowed by situational and perceptual factors

ECE450 - Software Engineering II

14

Basic approaches to conflict resolution

- Negotiation
 - ...is collaborative exploration:
 - participants seek a settlement that satisfies all parties as much as possible.
 - also known as:
 - integrative behaviour
 - constructive negotiation
 - distinct from:
 - distributive/competitive negotiation
- Competition
 - is maximizing your own gain:
 - no regard for the degree of satisfaction of other parties.
 - but not necessarily hostile!
 - Extreme form:
 - when all gains by one party are at the expense of others
 - I.e a zero-sum game.
- Third Party Resolution
 - participants appeal to outside source
 - the rule-book, a figure of authority, or the toss of a coin.
 - can occur with the breakdown of either negotiation or competition as resolution methods.
 - **judicial**: cases presented by each participant are taken into account
 - **extra-judicial**: a decision is determined by factors other than the cases presented
 - (e.g. relative status of participants)
 - **arbitrary**: e.g. toss of a coin

ECE450 - Software Engineering II

15