

10 Myths of Rapid Software Development

Steve McConnell

Construx

Delivering Software Project Success

© 2000-2002. All Rights Reserved.

Myth #1

**Rapid Development
is a New Issue**



Rapid Development Has Been an Issue for 30 Years!

- ❖ **Gene Bylinsky (1967)**
“All significant programming problems turn out to be emergencies”
- ❖ **Fred Brooks (Mythical Man-Month, 1975):**
“More software projects have gone awry for lack of calendar time than all other causes combined.”

3



Why Do We Need Rapid Development?

Let's walk our way through a typical project:

1. Developers estimate the schedule
2. To please the customer, management cuts the schedule by 50%
3. Team abbreviates requirements because there isn't enough time for it
4. Team abbreviates design because there isn't enough time for it
5. Team begins coding and testing

4



Why Do We Need Rapid Development? (cont.)

- 6. Team adds features that were missed when requirements work was abbreviated**
The project gets later
- 7. Team codes “quick and dirty” solutions to design problems**
→ The project gets later
- 8. Team fixes the bugs created by the quick and dirty workarounds**
→ The project gets later

5



Why Do We Need Rapid Development? (cont.)

- After many more of these cycles...**
- 9. Team releases the software over budget and with less functionality than desired ...**
... close to the time developers originally estimated in Step 1!

6

Myth #2

Productivity Is About the Same at Every Company



Productivity Varies Greatly

- ❖ **20:1 variations in productivity between different programmers**
- ❖ **10:1 variations in productivity between different companies working in the same industries**
- ❖ **Organizations that focus on software best practices report improvements in their own productivity of 2-5x over time**

Myth #3

**Working Hard Promotes
Rapid Development**



Old Saying

***“It’s better to work smart
than to work hard”***



**Microsoft Changed the Old
Saying to...**

***“It’s better to work smart
and hard”***

11



**Amazon.com Changed the
Saying to...**

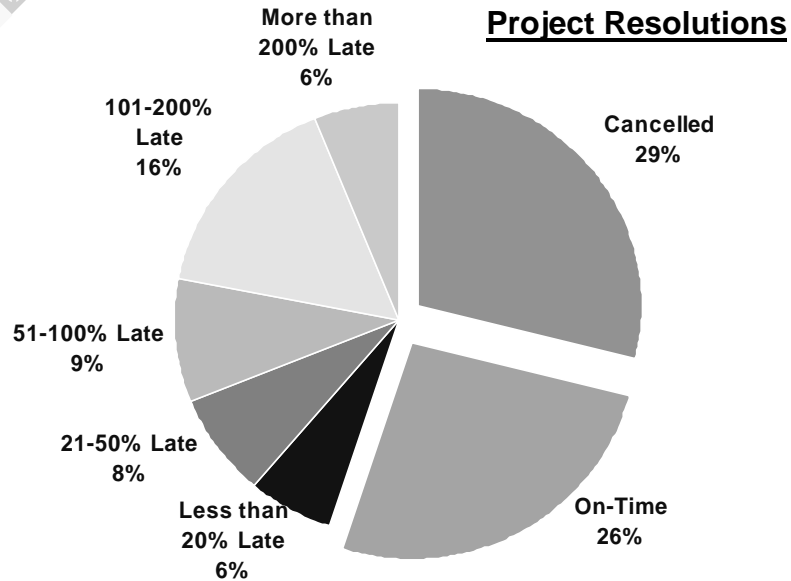
***“It’s better to work smart
and hard and long!”***

- ❖ **Should these expectations apply to your company?**
- ❖ **Should these expectations apply to anyone?**

12



Reality Check:



