



Foto: Thomas Josek

Software Engineering

SW Management

Software & Systems Engineering | Prof. Dr. Andreas Vogelsang | 15.01.2024



@andivogelsang



vogelsang@cs.uni-koeln.de

Learning Goals for Today

- Know what SW project management is
- Know the importance of people management
- Know the challenges of estimations
- Know methods of effort estimations
- Know methods of project planning
- Know how to monitor and control SE processes



SW Project Management

Software Development Project

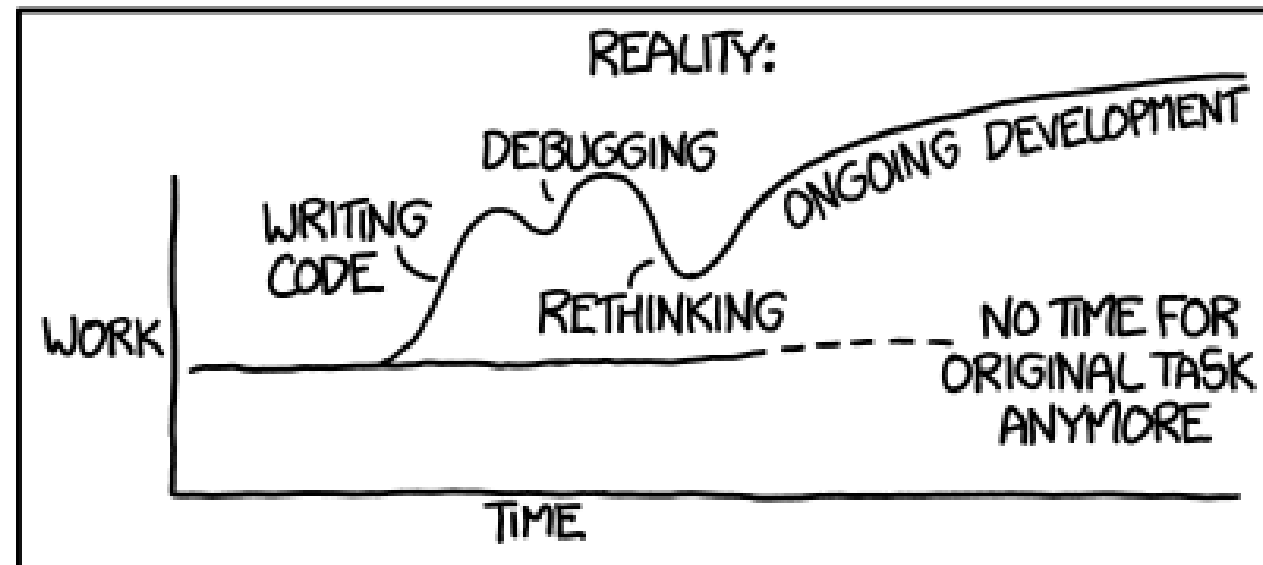
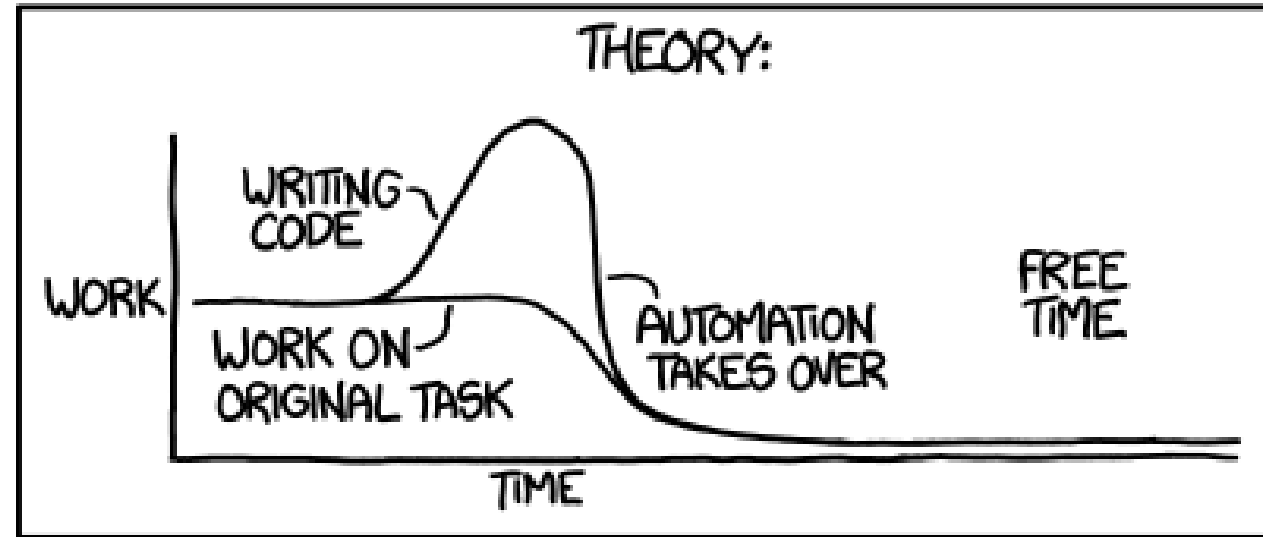
Software Development Project

- aka. software engineering project
- temporary activity with start and end date
- has goals
 - creation/modification of a software product
 - creation/modification of components for future projects
 - gain experience/knowledge
 - capacity utilization ([Mitarbeiterauslastung](#))
 - ...
- is successful if goals are largely fulfilled

A personal side note and view

- Project Management has been a key activity in SW Engineering because SW development has long been organized in *projects* (i.e., with a clear end).
- Nowadays, most SW is developed incrementally with a long-term vision and no fixed end.
- Software development has changed from *project* to *product* development.
- Classical methods from project management have become less relevant.

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



Project Management

Motivation

Good management cannot guarantee project success. However, bad management usually results in project failure: The software may be delivered late, cost more than originally estimated, or fail to meet the expectations of customers.

Goals of Project Management

- deliver the software to the customer at the agreed **time**
- keep overall **costs** within budget
- deliver software that meets the customer's **expectations**
- maintain a coherent and well-functioning development **team**

Project Management depends on...

