



Saxonia Systems

Wir lieben IT.

www.saxsys.de

Agile software development

FASTER, BETTER, CHEAPER

Michel Löhr

michel.loehr@saxsys.de



Dresden · Frankfurt/Main · Leipzig · München · Hamburg · Görlitz · Berlin





Agenda

→ Introduction	1
→ Shortcomings Traditional Software Development	2
→ EnergyBall Game	2
→ Introduction into Scrum	3
→ The Marshmallow Challenge	2
→ Agile Planning	2
→ Business Value Game	5





Saxonia Systems

Wir lieben IT.

www.saxsys.de

Experiences traditional projects

- Qimonda: MES development , > 20 milion €
- Otto: Logistics & order management system
 - 1st generation: ~ 150 MY
 - 2st generation Java: EOL
 - 3st generation SAP: ???





Standish Reports

→ Standish Group „Chaos Reports“

→ 1994 – 82% challenged or failures

→ 2001 – 72% challenged or failures

→ 2009 – 68% challenged or failures

→ Definition of project „success“

→ Successful: on time, on budget, all specified features

→ Challenged: completed and operational, but over budget, late, and with fewer features than initially specified

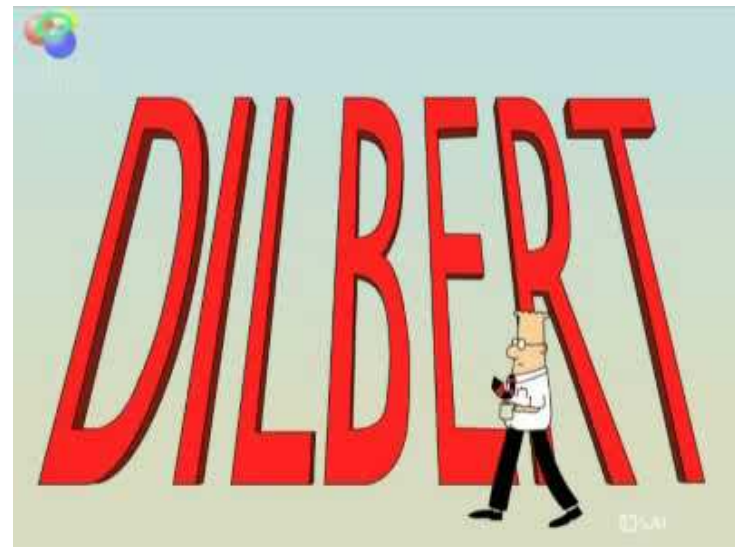
→ Failed: canceled before completion or never implemented





Standish Reports

- The Standish data are NOT a good indicator of poor software development performance
- However, they ARE an indicator of systematic failure of our planning and measurement processes.

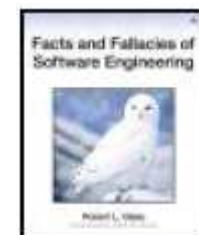




Biggest challenge in SW development: complexity

- „For every 25 percent increase in problem complexity, there is a 100 percent increase in complexity of the software solution. That's not a condition to try to change, that's just the way it is.”
- “Eighty percent of software work is intellectual. A fair amount of it is creative. Little of it is clerical.”

**Facts and Fallacies of
Software Engineering**
Robert L. Glass





Complexity I

→ Why are people so important?

- Because it takes considerable intelligence and skill to overcome complexity.

→ Why is estimation so difficult?

- Because our solutions are so much more complicated than our problems appear to be.

→ Why is reuse-in-the-large unsuccessful?

- Because complexity magnifies diversity.

→ Why is there a requirements explosion

- Because we are moving from the 25 percent part of the world to the 100 percent part.

→ Why does software have so many errors?

- Because it is so difficult to get it right the first time.





