Modern Project Management

Dr. Mark C. Paulk

SE 4381, Software Project Planning and Management

Jonsson School of Engineering and Computer Science

Management Topics

1. Modern project management

PMBOK

2. Organization strategy and project selection

3. Organization: structure and culture

4. Defining the project

5. Estimating times and costs

6. Developing a project plan

7. Managing risk

8. Scheduling resources and cost

9. Reducing project duration

10. Leadership

11. Teams

12. Outsourcing

13. Monitoring progress

14. Project closure

15. International projects

16. Oversight

17. Agile PM

Critical chain project management

The Software Crisis

Software projects are typically delivered late, over budget, and with less functionality than promised.

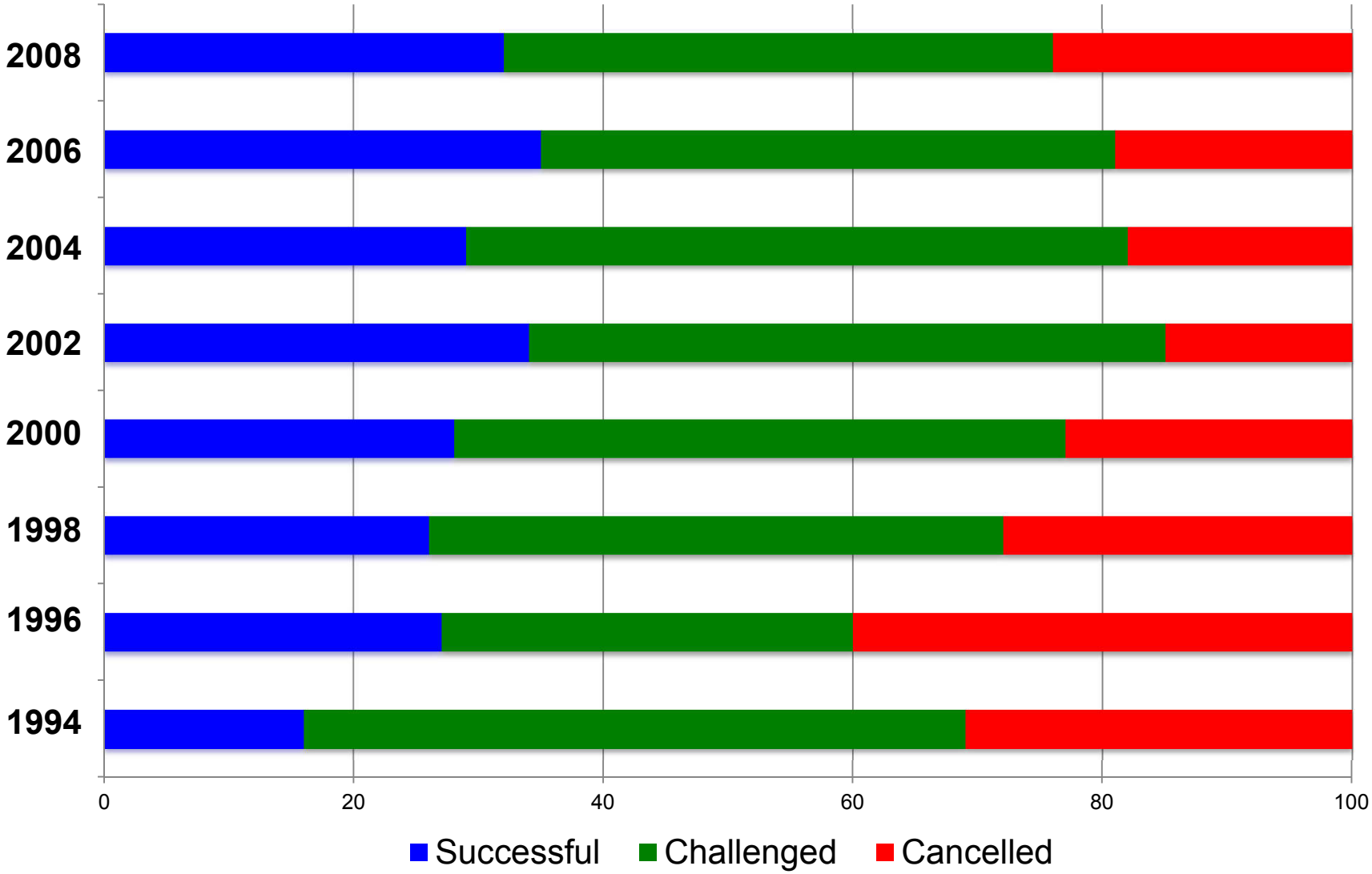
Some of the crisis is because the requirements change.

- **Jones reports typical requirements volatility is 1.6% per month**
- **agile projects typically deal with higher volatility**

Some of the crisis is because of poor management and engineering practices.

- **caught by problems with known solutions**

Standish CHAOS Trends



Improvements in Project Outcomes

The Standish Group credits improvements to

- **better project management**
- **emerging Web infrastructure**
- **iterative development**

Are the Standish Group numbers credible?

