# Lecture 9: Estimation and Prioritization

- ➜ Project planning
- ➜ Estimating Effort
- ➜ Prioritizing Stakeholder's needs
- ➜ Trade-offs between stakeholder goals

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Project Planning

**Given:**
- A list of customer requirements
  - E.g. a set of use cases, a set of change requests, etc.

**Estimate:**
- How long each one will take to implement (cost)
- How important each one is (value)

**Plan:**
- Which requests should be included in the next release

**Complication:**
- Customers care about other stuff to: quality, performance, security, usability,…

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Principles of Management

### A manager can control 4 things:

- **Resources** (can get more dollars, facilities, personnel)
- **Time** (can increase schedule, delay milestones, etc.)
- **Product** (can reduce functionality - e.g. scrub requirements)
- **Risk** (can decide which risks are acceptable)

### Approach (applies to any management)

- Understand the goals and objectives
  - quantify them where possible
- Understand the constraints
  - if there is uncertainty, use probability estimates
- Plan to meet the objectives within the constraints
- Monitor and adjust the plan
- Preserve a calm, productive, positive work environment

### Note:

- You cannot control what you cannot measure!

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Strategies

### Fixed Product

1. Identify customer requirements
2. Estimate size of software needed to meet them
3. Calculate time required to build this much software
4. Get customer to agree to the cost & schedule

### Timeboxing

1. Fix a date for next release
2. Obtain prioritized list of requirements
3. Estimate effort for each requirements
4. Select requirements off the list until the “box” is full

### Fixed Cost

1. Agree with customer how much they wish to spend
2. Obtain prioritized list of requirements
3. Estimate cost of each requirement
4. Select requirements off the list until the “cost” is used up

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Estimating Effort: COCOMO

***Source:*** *Adapted from van Vliet, 1999, section 7.3.2*

### COnstructive COst Model (COCOMO)

- Used to predict cost of a project from a measure of size (lines of code)
- Basic model is:

effort project specific factors

$$E = aL^b$$

lines of code

### Modeling process

- Establish type of project (organic, semidetached, embedded)
  - this gives sets of values for a and b
- Identify the component modules, and estimate L for each module
- Adjust L according to how much is reused
  - COCOMO has a model for adjusting according to how much design, code and integration data is reused
- Compute effort for each module using $E = aL^b$
- Adjust E according to difficulty of the project
  - COCOMO identifies 15 effort multipliers to take into account
  - **Product attributes:** eg required reliability, complexity, database size
  - **Computer attributes:** eg execution time constraints, storage constraints, etc.
  - **Personnel attributes:** eg capability & experience of analysts and programmers,
  - **Project attributes:** eg use of CASE tools, programming language, schedule
- Compute time using $T = cE^d$
  - c and d provided for different project types like a and b were

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Estimating Size: Function Points

***Source:*** *Adapted from van Vliet, 1999, section 7.3.5*

### Function Points

- used to calculate size of software from a statement of the problem
- tries to address variability in lines of code estimates used in models such as COCOMO
  - e.g. because SLOC varies with different languages
- Originally for information systems, although other variants exist
- Basic model is:

metric from problem statement

$$FP = a_1 I + a_2 O + a_3 E + a_4 L + a_5 F$$

weighting factor for this metric

### Example

- Sets of weightings ($a_i$) provided for different types of project
- Measure properties of the problem statement:
  - I = number of user inputs (data entry)
  - O = number of user outputs (reports, screens, error messages)
  - E = number of user queries
  - L = number of files
  - F = number of external interfaces (to other devices, systems)
- Example calculation:
  - $FP = 4I + 5O + 4E + 10L + 7F$

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Agile Estimating

### Estimation in Practice:

People tend to underestimate effort needed

Most estimates are made to please the {boss, customer, …}

Easier to estimate small chunks of work than large ones

### Three-point estimating

Gets much better estimates than asking for a range

w = worst possible case

m = most likely case

b = best possible case

$$E = \sum_i \frac{w_i + 4m_i + b_i}{6}$$

…and don't forget: effort < duration !!

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## A Cost-Value Approach

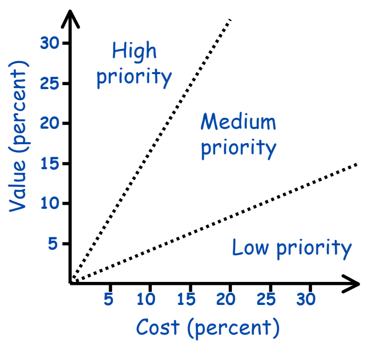
***Source:*** *Adapted from Karlsson & Ryan 1997*

### 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.



© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## 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

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Stakeholders

### Stakeholder analysis:

- Identify all the people who must be consulted during information acquisition

### Example stakeholders

- Users
  - concerned with the features and functionality of the new system
- Designers
  - want to build a perfect system, or reuse existing code
- Systems analysts
  - want to "get the requirements right"
- Training and user support staff
  - want to make sure the new system is usable and manageable
- Business analysts
  - want to make sure "we are doing better than the competition"
- Technical authors
  - will prepare user manuals and other documentation for the new system
- The project manager
  - wants to complete the project on time, within budget, with all objectives met.
- "The customer"
  - Wants to get best value for money invested!