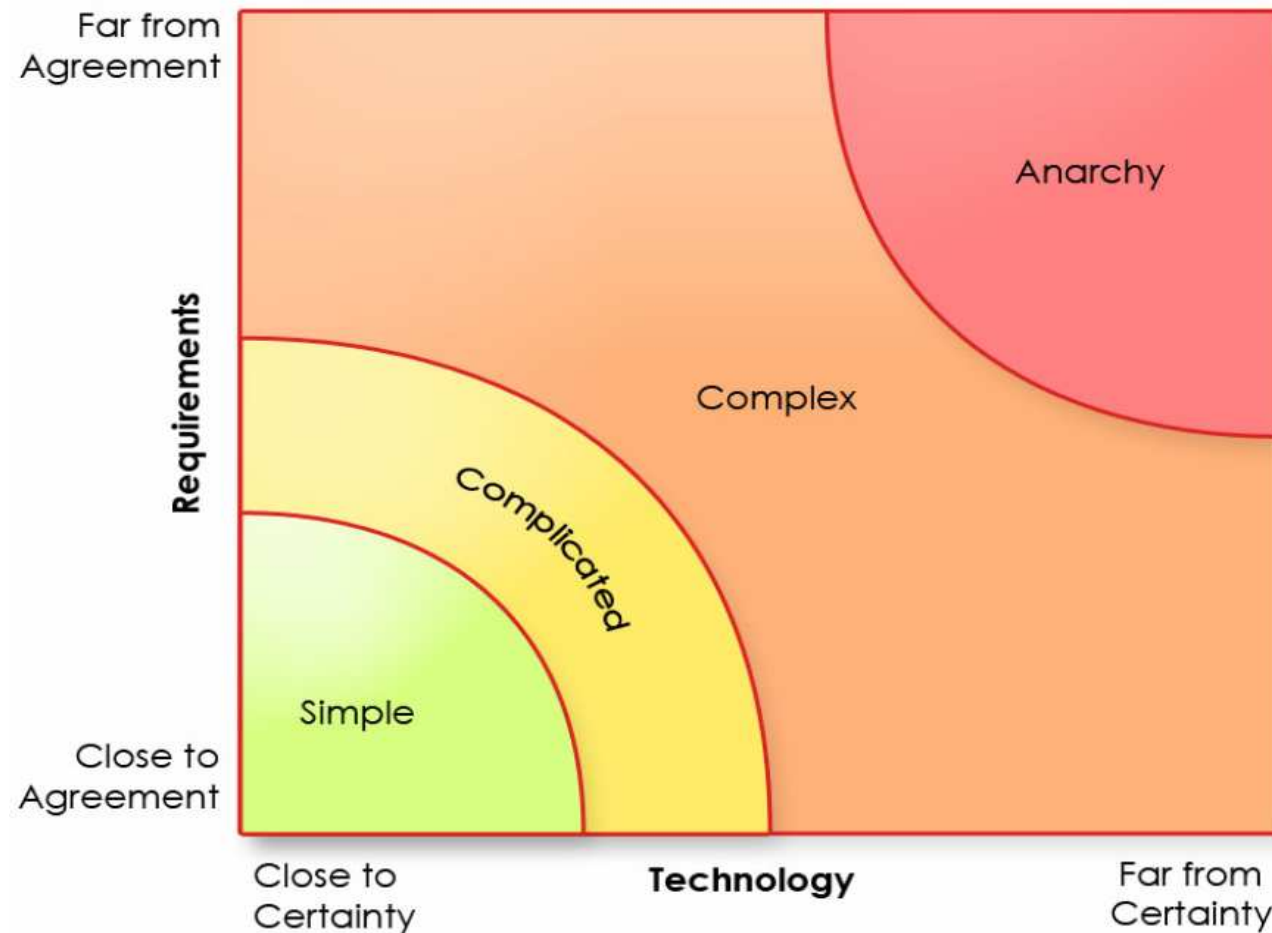
Complexity II

- Why are there so many different correct approaches to designing the solution to a problem?
 - Because the solution space is so complex.
- Why do the best designers use iterative, heuristic approaches?
 - Because there are seldom any simple and obvious design solutions.
- Why is 100 percent test coverage rarely possible and, in any case, insufficient?
 - Because of the enormous number of paths in most programs and because software complexity leads to errors that coverage cannot trap.
- Why are inspections the most effective and efficient error removal approach?
 - Because it takes a human to filter through all that complexity to spot errors.