Company size: large companies have management hierarchies and reporting / budgeting / approval processes

Customers: external customers (i.e., government agencies) usually have policies

Software size: large systems require multiple development teams in different companies / locations

Software type: safety-critical systems require all design decisions to be documented

Dev. process: project management heavily depends on process model

Activities in Project Management

Project Planning

Project managers are responsible for **planning**, **estimating**, and **scheduling** project development and assigning people to tasks. They supervise the work to ensure that it is carried out to the required standards, and they **monitor progress** to check that the development is on time and within budget.

Risk Management

Project managers must **assess the risks** that may affect a project, monitor these risks, and act when problems arise.

People Management

Project managers are responsible for **managing a team** of people. They must choose people for their team and establish ways of working that lead to effective team performance.

Reporting

Project managers are usually responsible for **reporting on the progress** of a project to customers and to the managers of the company developing the software. They must be able to communicate at a range of levels, from detailed technical information to management summaries.

Proposal Writing

The first stage in a software project may involve writing a proposal to **win a contract** to carry out an item of work. The proposal describes the objectives of the project and how it will be carried out. It usually includes **cost and schedule estimates** and justifies why the project contract should be awarded to a particular organization or team. Proposal writing is a critical task as the survival of many software companies depends on having enough proposals accepted and contracts awarded.

Risk Management

Classification of Risks

Project Risks: affect project schedule or resources: loss of an experienced system architect may result in longer development time

Product Risks: affect software quality: purchased component may not scale

Business Risks: affect organization/company: product of a competitor may reduce number of sales

Risk Assessment

Probability: rare, unlikely, possible, probable, almost certain

Severity: insignificant, minor, moderate, major, catastrophic

Stages in Risk Management

1. **Risk Identification:** identify possible project, product, and business risks
2. **Risk Analysis:** assess likelihood and consequences
3. **Risk Planning:** plan how to address risks: avoidance or minimization of effects
4. **Risk Monitoring:** regularly assess risks and revise plans if needed

Risks in Agile Development

Reduced risks for requirements changes, increased risks for loss of stuff due to fewer documentation

Risk Assessment

Likelihood of residual risk

Almost Certain 5	5 <i>Supplementary Issue</i>	10 <i>Issue</i>	15 <i>Unacceptable</i>	20 <i>Unacceptable</i>	25 <i>Unacceptable</i>
Probable 4	3 <i>Acceptable</i>	8 <i>Supplementary Issue</i>	12 <i>Issue</i>	16 <i>Unacceptable</i>	20 <i>Unacceptable</i>
Possible 3	3 <i>Acceptable</i>	6 <i>Supplementary Issue</i>	9 <i>Issue</i>	12 <i>Issue</i>	15 <i>Unacceptable</i>
Unlikely 2	2 <i>Acceptable</i>	4 <i>Acceptable</i>	6 <i>Supplementary Issue</i>	8 <i>Supplementary Issue</i>	10 <i>Issue</i>
Rare 1	1 <i>Acceptable</i>	2 <i>Acceptable</i>	3 <i>Acceptable</i>	4 <i>Acceptable</i>	5 <i>Issue</i>
	Insignificant	Minor	Moderate	Major	Catastrophic

People Management

Motivation

The people working in a software organization are its **greatest assets**. It is expensive to recruit and retain good people, and it is up to software managers to ensure that the engineers working on a project are as **productive** as possible. In successful companies and economies, this productivity is achieved when people are respected by the organization and are assigned responsibilities that reflect their skills and experience.

Critical Factors

Consistency: treat people comparably with similar rewards

Respect: let all people contribute and respect their differences in skills

Inclusion: consider views of minorities (juniors, diff. cultural background,...)

Honesty: manager is honest about own skills and team performance

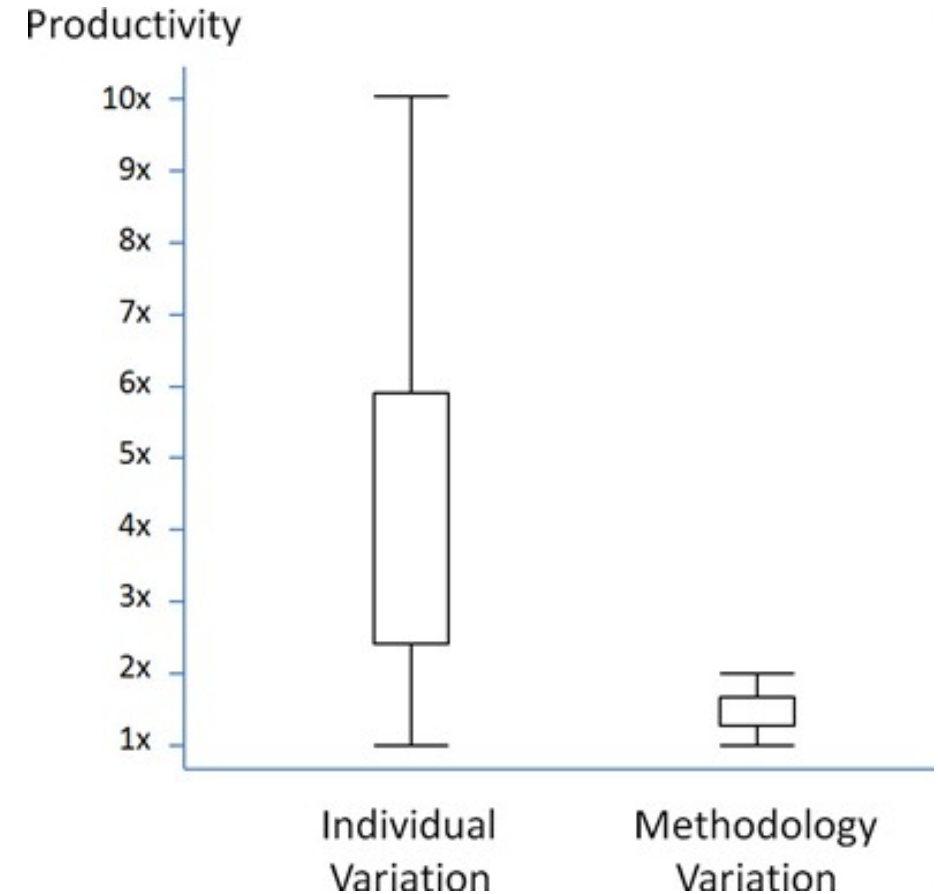
Teamwork

Most professional software is developed by project teams that range in size from two to several hundred people. However, as it is impossible for everyone in a large group to work together on a single problem, **large teams are usually split** into several smaller groups. Each group is responsible for developing part of the overall system.