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Identifying Stakeholders' Goals

***Source:*** *Adapted from Anton, 1996.*

### Approach

- Focus on ***why*** a system is required
- Express the 'why' as a set of stakeholder goals
- Use goal refinement to arrive at specific requirements
- Goal analysis
  - document, organize and classify goals
- Goal evolution
  - refine, elaborate, and operationalize goals
- Goal hierarchies show refinements and alternatives

### Advantages

- Reasonably intuitive
- Explicit declaration of goals provides sound basis for conflict resolution

### Disadvantages

- Captures a static picture - what if goals change over time?
- Can regress forever up (or down) the goal hierarchy

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Goal Modeling

### (Hard) Goals:

- Describe functions that must be carried out. E.g.
  - Satisfaction goals
  - Information goals

### Softgoals:

- Cannot really be fully satisfied. E.g.
  - Accuracy
  - Performance
  - Security
  - …

### Also classified temporally:

- Achieve/Cease goals
  - Reach some desired state eventually
- Maintain/Avoid goals
  - Keep some property invariant
- Optimize
  - A criterion for selecting behaviours

### Agents:

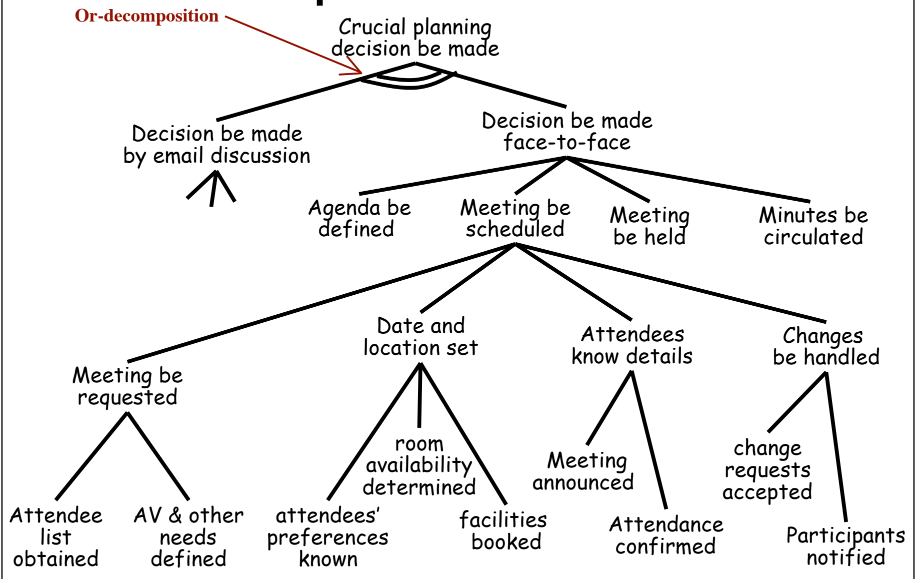
- Owners of goals
- Choice of when to ascribe goals to agents:
  - Identify agents first, and then their goals
  - Identify goals first, and then allocate them to agents during operationalization

### Modelling Tips:

- Multiple sources yield better goals
- Associate stakeholders with each goal
  - reveals viewpoints and conflict
- Use scenarios to explore how goals can be met
- Explicit consideration of obstacles helps to elicit exceptions

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Example Goal Elaboration



© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Goal Analysis

### Goal Elaboration:

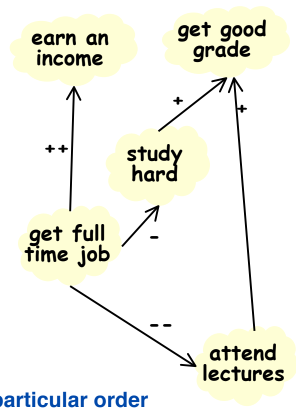
- "Why" questions explore higher goals (context)
- "How" questions explore lower goals (operations)
- "How else" questions explore alternatives

### Relationships between goals:

- One goal **helps** achieve another (+)
- One goal **hurts** achievement of another (-)
- One goal **makes** another (++)
  - Achievement of goal A guarantees achievement of goal B
- One goal **breaks** another (--)
  - Achievement of goal A prevents achievement of goal B
- Precedence ordering – if goals must be achieved in a particular order

### Obstacle Analysis:

- Can this goal be obstructed, if so how?
- What are the consequences of obstructing it?

