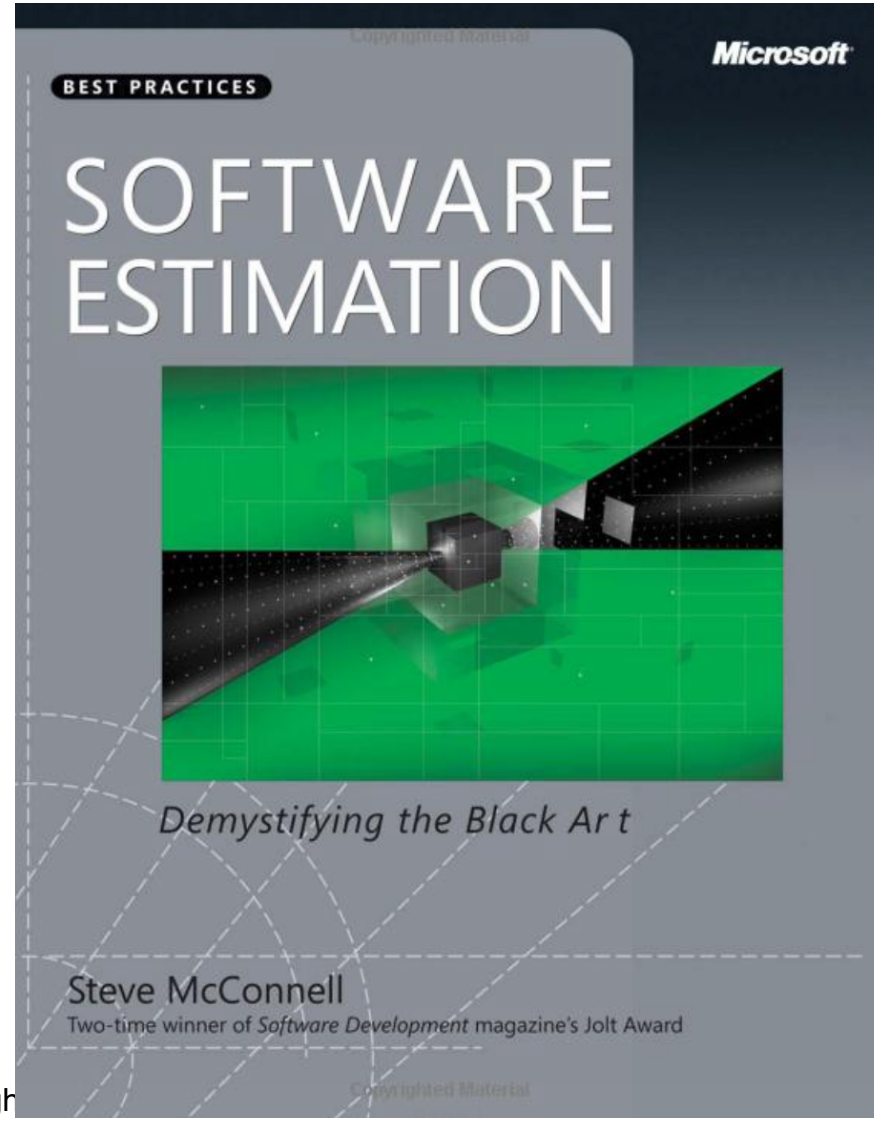
CS445 / ECE451 / CS645 / SE463

Software Requirements Specification & Analysis

Software Estimation

Sources



Lecture content comes from

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

Why Estimate Software Cost/Effort?

- To assess economic feasibility
- To understand resource needs
- To provide a basis for agreeing to a job
- To make commitments that we can meet

Terminology

An estimate is a prediction of how long a project will take or how much it will cost.

A target is a statement of a desirable business objective.

A commitment is a promise to deliver.

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

Software Estimation

How long (hours) would it take you to implement a Stack ADT (push, pop, top)?

How long would it take you to re-implement your OS project?

How long would it take you to implement your CS 445 / ECE 451 / SE 463 / CS 645 project?

Inaccurate Estimates

"An estimate is the most optimistic prediction that has a non-zero probability of coming true.

Accepting this definition leads irrevocably toward a method called what's-the-earliest-date-by-which-you-can't-prove-you-won't-be-finished estimating."