The Mythical 10x Engineer

10x Engineer

- Metaphor for strong individual differences in productivity of individuals in SW engineering
- An engineer that is 10x as productive as the least productive engineer
- Studies indicate large individual differences
 - Time for coding 20:1
 - Time for debugging 25:1
 - Program size 5:1
 - Execution times 10:1
- Differences cannot be explained by
 - Use of a specific programming language
 - Experience in years



10x in Teams?

Lotus 1-2-3 v3 (1990)

- 260 Person-Years
- 400,000 Lines of Code

MS Excel 3.0 (1990)

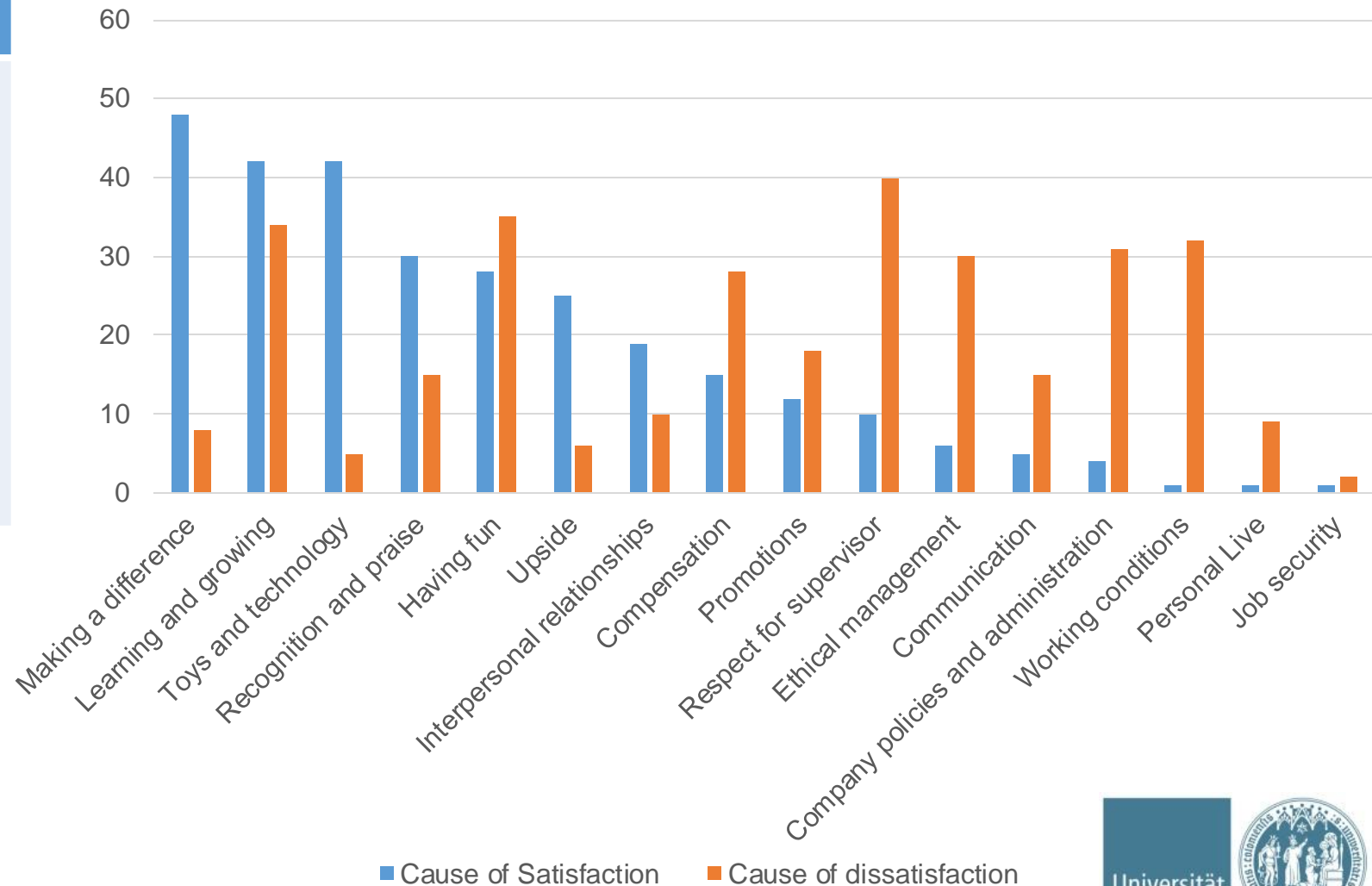
- 50 Person-Years
- 649,000 Lines of Code

Differences are more in
the order of 3x

Motivating People

Two-factor theory

- a.k.a. Herzberg's motivation-hygiene theory
- Factors for satisfaction (motivators) and dissatisfaction (hygiene factors) are different
- Mitigating dissatisfaction factors does not necessarily lead to satisfaction





Effort Estimation

WHEN A USER TAKES A PHOTO,
THE APP SHOULD CHECK WHETHER
THEY'RE IN A NATIONAL PARK...

SURE, EASY GIS LOOKUP.
GIMME A FEW HOURS.

... AND CHECK WHETHER
THE PHOTO IS OF A BIRD.

I'LL NEED A RESEARCH
TEAM AND FIVE YEARS.