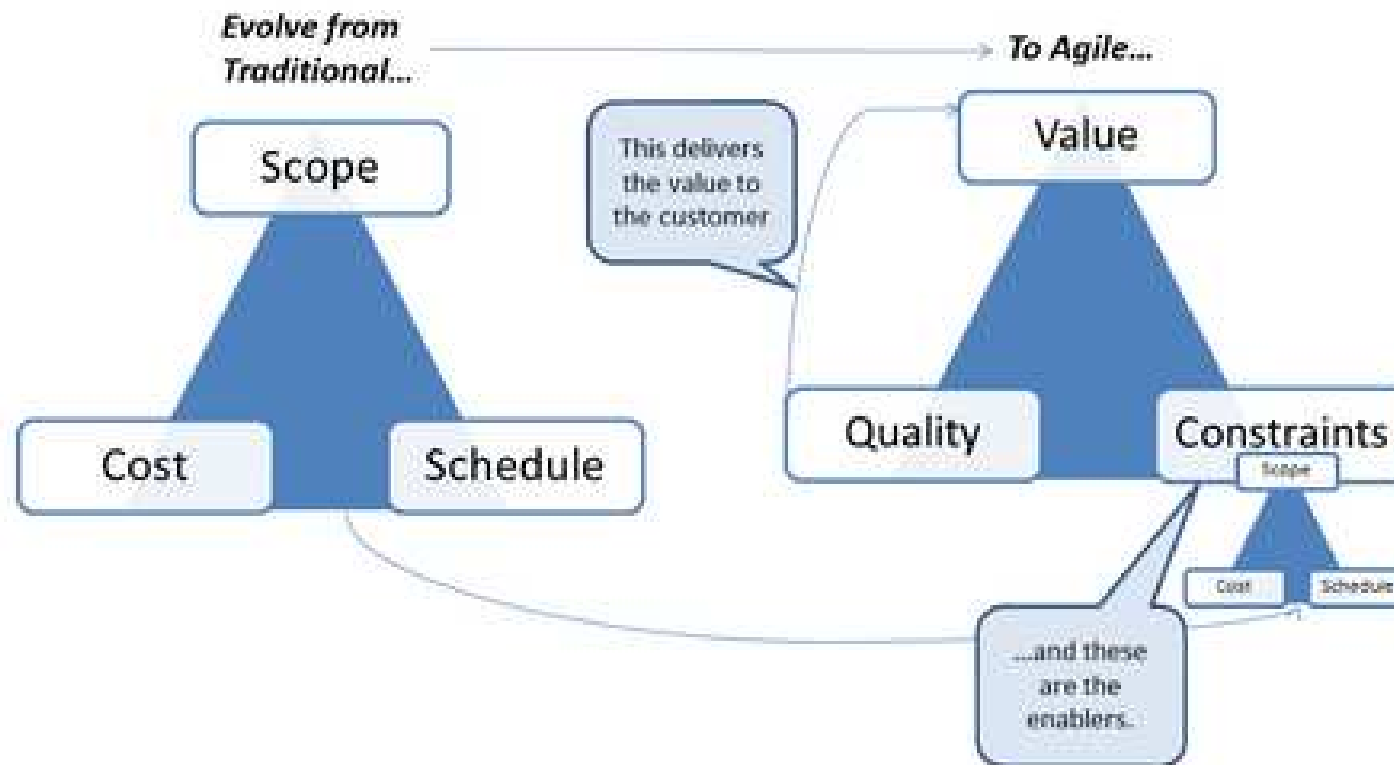


The Spectrum of Process Complexity