Tom DeMarco

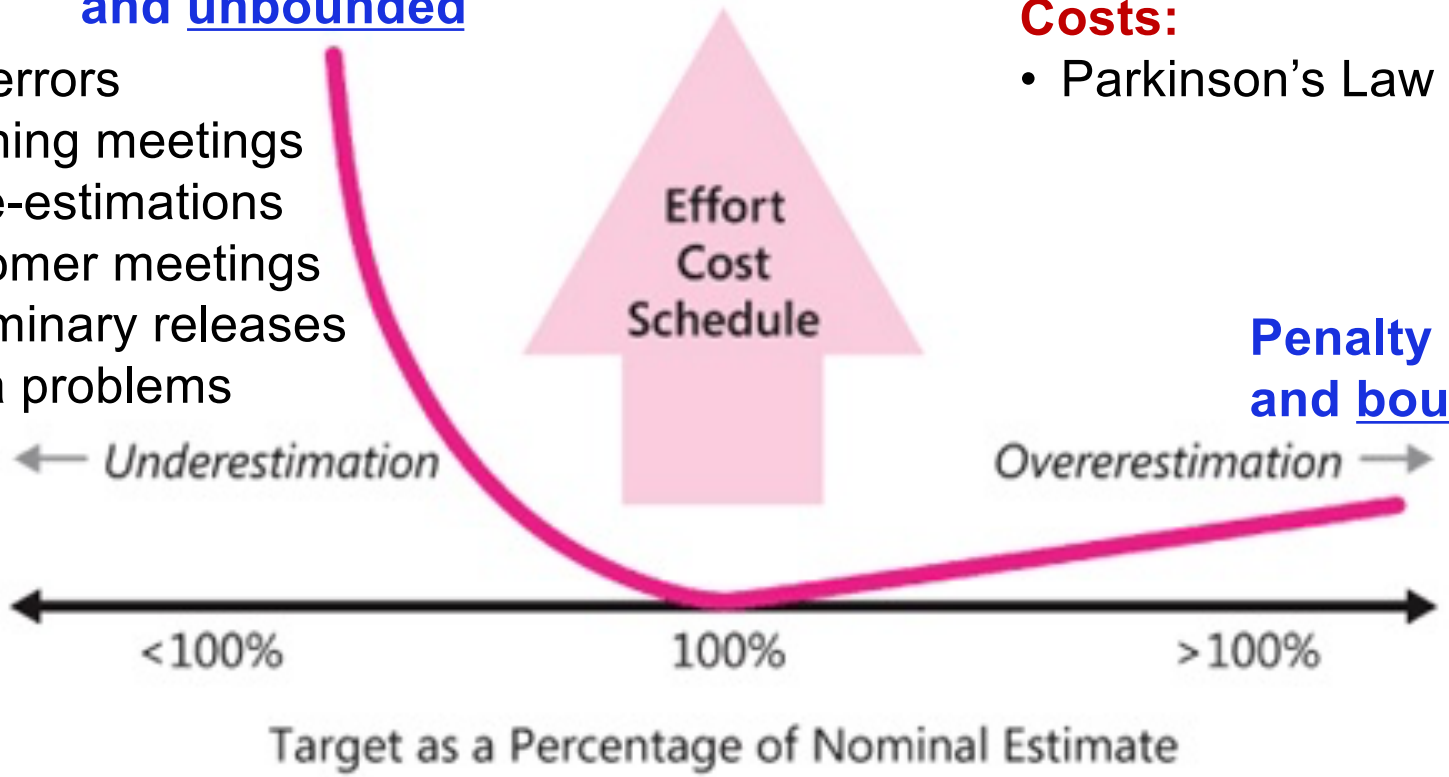
Pricing to win means bidding as low as possible to beat the competition and win the contract.