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

Task: Estimate the time needed

- Project A:
A simple web version of Monopoly with street names of Cologne
 - Team: Just you
- Project B:
A smartphone app for a bank
 - Team: You and a team of 4 developers;
one with experience in iPhone app development;
one with security expertise
- Estimate the time for development in 8h days (1 month = 20 days, 1 year = 220 days)

General Approaches to Effort Estimation

- Expert estimation approaches
 - What we just did
 - The estimate is produced based on judgmental processes by experts
- Analytical approaches
 - The estimate is based on an underlying model and may be parameterized by historical data

Planning Poker

Planning Poker

- Expert estimation approach
- Based on relative instead of absolute effort (*story points* instead of hours)

Process (everything in one meeting)

1. Distribute cards with numbers on them in the team
2. The facilitator will read out a user story or task
3. The user story or task is discussed briefly
4. Simultaneously, every team member flips a card that reflects the prospective effort
5. Depending on the variance in numbers:
 - a) Take the median value as estimate
 - b) Ask outliers for reasons, discuss and vote again
6. Move on to the next user story or task

Possible card decks

- Playing Cards: Ace, 2, 3, 4, 5, 6... King
- Modified Fibonacci: 0, ½, 1, 3, 5, 8, 13, 20, 40, 100
- Sequential: 0, ½, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15...
- T-Shirt Sizes: XXS, XS, S, M, L, XL, XXL
- With Icons: Infinity means it can't be done, coffee cup means I need a break

Reference Story

In agile approaches, effort is often not estimated absolutely but relatively to a *reference story*.

How much easier/harder is a task in comparison to the reference story

Planning Poker Demo

- A Mensa web application
 - Team: A team of 4 students (including you)
- Reference story:
As a user, I want to browse the Mensa menu of the day so that I can decide whether to go to the Mensa
Estimation: 5 SP
- <https://planningpokeronline.com/8gzlZ0SIWhP8LwdfAw97/>



Wideband Delphi

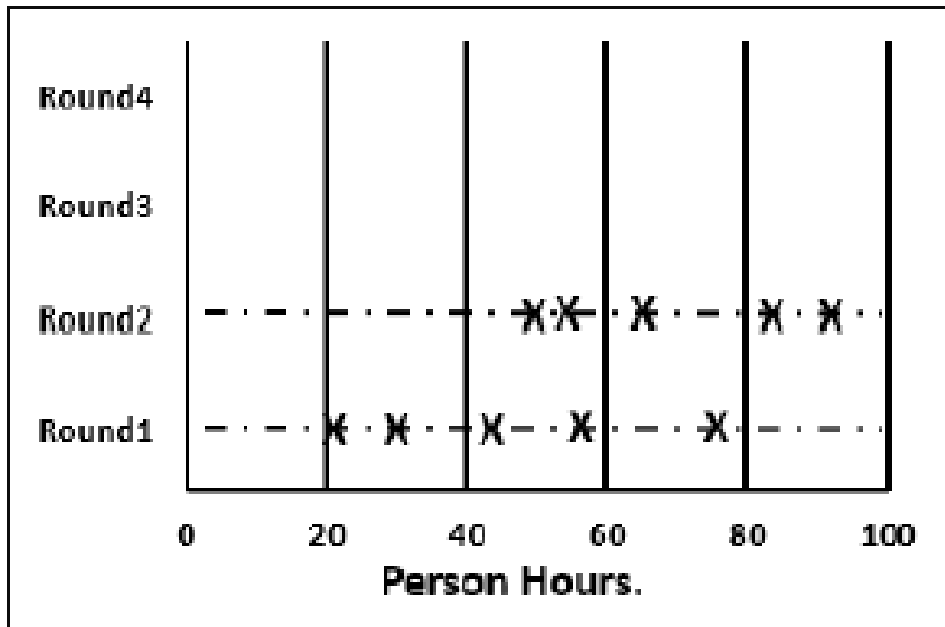
Wideband Delphi Method

- Expert estimation approach
- Extension of Delphi method

Process

1. Kickoff meeting: specification and task list
2. Each estimator independently estimates all tasks
3. Estimation meeting:
 - a) Visualize estimates (anonymously),
 - b) discuss problems and assumptions,
 - c) change estimates, repeat
4. Do the steps of 3. until
 - results are converged to an acceptably narrow range.
 - all team members are unwilling to change their latest estimates.
 - the allotted estimation meeting time is over.

Wideband Delphi Estimation Sheet						
Project: <Project Name>				Estimation Units: Person Hours		
Estimation Team Member: <Name>				Date: <MM-DD-YY>		
Task	Initial Estimate	Change 1	Change 2	Change 3	Change 4	Final
Task1	n_1					
Task2	n_2					
Task3	n_3					
Task4	n_4					
Task5	n_5					
Task6	n_6					
Task7	n_7					
Task8	n_8					
Net Change						
Total		$\sum n_i$				



Constructive Cost Model (Cocomo)

Constructive Cost Model (Cocomo)

- Data-driven cost estimation model based on past experience
- Cocomo I (1981): based on 63 projects at one company
- Cocomo II (2000): based on 161 projects
- 3 levels of detail: basic, intermediate, detailed

Basic Cocomo

Input

- Estimated LOC
- Project difficulty: simple, complicated, complex

Estimated effort

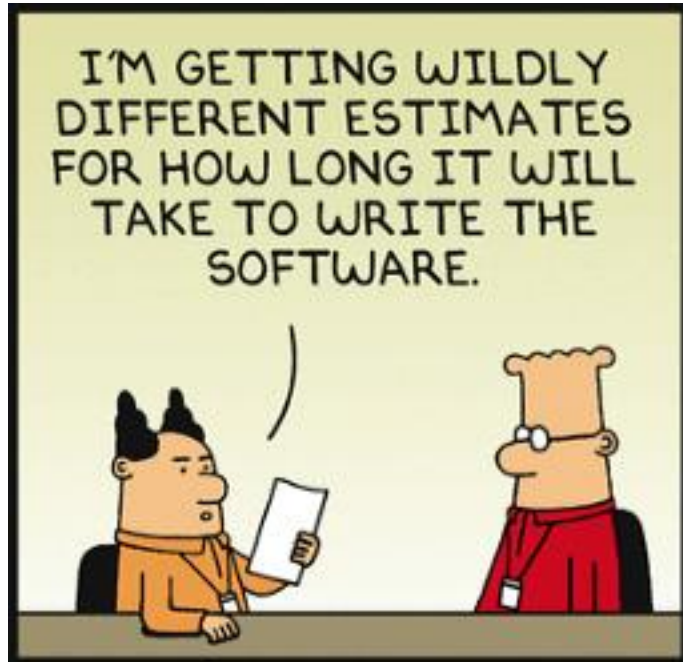
- Simple project: $PM = 2.4 * LOC^{1,05}$
- complicated project: $PM = 3.0 * LOC^{1,12}$
- complex project: $PM = 3.6 * LOC^{1,20}$