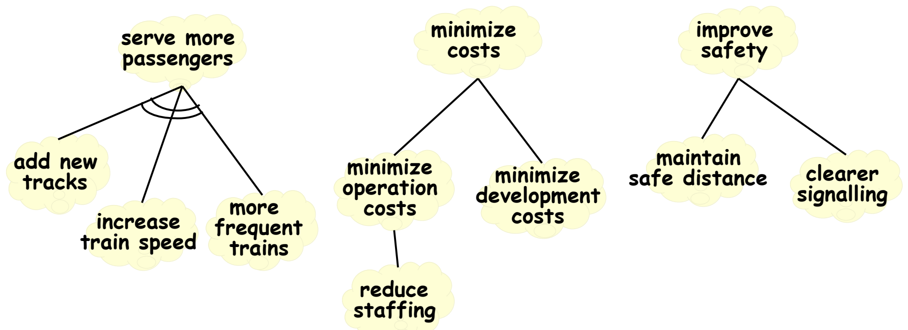


© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Softgoals

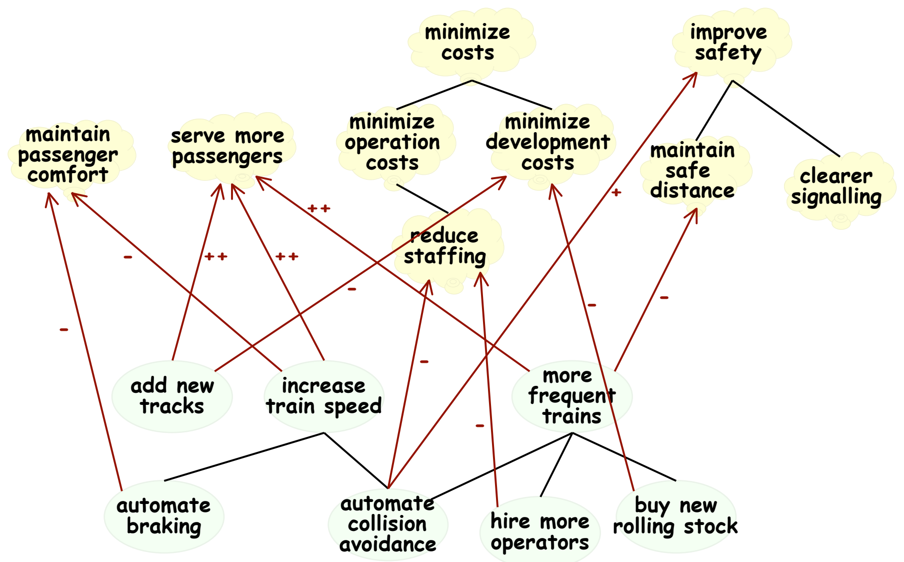
### Some goals can never be fully satisfied

- Treat these as softgoals
  - E.g. "system be easy to use"; "access be secure"
  - Also known as 'non-functional requirements'; 'quality requirements'
- Will look for things that contribute to satisficing the softgoals
- E.g. for a train system:



© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Softgoals as selection criteria



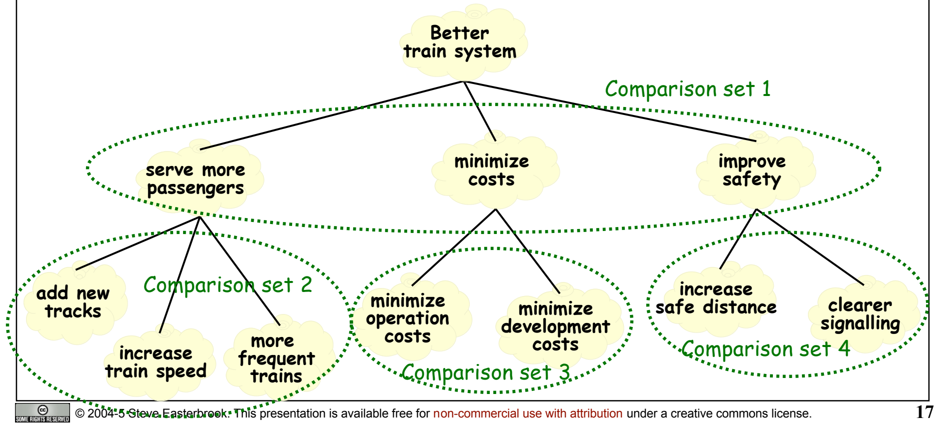
© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Hierarchical Prioritization

Group Requirements into a hierarchy

- E.g. A goal tree
- E.g. A NFR tree

Only make comparisons between branches of a single node:



© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

## Advice from ICONIX

Plan at appropriate detail

Negotiate the scope (when faced with fixed deadline)

Customer dictates priority

Adjust the plan to fit reality (small release cycles help)

Get feedback on progress and risks

Try to get it right first time (rather than fix it later)

Use 3 types of release: internal, investigative, production

Plan to refactor when necessary (avoid rot)

Consider high impact decisions during early iterations

© 2004-5 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.