13



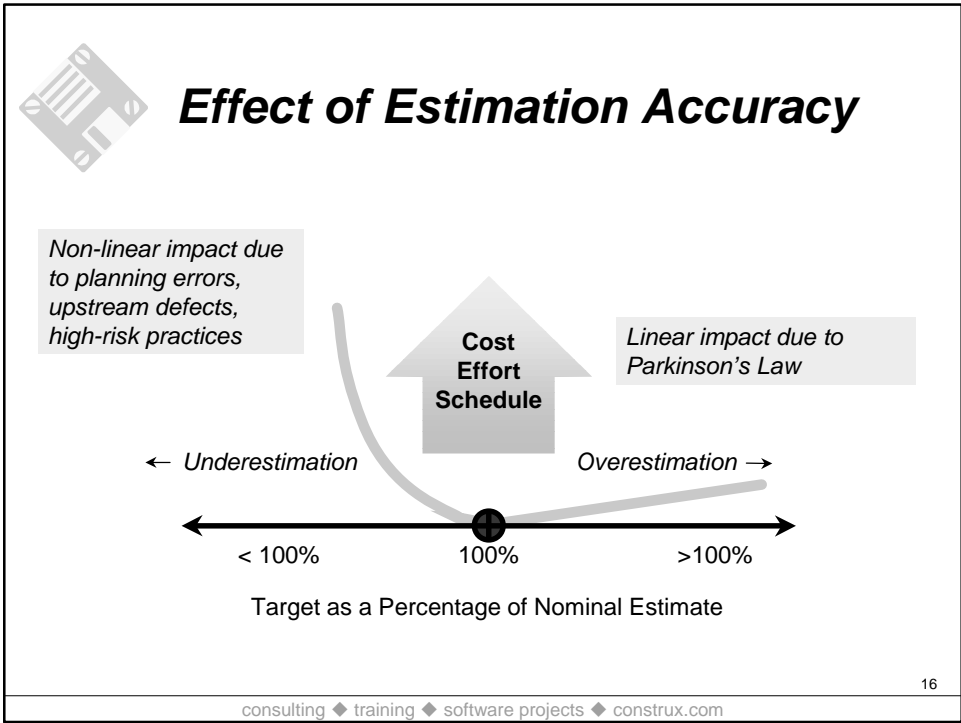
Bottom Line

- ❖ Working hard *and* smart usually means working *dumb!*
- ❖ The average project spends 40-80% of its budget on unplanned rework (defect corrections)--that is working dumb
- ❖ “Work smart, not hard” turns out to be the right idea after all

14

Myth #4

Short Estimates Produce Short Projects





Short Estimates Increase Cost and Schedule

- ❖ **Short estimates lead to planning mistakes and quality mistakes that make projects take *longer***
- ❖ **Underestimation is a common, severe problem**
- ❖ **Very little estimation is actually done-- mostly target setting**
- ❖ **Part of achieving true rapid development is making software estimates rational**

17



Improved Estimation



From a set of U.S. Air Force projects

18



Improved Estimation

- ❖ Estimates must be made accurate before rapid development is achieved
- ❖ This typically means *increasing* estimates (common practice is under-estimation)
- ❖ This is hard because people want to believe they will develop software faster than they usually do

Myth #5

Productivity Can Be Improved Tactically, on a Project-by-Project Basis



Popular Variations on This Myth:

“We won’t make as many mistakes on this project as we made on our last project.”

“Fewer things will go wrong on this project than our last project.”

“We’ll be a lot more productive this time than we were last time.”

21



Improving Productivity is Primarily an Organizational Strategy

- ❖ **Productivity is affected by organizational influences that are hard to change on a project-by-project basis**
 - ◆ **How many projects do developers have to work on at the same time?**
 - ◆ **How many customer demos & trade shows do developers have to support?**
 - ◆ **Can management remove a problem programmers from the team?**
 - ◆ **Are developers allowed to talk to the customers when they need to?**
 - ◆ **Etc...**

22



Unlike Mutual Fund Disclosures...

- ❖ In software, past performance is your *best* indicator of future performance
- ❖ We weren't stupid yesterday
- ❖ We aren't 500% smarter today
- ❖ Assume no more than 10% improvement in productivity from one project to the next considering all factors combined!

23



What is Rapid Development?

Rapid development can be project-level or organization-level

- ❖ **Project Level:** Project plans are tilted toward schedule performance (tactical, local optimization, based on tradeoffs, not productivity)
- ❖ **Organization Level:** Overall software development effectiveness is improved (raise baseline productivity--very powerful strategic, global optimization)

24

Myth #6

**You Can Trade-Off Quality
for Schedule or Cost**

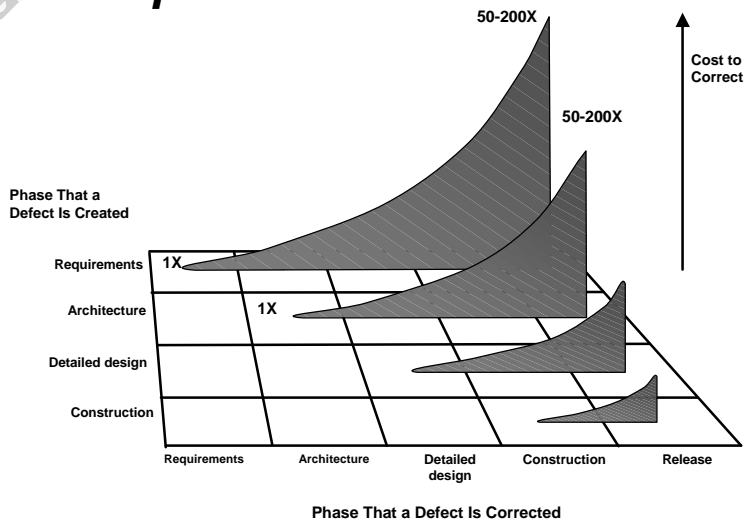


Cost of Quality

- ❖ **For most projects, unplanned defect correction work is the largest cost driver (40-80% of total cost)**