Intermediate Cocomo

Includes additional cost drivers as factors

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

$$PM = a * LOC^b * factor$$



Dilbert.com @ScottAdamsSays



10-20-16 © 2016 Scott Adams, Inc. /Dist. by Universal Uclick



Cost Estimation – Discussion

Team #NoEstimates

- Estimates are part of the software development process not because they're effective or beneficial, but because they've been a part of the process for so long that developers, managers, and team leaders assume they're a necessity.
- Estimates are always inaccurate and therefore pointless. When developers are asked to estimate the amount of time and work it will take to complete a project, they're asked to predict the future in a way that can't possibly account for the complex factors that will impact the estimate.
- Estimates are often padded by developers, encouraging mistrust between team members, managers, and clients. This can harm project transparency and even damage healthy working relationships.
- Estimates are a waste of valuable time. Instead of beginning the work, developers are forced to spend time and money creating estimates that aren't accurate to begin with and are therefore useless.

Team #Estimates

- Of course, estimates are always wrong—that's why they're called “estimates!”
- By refusing to provide estimates to management, developers remain insensitive to the complicated needs of the company that reach far beyond their project.
- Providing an estimate may impact multiple departments, teams, and individuals in ways the developer isn't privy to. Estimates should therefore be considered an important professional courtesy, even if developers themselves prefer to work without them.
- If you can't provide an estimate, you won't get funding—that's just how businesses work.
- If your estimates are inaccurate, why not improve them? Inaccuracy isn't a reason to abandon best practices.
- Businesses are built by predicting and analyzing market trends. Future forecasts are based on past performance. Estimates, therefore, can be created by assessing the time, cost, and units of work completed in past projects and by projecting them onto the current project.

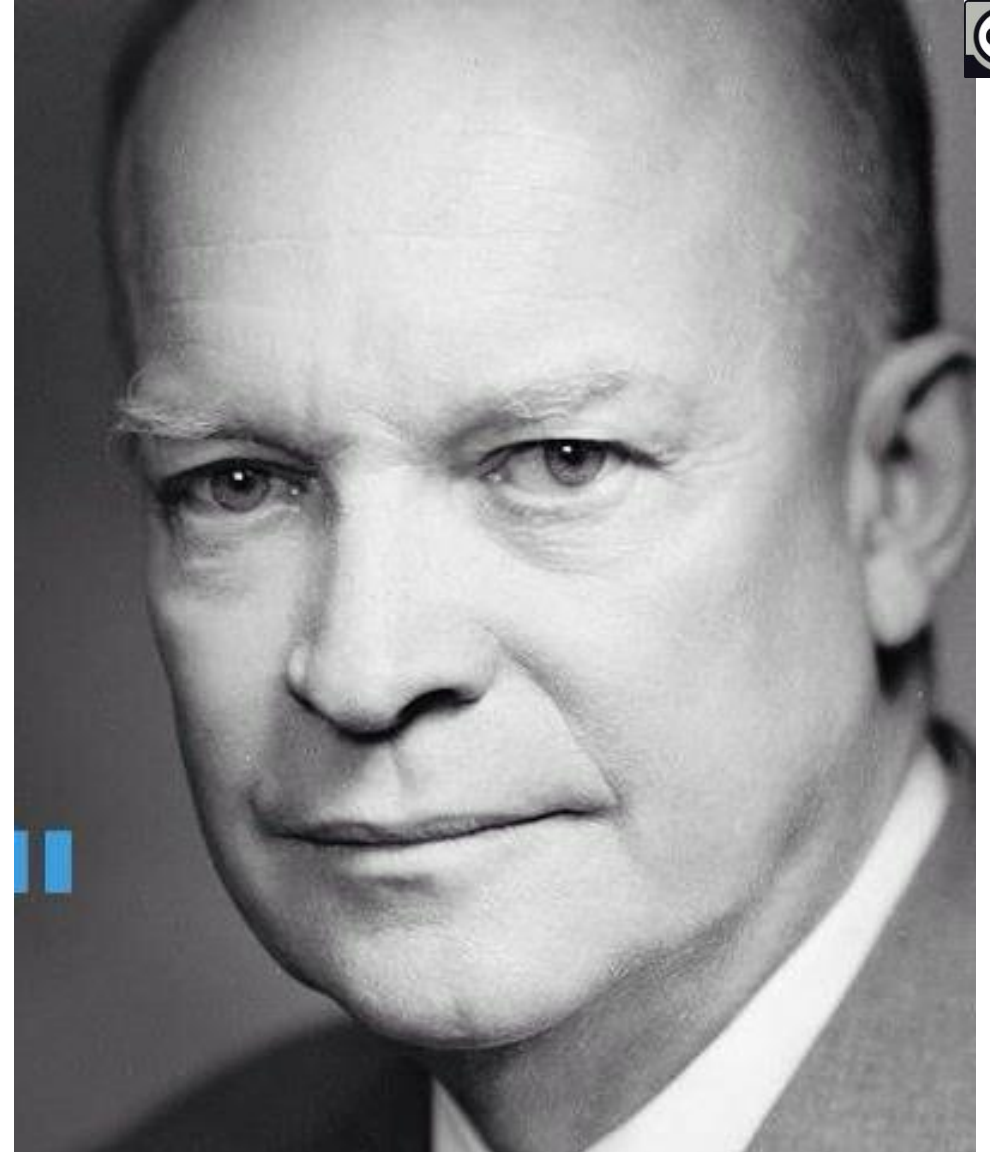


Project Planning and Scheduling



Antoine de Saint-Exupéry (1900–1944)

“A goal without a plan is just a wish.”