Effects of Inaccurate Estimates

Penalty is nonlinear and unbounded

Costs:
- upstream errors
- extra planning meetings
- frequent re-estimations
- extra customer meetings
- extra preliminary releases
- fixing extra problems

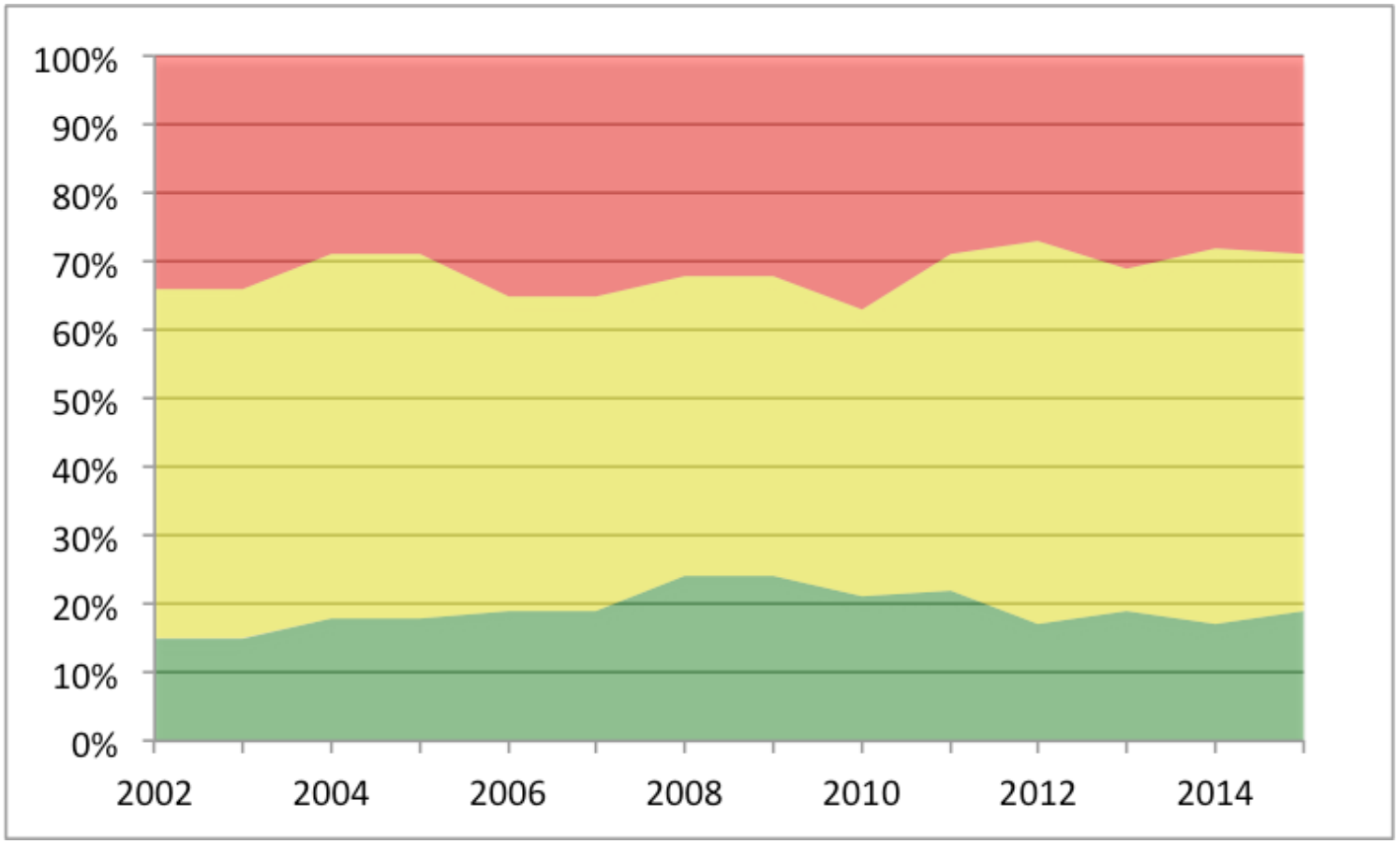
Costs:
- Parkinson's Law



Penalty is linear and bounded

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006. (Figure 3.1)

Industry's Track Record



The Standish Group, The CHAOS Report, 2002-2015

Sources of Estimation Error

Omitted Activities

- Omitted Functional / Quality Requirements
  - Initialization, data conversion, glue code
- Software Development Activities
  - Integration, test data, performance tuning, technical reviews
- Non-Software Development Activities
  - Vacations, sick days, training, company meetings

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

Sources of Estimation Error

Optimism

- Developer estimates tended to contain an optimism factor of 20% to 30%
- Managers believe that projects can be completed 30% faster than previous projects
  - Team more productive, less will go wrong, no learning curve

Bias

- Managers who want estimates to align with targets will put pressure on estimates, schedules, project teams.