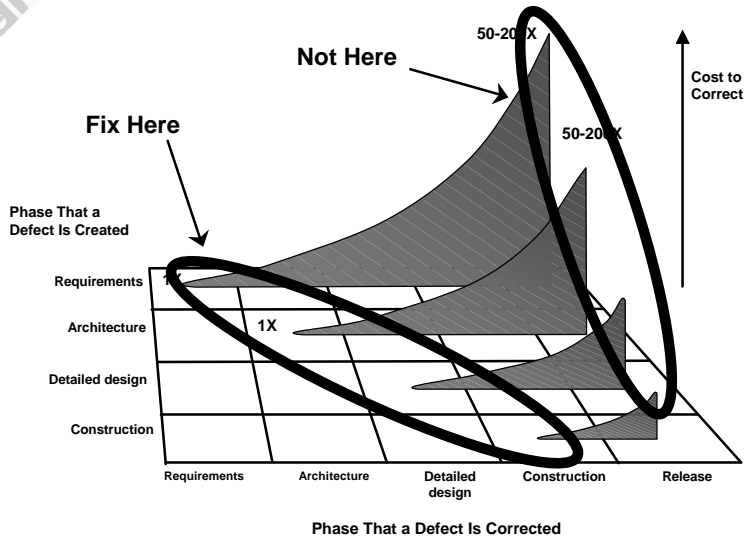
Late Defect Correction is Expensive



27



Fix More Defects Earlier!



28



Cost of Quality

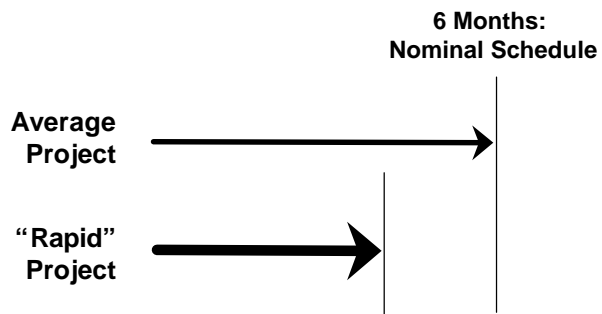
- ❖ **When organizations focus on quality, they typically improve in all categories at once**
- ❖ **Typical results (13 company study):**
 - ◆ **Duration: 3.5 years**
 - ◆ **Productivity gain per year: 35% (186% total)**
 - ◆ **Schedule reduction per year: 19% (52% total)**
 - ◆ **Post-release defect report reduction per year: 39% (82% total)**

Myth #7

**Customers and Managers
Want Short Schedules**



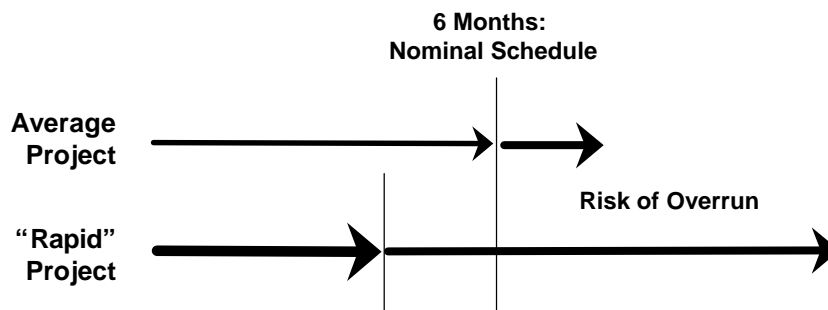
Speed-Oriented Practices-- Better Best Case ...