Dwight D. Eisenhower (1890–1969)

“Plans are nothing; planning is everything”

Project Planning

At the Proposal Stage

- when bidding for a contract
- enough resources?
- price for the bidding?
- not all requirements known
→ inevitably speculative

On Project Start

- who will work on the project?
- how to split into increments?
- refine initial estimates

Throughout the Project

- update plan based on new insights
- learn about the software and team capabilities
- estimates get more accurate

Software Pricing

- effort costs (software engineers / managers)
- hardware and software costs (incl. hardware maintenance and software support)
- travel and training costs
- price = estimated costs + profit + contingency (extra effort, 30-50%)

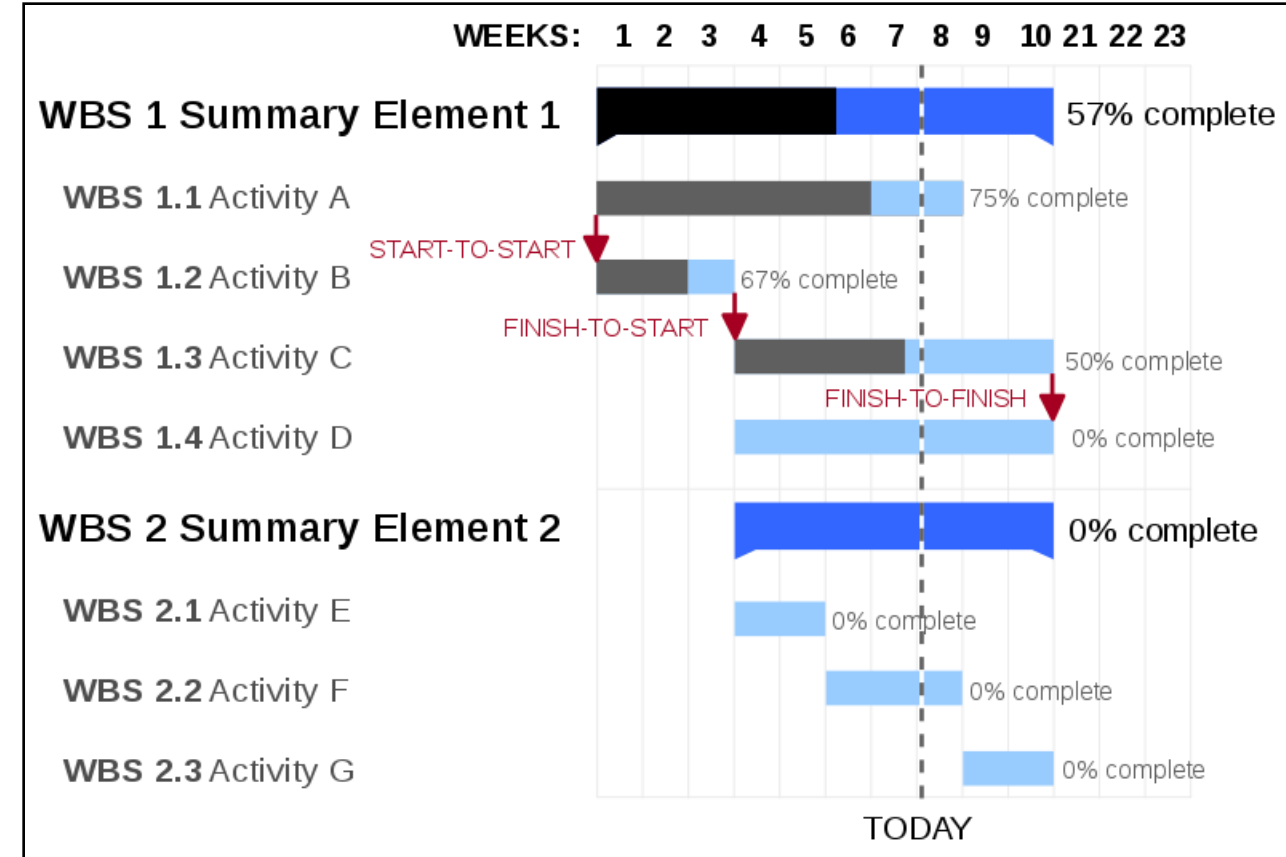
How relevant is pricing for SW really?

SW offers way more opportunities to make generate revenue than traditional products (see business models for digital products)

Gantt Chart

Gantt Chart

- named after Henry L. Gantt (1861–1919)
- bar chart with timeline on x axis and activities on the y axis
- *optional*: progress bars and marker for observation date
- *optional*: dependencies between tasks
- *optional, not shown*: highlight dependencies on the critical path
- **critical path**: tasks whose delay also delays the project



Network Diagram

Network Diagram (Netzplan)

- aka. PERT charts
- directed, acyclic graph
- nodes represent tasks
- edges represent dependencies

Metra Potential Method

Given project start date and **duration** of each activity we can compute:

- **earliest start** and **earliest finish** time with **forward pass**
- **latest start** and **latest finish** time with **backwards pass**
- **buffer** (time span between earliest and latest start/finish)