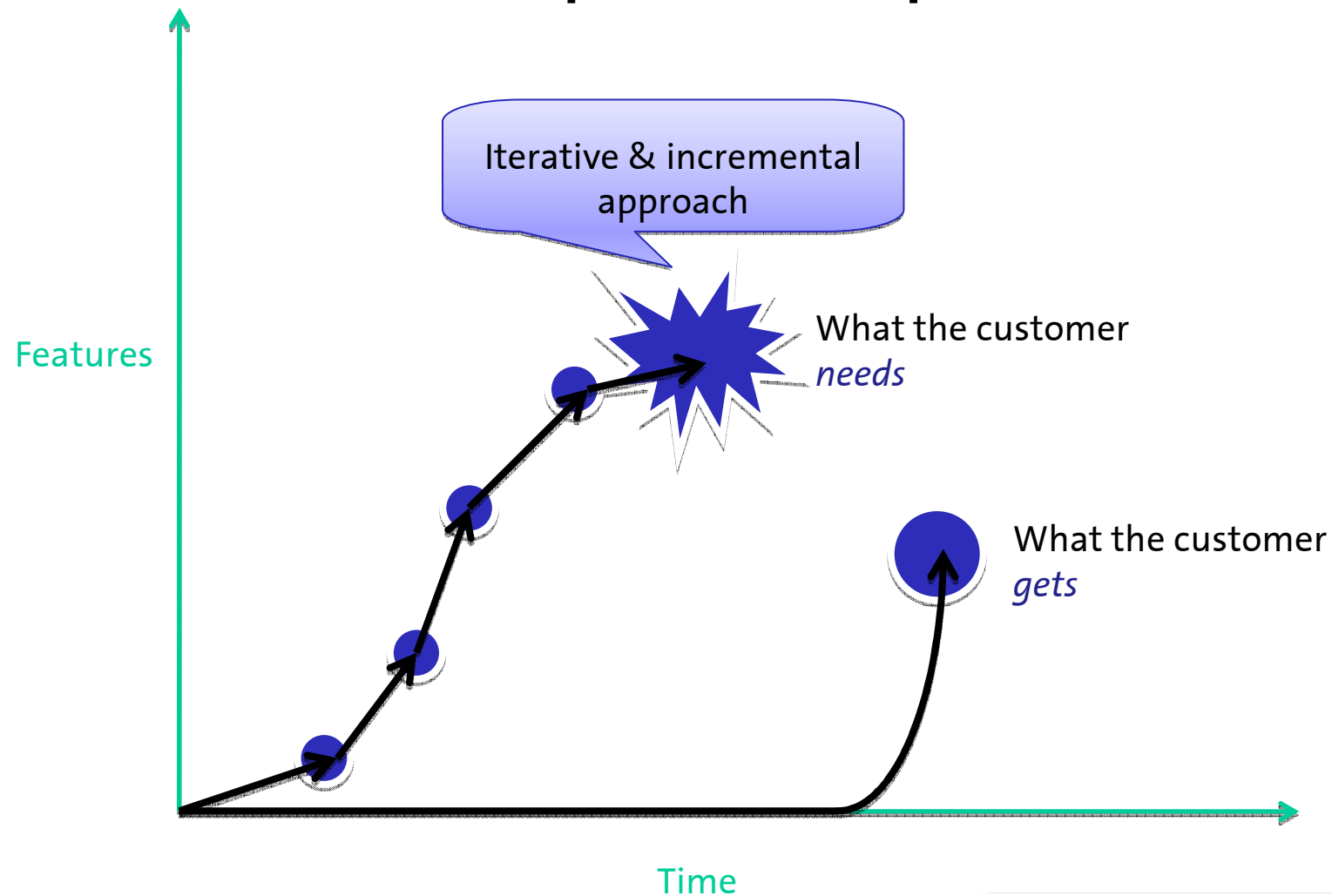


Project management





Requirements Gap





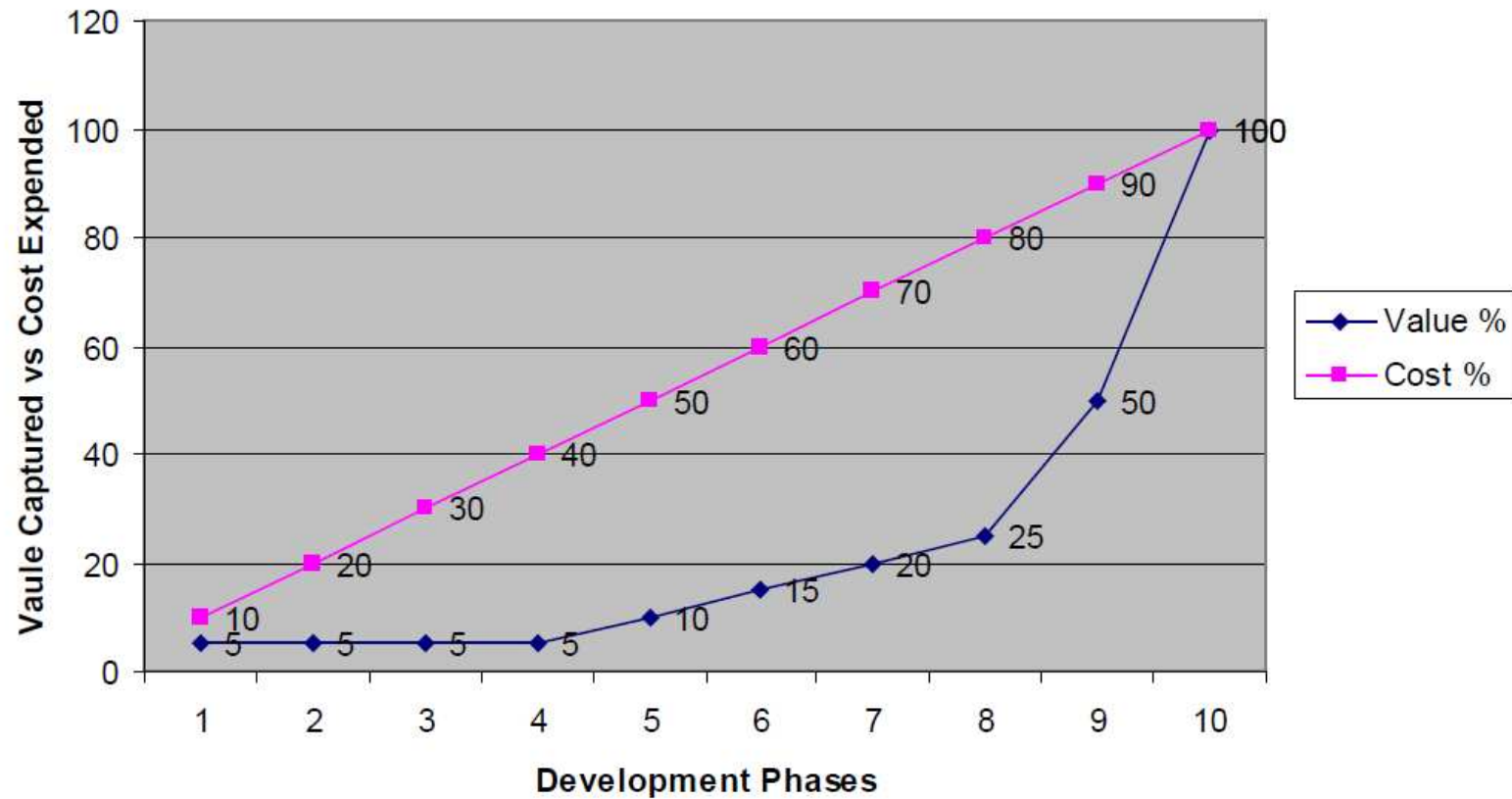
Saxonia Systems

Wir lieben IT.

www.saxsys.de

Value

Value Cost Ratio Curve (Traditional)