31



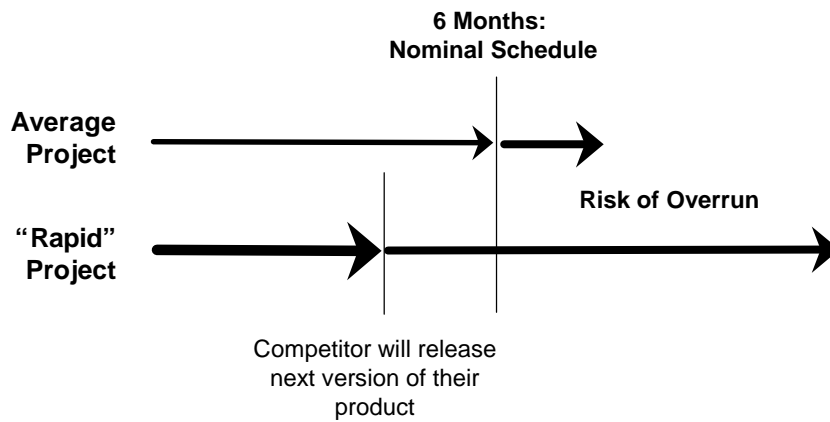
...Worse Worst Case



32



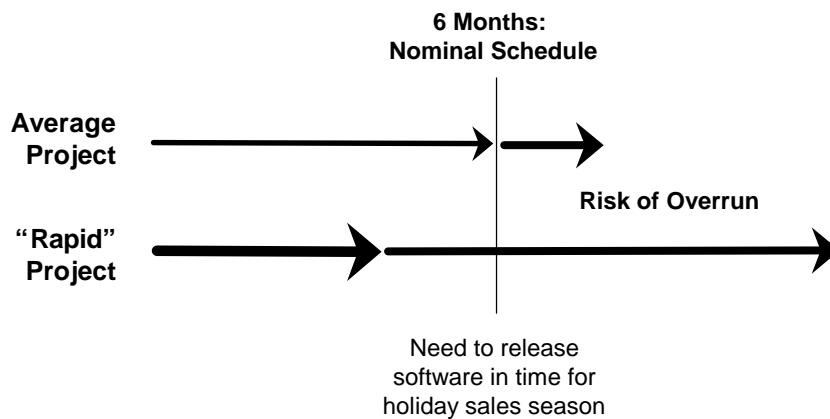
Sometimes People Want the Best Case



33



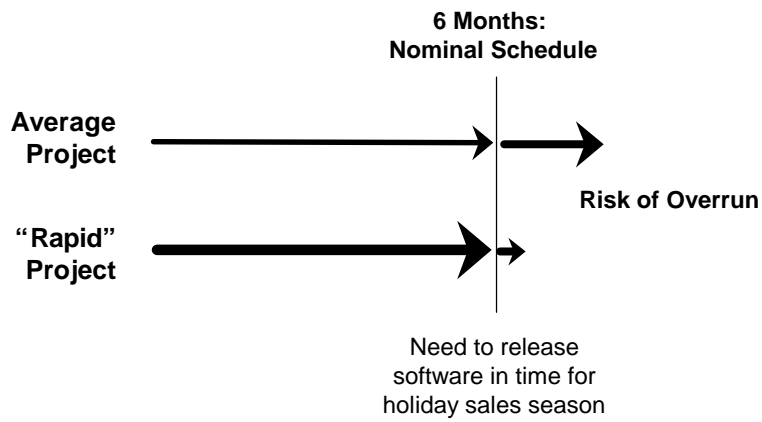
... Sometimes They Don't



34



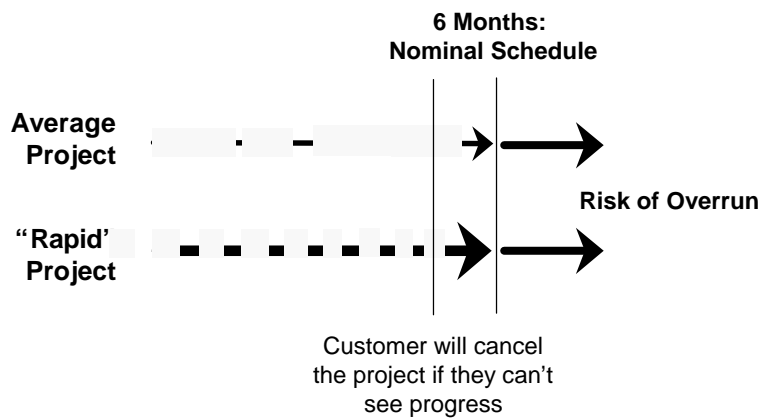
Sometimes They Want Risk-Reduction



35



Sometimes They Want Improved Visibility



36



People Want Many Different Kinds of “Rapid Development”

- ❖ Shortest possible schedule
- ❖ Least risk of overrun
- ❖ Better schedule visibility
- ❖ Better schedule predictability
- ❖ Reduced risk of project cancellation
- ❖ Decreased cost

Myth #8

**Smart Developers Exert
the Biggest Productivity
Impact**



A Lot of Truth to This

- ❖ **20:1 difference in productivity among programmers with similar experience levels**
- ❖ **Similar differences in design quality, program size, debugging performance, and debugging effectiveness**
- ❖ **Programmer effectiveness varies tremendously**
- ❖ **But ...**

39



What if...