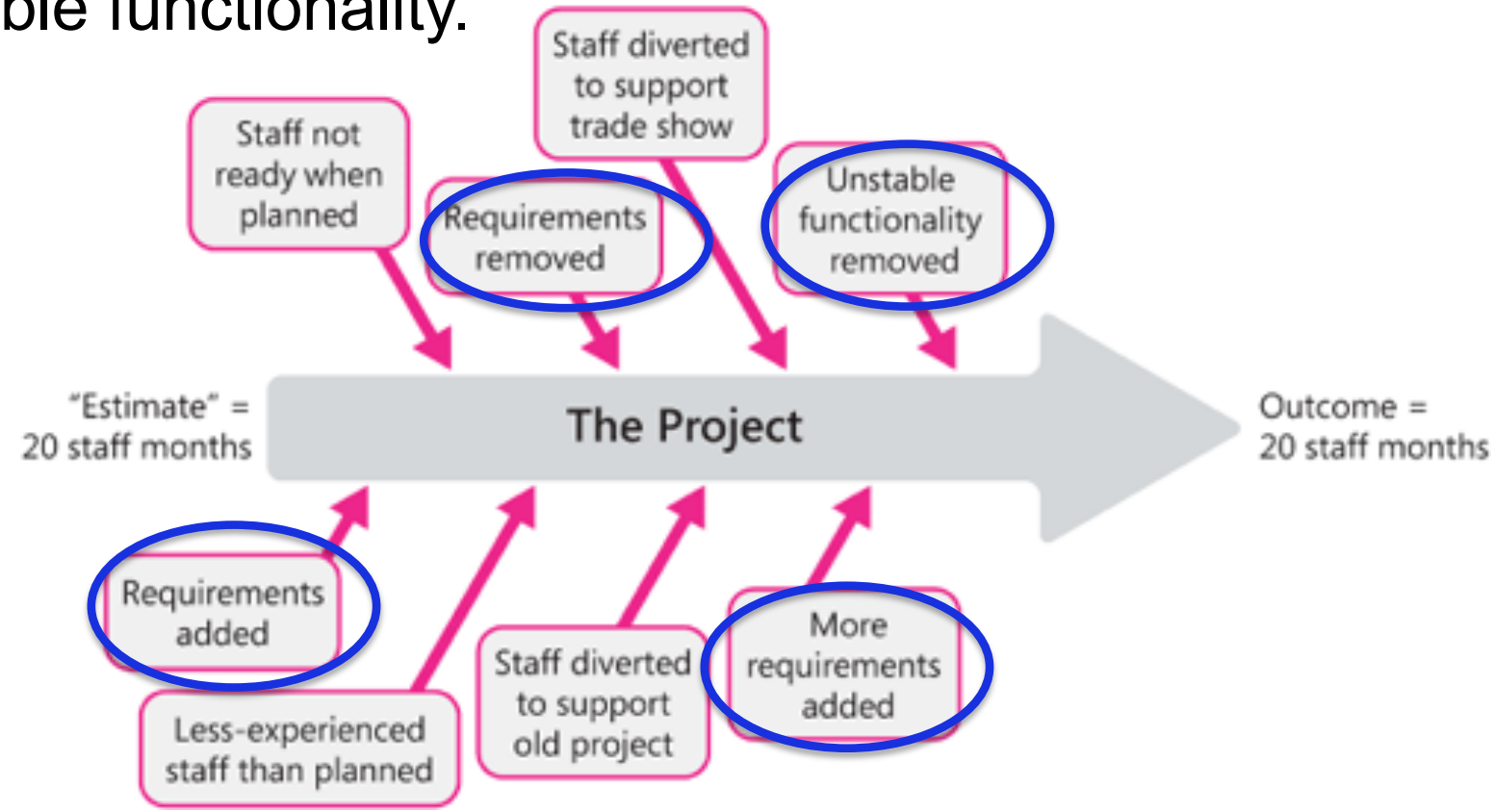
Subjectivity

- Some estimation techniques include multiple adjustment knobs that allow subjectivity to creep in.