- **other researchers have found similar results**

Are improvements really happening?

There is ample opportunity for improvement.

- **adopt the good engineering and management practices that we know work**
- **stop doing things that we know lead to disaster**

CHAOS Factors of Success

User involvement

Executive support

Clear business objectives

Emotional maturity

Optimizing scope

Agile process

Project management expertise

Skilled resources

Execution

Tools & infrastructure

"A Replicated Survey of IT Software Project Failures"

El Emam and Koru (2008)

26-34% of software projects cancelled or unsuccessful

Existing evidence is consistent and shows a decreasing trend in project failures (we're improving as an industry)

Problems

- **estimating the schedule and managing to that estimate**
- **changes in requirements and scope**
- **going over budget**
- **lack of senior management commitment**
  - **misalignment between IT and the business**
- **inappropriate management skills**

A Typical Software Warranty

This product is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Corporation X does not warrant that the functions contained in the product will meet your requirements or that the operation of the product will be uninterrupted or error-free.

Is this a rational response to the reality of software development – from the supplier's side?

Financial Implications of the Software Crisis

Cutter Consortium survey, 2002

78% of IT organizations have been involved in disputes that ended in litigation

67% of cases, functionality / performance did not meet software developer claims

56% of cases, promised delivery date slipped several times

45% of cases, defects were so severe the software was unusable

The Problem Is Management

When a software project is cancelled, management failure is usually the root cause.

"The task force is convinced that today's major problems with military software development are not technical problems, but management problems."

- **Report of The Defense Science Task Force on Military Software, Sept 1987, Fred Brooks Chairman**

Sociocultural Aspects

The #1 criterion for management candidates: "works well with others"

Good project managers balance their attention to both the

- **technical**
- **sociocultural**

aspects of project management.

The State of the Practice?

"I'd rather have it wrong than have it late. We can always fix it later."

- **A senior software manager (industry)**

"The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost."

- **A program manager (government)**

"By regularly putting the development process under extreme time pressure and then accepting poor-quality products, the software user community has shown its true quality standard."

- **DeMarco and Lister (*Peopleware*)**

The Iron Triangle

Projects have imposed constraints of scope, cost (resources), and time (schedule).

Changing one constraint affects one or both of the others.

- **You cannot "fix" all three at the same time.**

Karl Wiegers suggests there are really five interrelated project dimensions.

- **Creating a Software Engineering Culture, 1996**

- **scope**
- **schedule**
- **cost**
- **staff**
- **quality**

Basic Project Management Questions

Should this project be initiated?
Should it continue or be terminated?

Who is on the team?

Do we have effective (high performing) teams?

What life cycle model will we use?

What is our decision making process?
How is our team/project structured?

How are we managing risks?

Do we have good communications (expectation management) with the customer? With our staff?

Business Drivers – Defining Value

Operational excellence?
- **reliable products at competitive prices delivered with minimal inconvenience**

Product leadership?
- **leading edge products that enhance the customer's utility**

Customer intimacy?
- **segmenting and targeting market niches precisely**

M. Treacy and F. Wiersema, The Discipline of Market Leaders, 1997.

Dimensions of Project Success
(Shenhar, 1997)

Four universal dimensions of success
- **project efficiency**
- **impact on the customer**
- **business and direct success**
- **preparing for the future**

During project execution, only three dimensions seem important to project managers
- **to please the customer**
- **to meet time and budget goals**
- **to some extent to succeed commercially**

Fayol's Basic Management Functions

Planning – what are we aiming for and why?

Organizing – what is involved and why?

Staffing / Motivation – what motivates people to do their best work?

Directing – who decides what and when?

Controlling – who judges results and by what standards?

D. Cleland, Project Management: Strategic Design and Implementation, Second Edition, 1994.

What Is a Best Practice?

A management or technical practice consistently demonstrated to significantly improve the bottom line.

- **productivity**
- **development and/or maintenance costs**
- **schedule**
- **quality**
- **user satisfaction**
- **predictability of cost and schedule**

Demonstrated high return-on-investment (ROI)

Software Program Managers Network

Best practice? Good practice? Recommended?

CMMI-DEV v1.3

Level	Process Characteristics	Process Areas
5 Optimizing	*Focus is on quantitative continuous process improvement*	Causal Analysis & Resolution Organizational Performance Management
4 Quantitatively Managed	*Process is measured and controlled*	Organizational Process Performance Quantitative Project Management
3 Defined	*Process is characterized for the organization and is proactive*	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organization Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis & Resolution
2 Managed	*Process is characterized for projects and is often reactive*	Requirements Management Project Planning Project Monitoring & Control Supplier Agreement Management Product & Process Quality Assurance Configuration Management Measurement & Analysis
1 Initial	*Process is unpredictable, poorly controlled, and reactive*	

Software Program Managers Network

Software Program Managers Network

- **identify proven industry and government software best practices**
- **convey these practices to managers of large-scale DoD system acquisition programs**