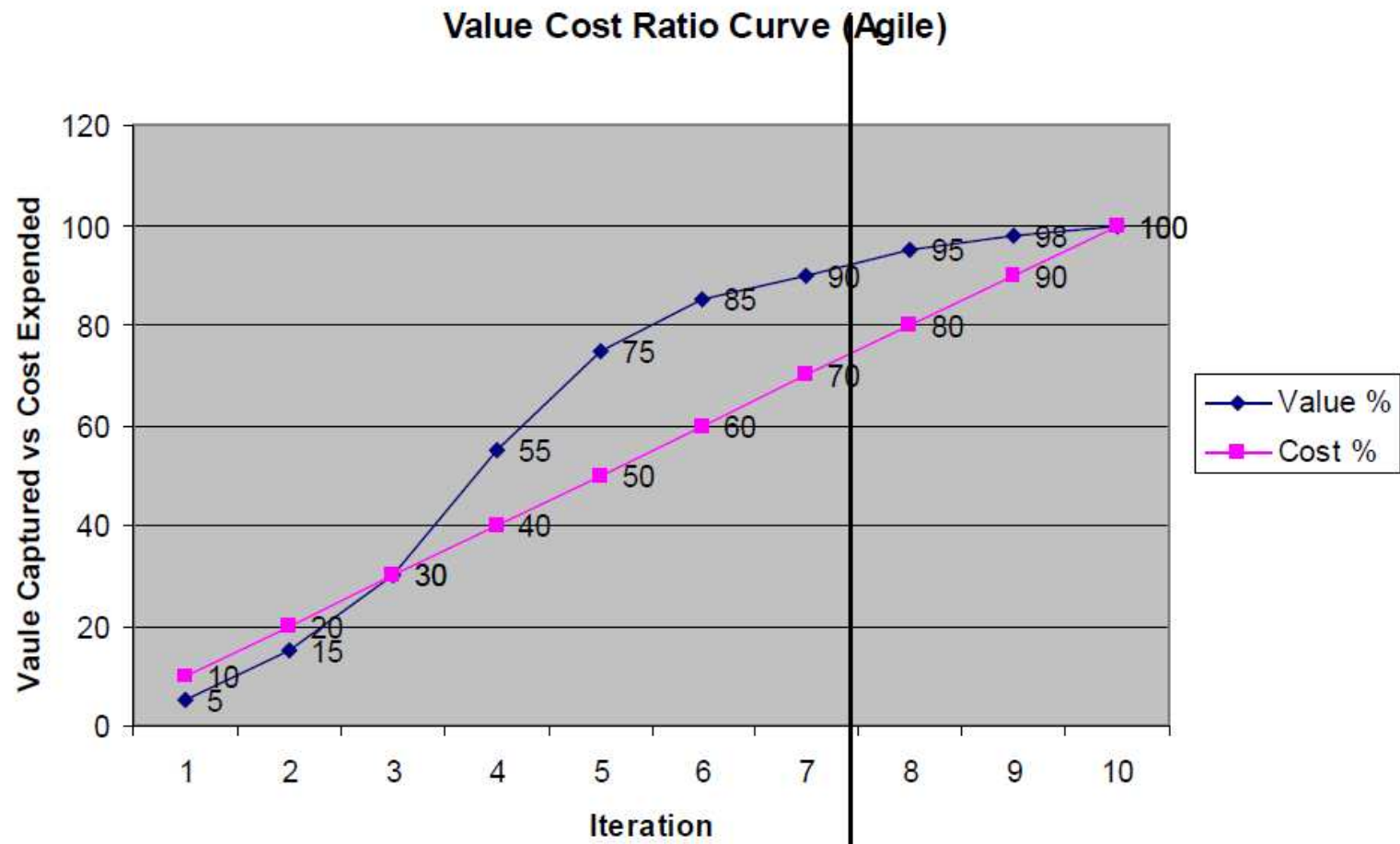


©2010 Jim Highsmith





Value





Saxonia Systems

Wir lieben IT.

www.saxsys.de

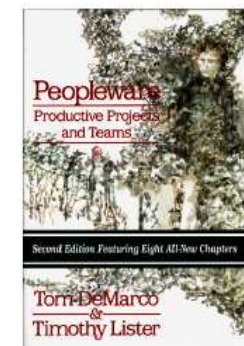
Quality

- “People under time pressure don't work better; they Just work faster. In order to work faster, they may have to sacrifice the quality of the product and their own job satisfaction.”
- “Quality, far beyond that required by the end user, is a means to higher productivity.”

