

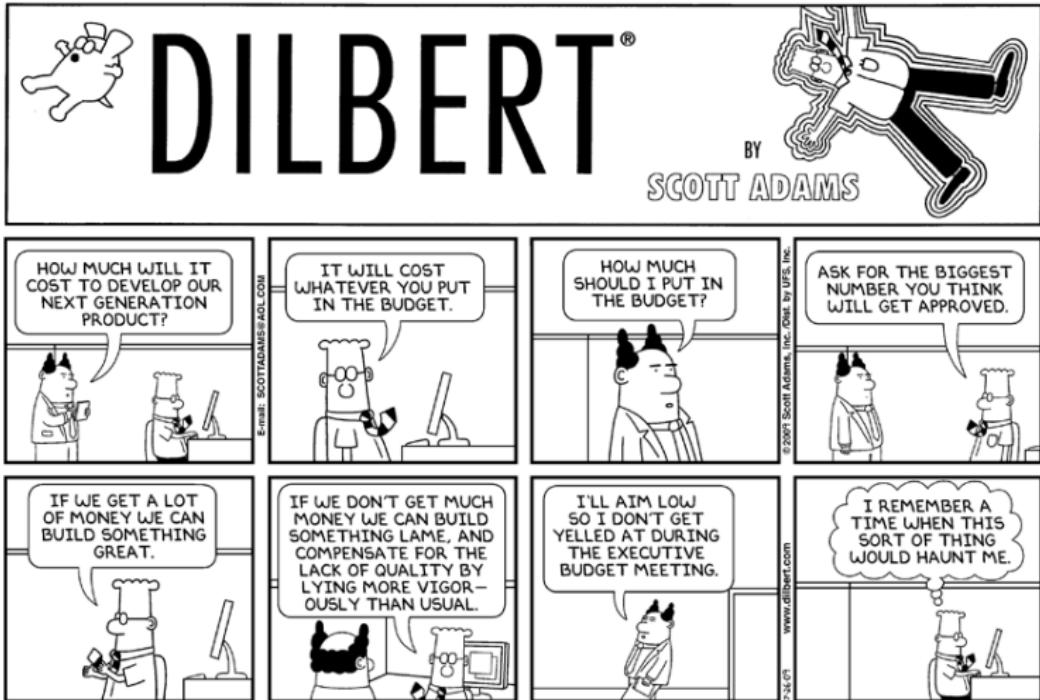
Time-planning and Scheduling

Large Systems Development

Anders Kalhauge



Fall 2018



Project Status

Estimation

Units of work

Approaches

Planning

Critical Path

Burndown charts

The Iron Triangle

Individual group meetings

Group A 10:00

Group B 10:15

Group C 10:30

Group D 10:45

Group E 11:15

Group F 11:30

Group G 11:45

Project Status

Estimation

Units of work

Approaches

Planning

Critical Path

Burndown charts

The Iron Triangle

- Workhours or man hours
- Source Lines Of Code (SLOC)
- Function Points
- Story Points

- Physical lines of code (comment lines excluded)
- Logical lines of code (language dependent)

```
if (a > 100) println("Hello"); // How many lines?
```

1 Physical (LOC), 2 Logical (LLOC), and 1 comment

```
// How many lines?  
if (a > 100) {  
    println("Hello");  
}
```

3 Physical (LOC), 2 Logical (LLOC), and 1 comment

Express the business functionality

- Outputs
- Inquiries
- Inputs
- Internal files
- External interfaces

1. Put all the stories (objects) in front of developers.
2. Let them choose one story (object) that will get the number Effort=1. Typically it is a small story like „I as a user want to save a file to disk“.
3. Then compare all remaining stories relatively to referenced story and other, already estimated stories.

- Expert estimation
 - Planning Poker
- Formal estimation model
 - COCOMO
- Combination-based estimation



- **A Moderator**, who will not play, chairs the meeting.
- **The Product Owner** provides a short overview of one user story to be estimated.
- **The team** is given an opportunity to ask questions and discuss to clarify assumptions and risks. A summary of the discussion is recorded, e.g. by the Moderator.

1. Each individual lays a card face down representing their estimate for the story. Units used vary - they can be days duration, ideal days or story points. During discussion, **numbers must not be mentioned at all** in relation to feature size to avoid anchoring.
2. Everyone calls their cards simultaneously by turning them over.
3. People with high estimates and low estimates are given a **soap box** to offer their justification for their estimate and then discussion continues.
4. Repeat the estimation process until a consensus is reached. The developer who was likely to own the deliverable has a large portion of the "consensus vote", although the Moderator can negotiate the consensus.



Program size is expressed in thousands of lines of code *KLOC*.
COCOMO applies to three classes of software projects:

- Organic projects
 - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects
 - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects
 - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.