***16 Critical Software Practices*™ specifically address underlying cost and schedule drivers that have caused many software intensive systems to be delivered over budget, behind schedule and with significant performance shortfalls.**

<URL: www.spmn.com/>

16 Critical Software Practices

PROJECT INTEGRITY

1. **Adopt Continuous Program Risk Management**
2. **Estimate Cost and Schedule Empirically**
3. **Use Metrics to Manage**
4. **Track Earned Value**
5. **Track Defects against Quality Targets**
6. **Treat People as the Most Important Resource**

CONSTRUCTION INTEGRITY

7. **Adopt Life Cycle Configuration Management**
8. **Manage and Trace Requirements**
9. **Use System-Based Software Design**
10. **Ensure Data and Database Interoperability**
11. **Define and Control Interfaces**
12. **Design Twice, Code Once**
13. **Assess Reuse Risks and Costs**

PRODUCT STABILITY AND INTEGRITY

14. **Inspect Requirements and Design**
15. **Manage Testing as a Continuous Process**
16. **Compile and Smoke Test Frequently**

McConnell's Rapid Development

S. McConnell, Rapid Development: Taming Wild Software Schedules,1996.

Best practices drawn from commercial, shrinkwrap environment, e.g., Microsoft

See the handout on eLearning for a list and brief description of the Rapid Development best practices.

Change Board

Daily Build and Smoke Test

Designing for Change

Evolutionary Delivery

Evolutionary Prototyping

Goal Setting

Inspections

Joint Application Development (JAD)

Lifecycle Model Selection

Measurement

Miniature Milestones

Outsourcing

Principled Negotiation

Productivity Environments

Rapid-Development Languages

Requirements Scrubbing

Reuse

Signing Up

Spiral Lifecycle Model

Staged Delivery

Theory W Management

Throwaway Prototyping

Timebox Development

Tools Group

Top-10 Risks List

User Interface Prototyping

Voluntary Overtime

ISO/IEC 12207 (Software Life Cycle Processes)

A common framework for software life cycle processes

- **with well-defined terminology**

Contains processes, activities, and tasks that are to be applied during the acquisition of

- **a system that contains software**
- **a stand-alone software product**
- **software service**
- **during the supply, development, operation, and maintenance of software products**

The Agile Alliance

Agile Manifesto

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

- ***individuals and interactions*** **over processes and tools**
- ***working software*** **over comprehensive documentation**
- ***customer collaboration*** **over contract negotiation**
- ***responding to change*** **over following a plan**

That is, while there is value on the items on the right, we value the items on the left more."

<URL: www.agilealliance.org>

Important Agile Methods

Scrum

Extreme Programming (XP)

Crystal Light Methods, specifically Crystal Clear

Feature Driven Development

Lean Development

Kanban

and others…

Critical Chain Project Management (CCPM)

E.M. Goldratt, Critical Chain,1997.

R.C. Newbold, Project Management in the Fast Lane: Applying the Theory of Constraints,1998.

"The uncertainties embedded in projects are the major causes of what we call mismanagement."

- **a new management philosophy**
- **research methods from accurate sciences adapted to systems that contain humans**
- **broad spectrum of robust applications**

D.J. Anderson, Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, 2004.

CCPM: Cost vs Throughput

The *cost world* implies any local improvement automatically translates into an organizational improvement.

Consider a chain of links…

When we deal with throughput, linkages are as important as links – the chain is only as strong as the weakest link.

The *throughput world* implies most local improvements do not contribute to global improvement…

The Six Traits of Good Project Managers *(Baker 1998)*

Enthusiasm for the project

The ability to manage change effectively

A tolerant attitude toward ambiguity

Team building and negotiation skills

A customer-first orientation

Adherence to the priorities of business

Project Management Advances
(Pinto, 2002)

Profound recent advances in project management

- **risk management**
- **scheduling**
  - **Critical Chain Project Management**
- **structure**
  - **heavyweight project organization**
  - **project management offices**
- **project team coordination**
  - **cross-functional cooperation**
  - **punctuated equilibrium model**
- **control**
  - **earned value analysis**
- **impact of new technologies**
  - **Internet, Web**
  - **distributed and virtual project teams**

Project Management Trends

(Pinto, 2002)

Principle reason for introduction of *PMOs* (project management offices) is desire to create a central administrative center for the coordination and management of project portfolios

Goal of *cross-functional teams* is to minimize time lost to rework cycles by creating a min-project organization from the beginning of the project

Punctuated equilibrium

- **real natural change comes about through long periods of stasis, interrupted by some cataclysmic event that propels upward, evolutionary adjustment**
- **five-staged model: forming, storming, norming, performing, and adjourning**

Touching All The Bases

Classic project management
- **PMBOK**

Software process management
- **Software CMM, CMMI**

DoD, government contracting
- **SPMN**

Commercial, shrinkwrap
- **McConnell**

Internet, Web-based, agile
- **Scrum, Extreme Programming**

Leading-edge management theory
- **Critical Chain Project Management**

Questions and Answers