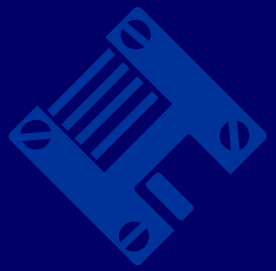
10 Keys to Successful Software Projects

An Executive Guide

www.construx.com

Construx

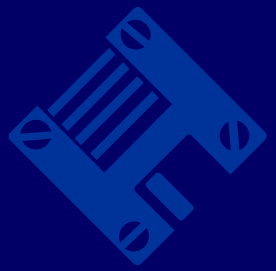
Delivering Software Project Success



State of the Art vs. State of the Practice

“The gap between the best software engineering practice and the average practice is very wide–perhaps wider than in any other engineering discipline.”

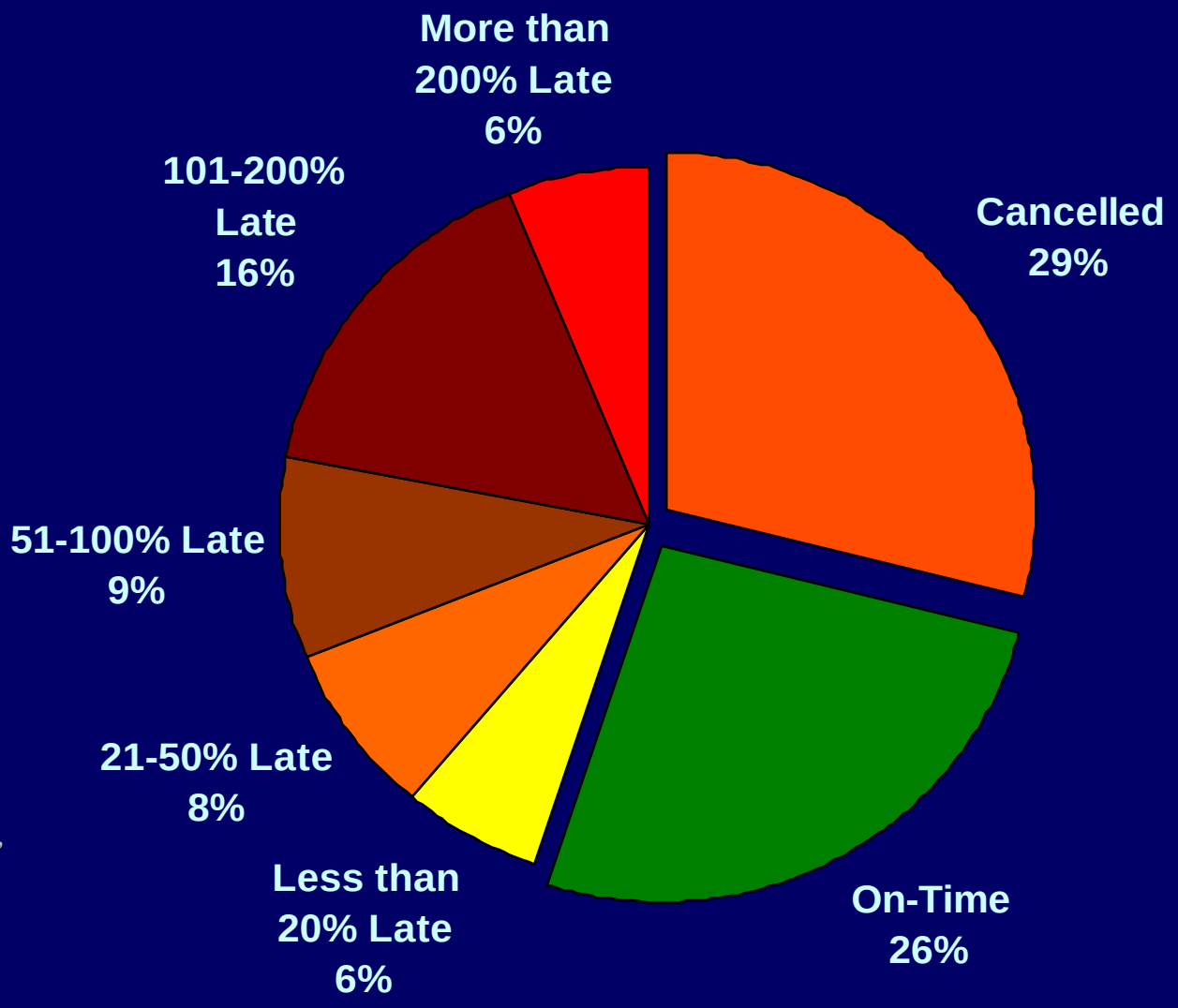
– Fred Brooks



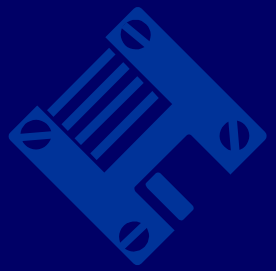
What is Brooks Talking About?