PeopleWare

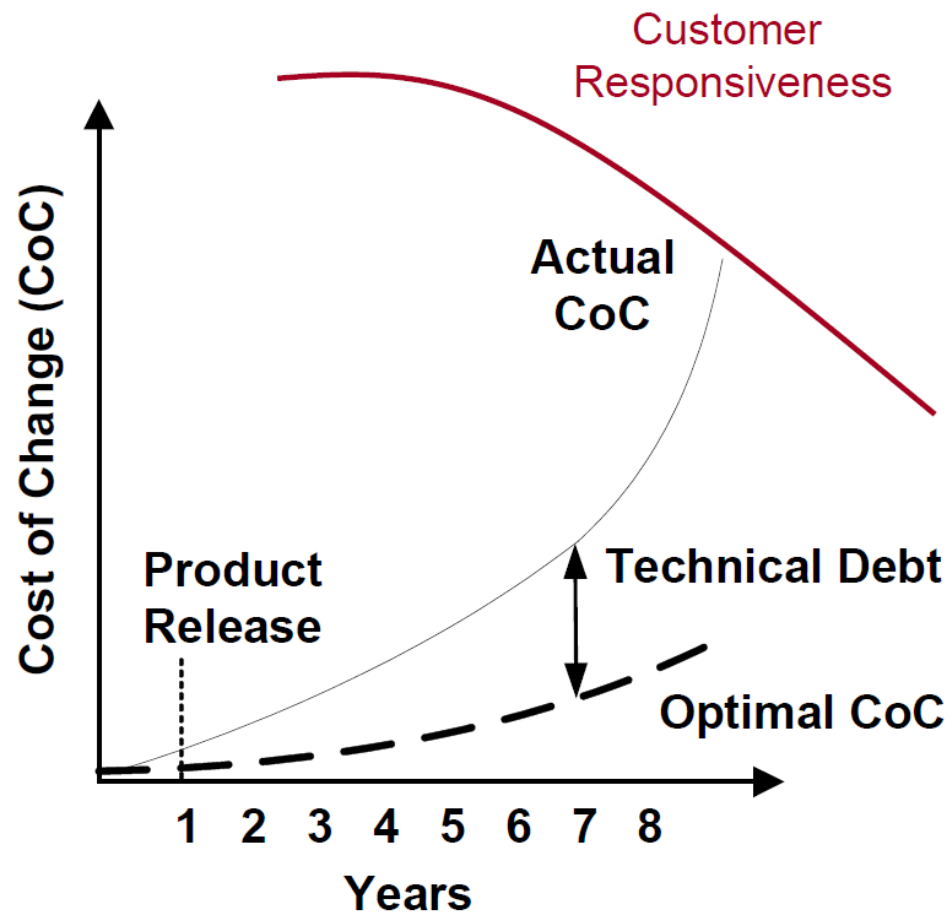
Productive Projects and Teams

*Tom DeMarco
Timothy Lister*





Technical Debt



©2010 Jim Highsmith

- Once on far right of curve, all choices are hard
- If nothing is done, it just gets worse
- In applications with high technical debt, estimating is nearly impossible
- Only 3 strategies:
 - Do nothing, it gets worse
 - Replace, high cost/risk
 - Incremental refactoring





Impact of Code Quality on Testing

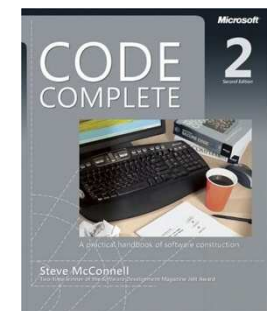
Given: Team of 4 people having 10 days producing 4 KLOC

Assume: 1/2 day to find & fix per defect.

Defects/KLOC	Testing Time
1	2 days
15	30 days

* Industry Average: "about 15 - 50 errors per 1000 lines of delivered code."

CODE COMPLETE
Steve McConnell





Saxonia Systems

Wir lieben IT.

www.saxsys.de

Some companies using Scrum



WEB.DE

IMMOBILIEN

SCOUT 24

immonet.de
schneller besser wohnen

XING >

PAYBACK



mobile.de

AUTO

SCOUT 24