Example Network for a Bachelor's Thesis

0	3	3
Background		
0	0	3



3	4	7
Concept		
3	0	7



7	4	11
Evaluation		
7	0	11



earliest start	duration	earliest finish
Task		
latest start	buffer	latest finish

0	1	1
Introduction		
10	10	11



11	1	12
Summary		
11	0	12



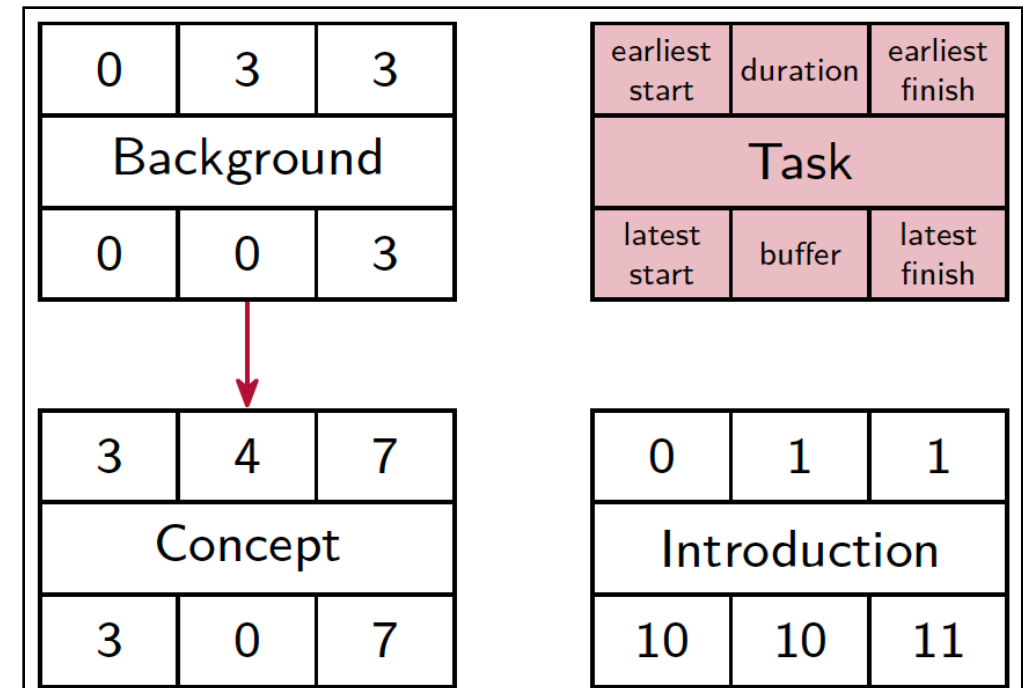
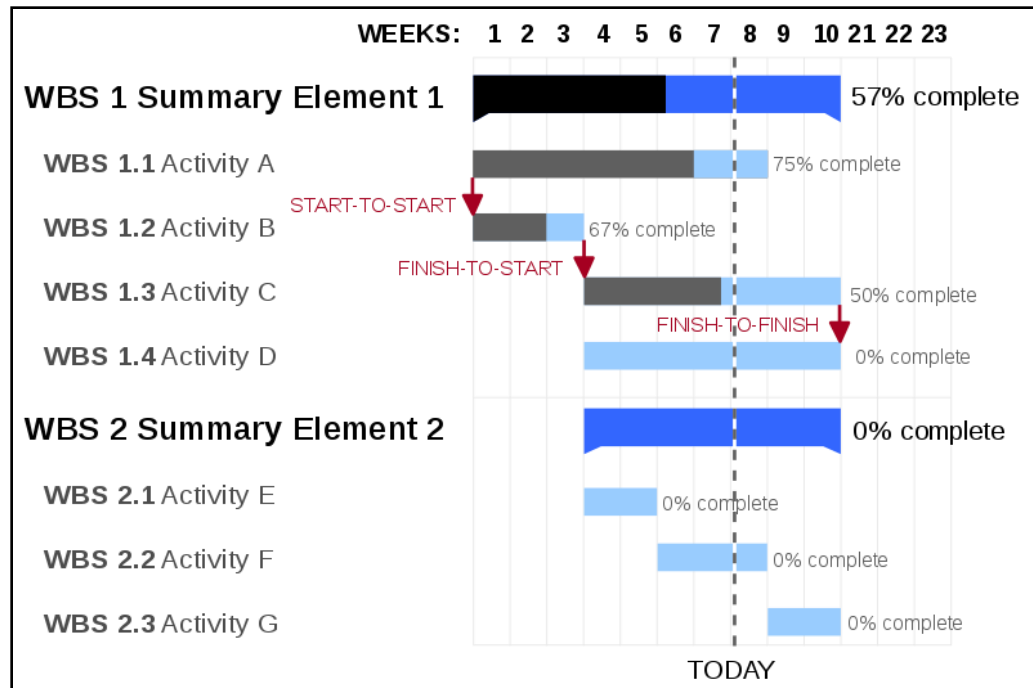
Gant Chart vs. Network Diagram

Gant Chart

- Very common technique
- Many tools available
- Great visualization of timing and progress

Network Diagram (Netzplan)

- Clear visualization of dependencies
- Explicitly includes buffer times (cf. metra potential method)