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

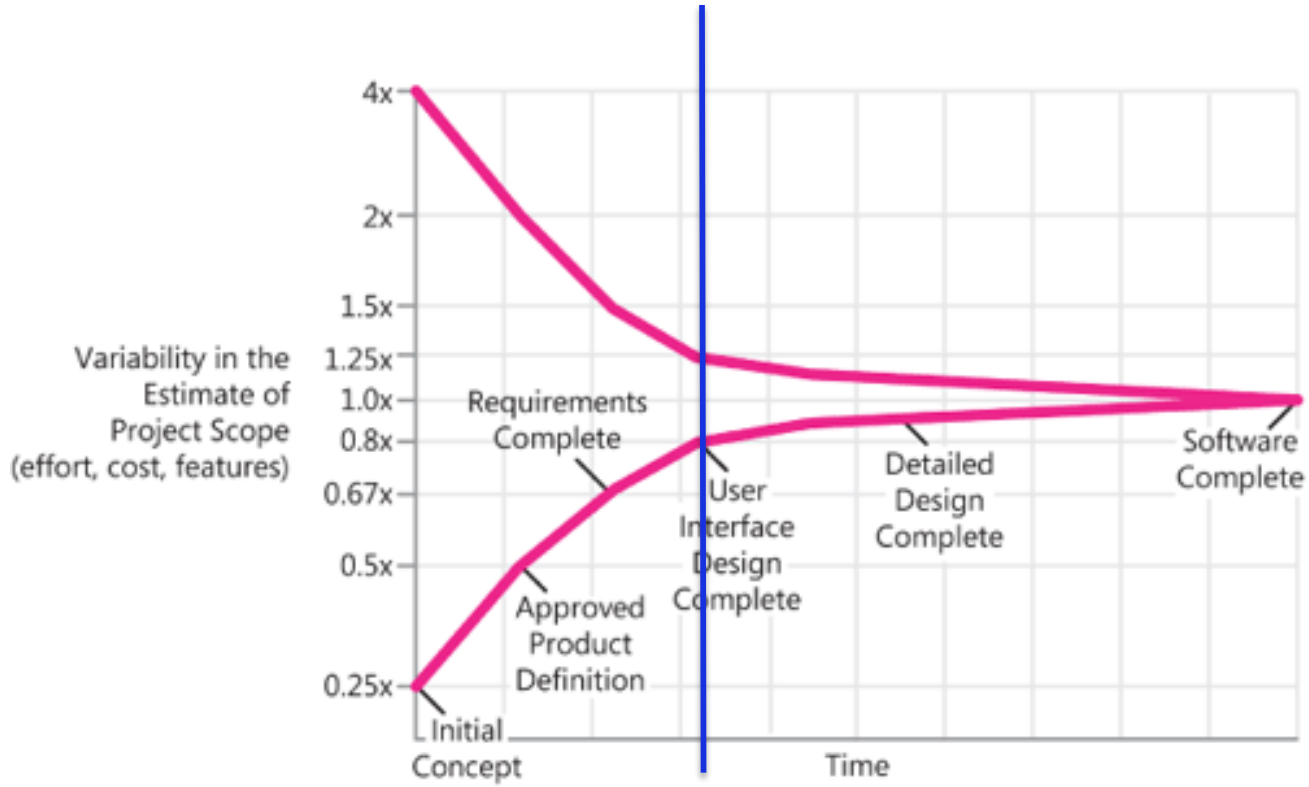
What is a Good Estimate?

A project is successful if it is delivered on time, within budget, with acceptable functionality.



Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006. (Figure 1.9)

Estimation Error



Uncertainty about how numerous requirements details and design decisions will be made declines over the course of the project.

Estimation accuracy improves rapidly for the first 30% of the project.

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006. (Figure 4.2)

Biggest Influences on Estimates

Project Size

- The largest driver in a software estimate is the size of the software being built
- Software projects have diseconomies of scale: larger projects require more coordination and communication

Kind of Software

- Business applications, embedded systems, real-time

Personnel Factors

- Productivity of individuals can vary by a factor of 10 or more
- Mostly accounted for if estimates are based on team's historical performance

Steve McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.

References

McConnell, S., *Software Estimation: Demystifying the Black Art,* 2006.