- ❖ **A company has two hero developers...**
- ❖ **That spend the whole project arguing with each other instead of working?**

40



What if...

- ❖ **A company's hero developers are assigned to a project ...**
- ❖ **That's ultimately cancelled?**

41



What if...

- ❖ **A company's hero developers very quickly produce functionality ...**
- ❖ **That's eventually cut from the product in order to meet a ship date?**

42



Smart Individuals Don't Guarantee Rapid Results

We should not ignore the importance of smart developers, but ...

- ❖ **Team cohesion matters at least as much**
- ❖ **Organizational characteristics make a significant difference too**

43

Myth #9

**Software Processes Apply
Only to Large, Old-
Fashioned, Bureaucratic
Projects**





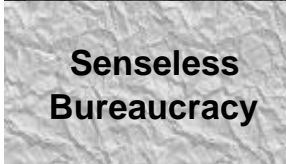

Telcordia (assessed at CMM Level 5)

- ❖ FCC mandated a change
- ❖ Within its Level 5 organization, Telcordia applied its proven practices to ...
 - ◆ Make changes in 3,000 instructions
 - ◆ Spread through 40% of a code base
 - ◆ Consisting of 1 million LOC
 - ◆ No errors reported in next year of operation
 - ◆ Project took 9 hours from requirements analysis through regression testing
- ❖ Admittedly, they did work 1 hour of overtime!

45



Good Processes are Based on Good Judgment

		Well-Defined Practices	
		Yes	No
Good Judgment	Yes		
	No		

Source: Adapted from Telcordia Technologies: The Journey to High Maturity, Bill Pitterman, *IEEE Software*, July 2000

46

Myth #10

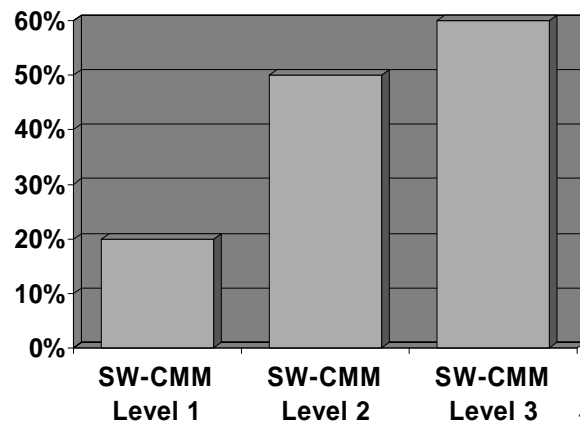
Systematic Development Approaches Hurt Morale



Effect of Process on Morale

❖ 50 Company Survey:

Percentage of people that rated their morale as “good” or “excellent”:





What's Really Bad for Morale?

- ❖ **Long hours**
- ❖ **Unrealistic schedules**
- ❖ **High stress**
- ❖ **Feeling of low productivity**
- ❖ **Working on poor-quality software**

49



Reports from Process-Intensive Companies

- ❖ **Cheyenne Mountain ATAMS Project:**

“Engineers on the project viewed it as a positive experience. They felt the process brought out everyone's best performance, and they said they would be reluctant to develop software without it.”

50



Reports from Process-Intensive Companies

❖ Telcordia (CMM Level 5):

“Our software business has doubled, our profits have grown, and we have measured customer satisfaction at better than 95%. Our on-time delivery is 98% to 99% over the last three years. Our employee turnover rate is in low single digits.”

51



Conclusions



Conclusions

- ❖ **Rapid development is possible!**
- ❖ **Some companies are developing amazingly rapidly (e.g., Telcordia)**
- ❖ **Achieving rapid development requires a combination of well-defined practices and good judgment**
- ❖ **It requires a commitment to avoid the problems associated with the 10 myths**

53



Construx Services

- ❖ **Consulting**
 - ♦ **Project Reviews**
 - ♦ **Requirements Workshops**
 - ♦ **Project Planning**
 - ♦ **Coaching**
 - ♦ **Organizational Assessments**
- ❖ **Training**
 - ♦ **Public Seminars**
 - ♦ **Onsite Seminars**
 - ♦ **Training Needs Assessment**
 - ♦ **Customized Curriculums**
 - ♦ **Executive Briefings**
- ❖ **Software Projects**

54