Process Metrics

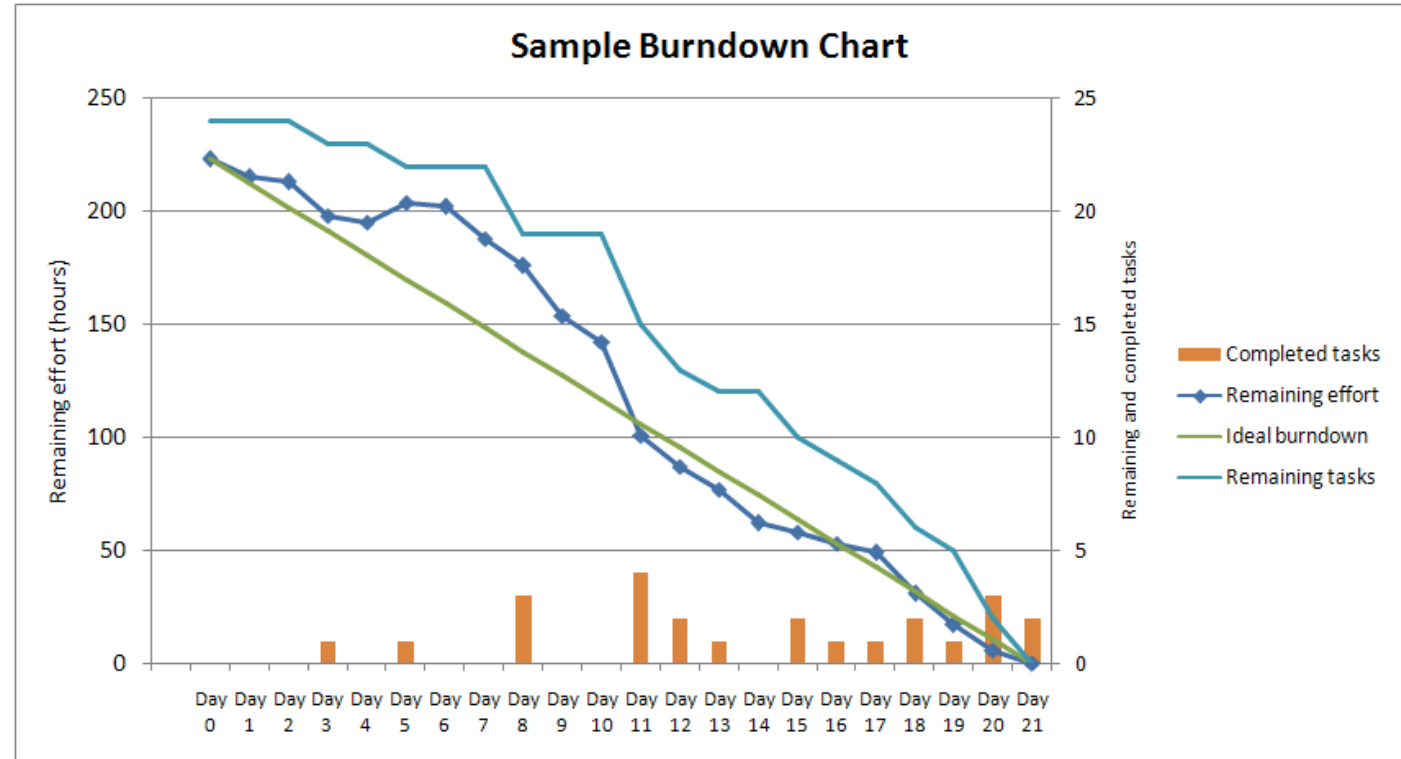
Burn-Down Chart

Burn-Down Chart

- Graphical representation of work left to do versus time.
- Vertical axis: Remaining work
Horizontal axis: remaining time
- Needs regular and continuous update of remaining effort
- Useful for predicting when all work will be completed.
- Can be applied to any project containing measurable progress over time

Interpretation

- Days 4 – 10: Team is behind schedule
- Days 11 – 21: Team is on track
- Day 21: Team has completed all tasks on time



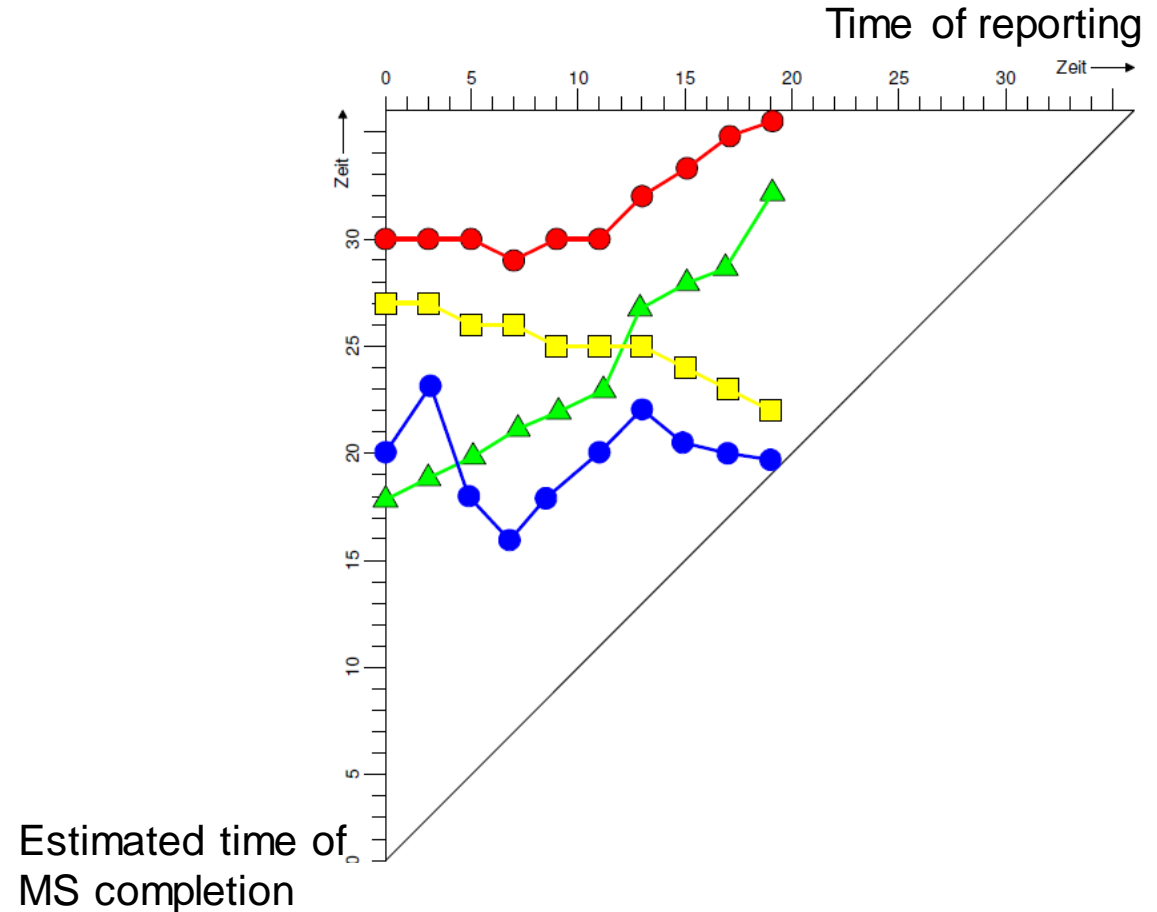
Milestone Trend Analysis

Milestone Trend Analysis

- Graphical representation of changes in the planning of milestone completion dates
- Prerequisites: Defined milestones with estimated times of completion
- Updates of milestone completion dates need to be recorded regularly

Interpretation

- Strongly increasing (green line): Estimation was too optimistic
- Decreasing (yellow line): Too much safety buffer
- Delayed change in trend (red line): Unreliable early estimations
- Zick-zack pattern (blue line): unreliable estimations



Goodhart's law

"When a measure becomes a target,
it ceases to be a good measure."

Summary

SW Management

- A SW development project is a temporary activity with fixed start and end date
- SW project management is concerned with delivering accepted SW on time, within budget, and with a well-functioning team
- SW project management includes risk management, people management, project planning, reporting, and proposal writing
- Project planning is necessary, but it is even more important to monitor and update the project plan during the project
- Effort estimation is hard