- Effort Applied: $E = a_b \times KLOC^{b_b}$ man-months
- Development Time: $D = c_b \times E^{d_b}$ months
- People required: $P = E/D$

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Project Status

Estimation

Units of work

Approaches

Planning

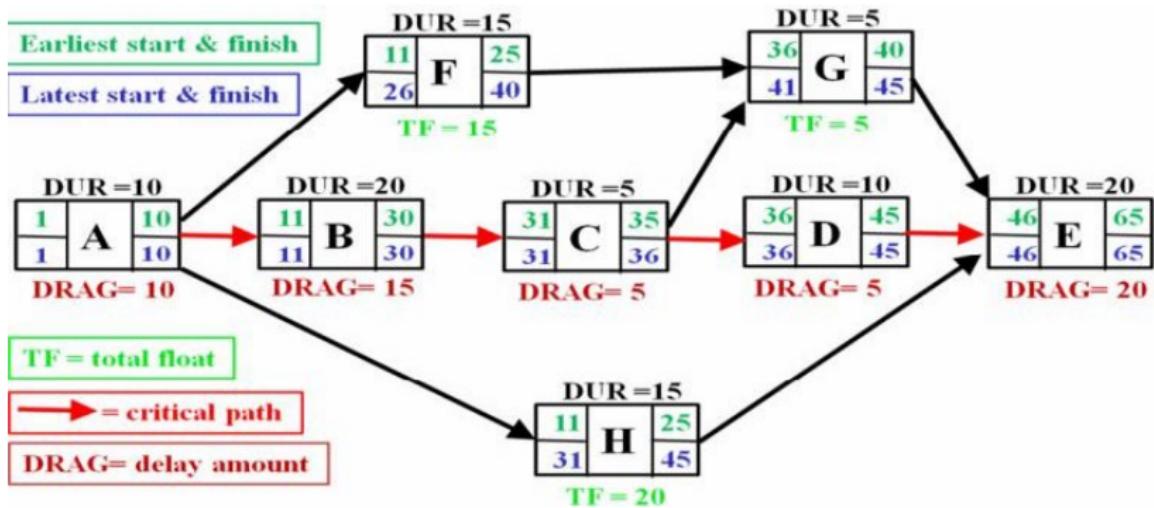
Critical Path

Burndown charts

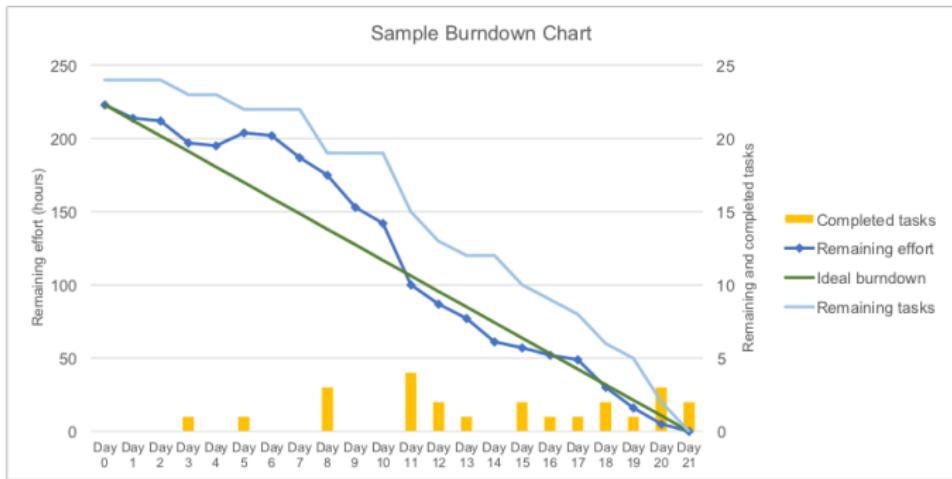
The Iron Triangle

- List of all activities
- Duration of each activity (see estimation)
- Dependencies between activities
- Logical end points (milestones, deliverables)

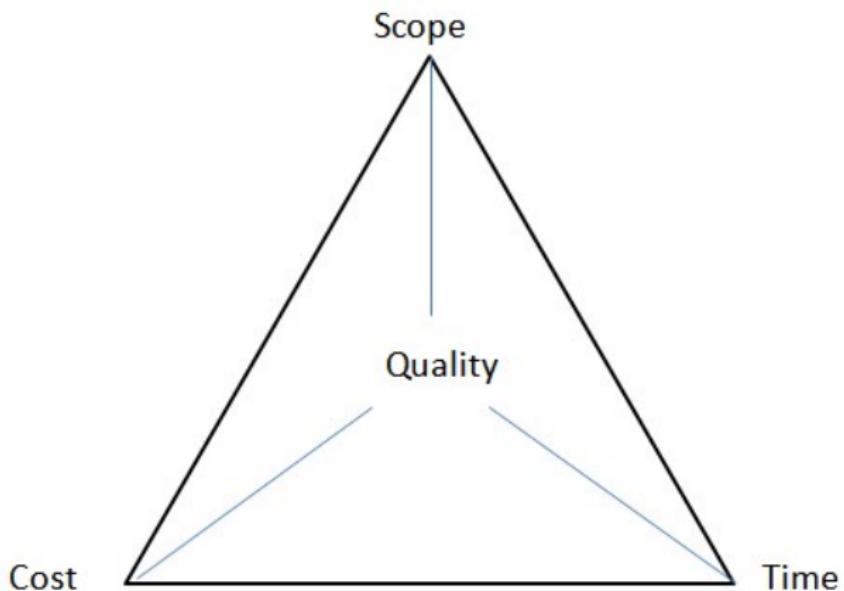
Critical Path



Burndown Chart



The Iron Triangle



CLASSIC If **scope** is the driver **quality** will be adjusted due to bad planning or unforeseen changes.

AGILE If **quality** is the driver **scope** will be adjusted due to bad planning or unforeseen changes.

- <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>
- https://en.wikipedia.org/wiki/Planning_poker
- <https://play.google.com/store/apps/details?id=artarmin.android.scrum.poker&hl=da>
- <https://itunes.apple.com/us/app/scrum-poker-planning-cards/id893134104?mt=8>