- Project planning and management practices
  - Automated estimation tools (1973)
  - Evolutionary delivery (1988)
  - Measurement (1977)
  - Productivity environments (1984)
  - Risk management planning (1981)
- Similar situation in requirements, design, testing, quality assurance, etc.

Typical Project Outcomes

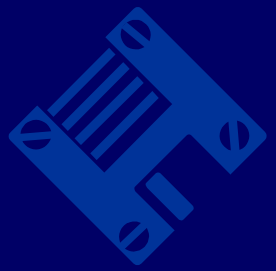


Source: Standish Group Survey, 1999 (from a survey of 8000 business systems projects)



Most Common Sources of Cancellations and Overruns

1. **Ill-defined or changing requirements**
2. **Poor project planning/management**
3. **Uncontrolled quality problems**
4. **Unrealistic expectations/inaccurate estimates**
5. **Naive adoption of new technology**



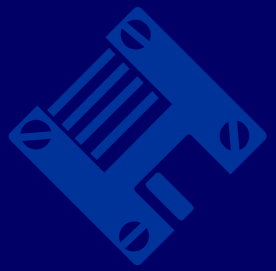
Project Success

Success = Planning * Execution


Planning

Key #1

Clear Vision

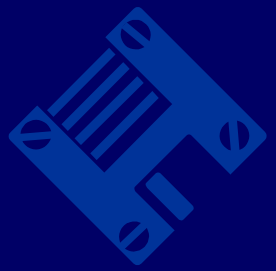


Clear Vision

- **Project teams work toward the goals you set for them**
- **Too many goals = no goals**
- **Good vision statement describes *what to leave out*—prioritizes**

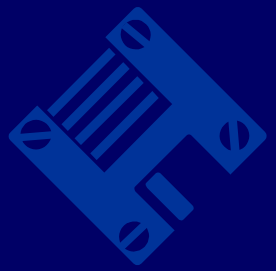
Key #2

Stable, Complete, Written Requirements



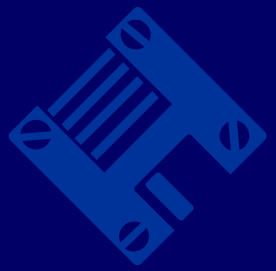
Common Requirements Problems

- Customer changes his/her mind
- Customer doesn’t know what he/she wants
- Market conditions change
- Competitor releases a new version
- Etc.



Are Requirements Changes Inevitable?

- Requirements change is the most common software project risk
- Achieving 100% stable requirements usually isn't possible, but...
- Most requirements changes arise from requirements that were incompletely defined in the first place
- With appropriate practices, most of the "common problems" are just excuses

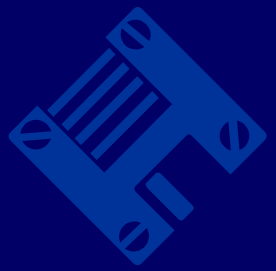


We Have Many Techniques for Defining Requirements

- User interface prototyping
- Requirements workshop
- User interview
- Use cases
- User manual as spec
- Usability studies
- Requirements reviews/inspections
- Incremental delivery
- **These practices are important because stable requirements are needed for high quality software**

Key #3

Detailed User Interface Prototypes

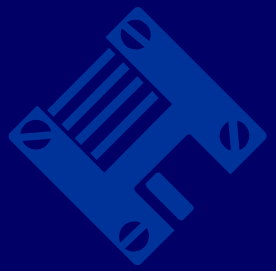


User Interface Prototypes

- Addresses the most common project risk–changing requirements
- Involves users with a “live” medium
- Correlated with lower costs, shorter schedules, and higher user satisfaction
- May have to think about who the “user” is
- Significant skill required to develop prototypes successfully

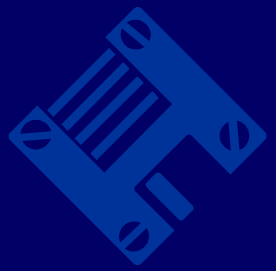
Key #4

Effective Project Management



Project Management

- **Poor planning/management is the second most common project risk**
- **Where do most project managers come from?**
- **What are they trained to do?**
- **Some people don't appreciate software project management—they've never seen *good* project management!**

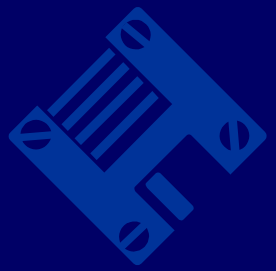


Project Manager Responsibilities

- **Good software management requires significant software-specific expertise**
  - **Estimation**
  - **Lifecycle selection**
  - **QA planning**
  - **Technical staffing**
  - **Project tracking**
  - **Risk management**
  - **Data collection**

Key #5

Accurate Estimates

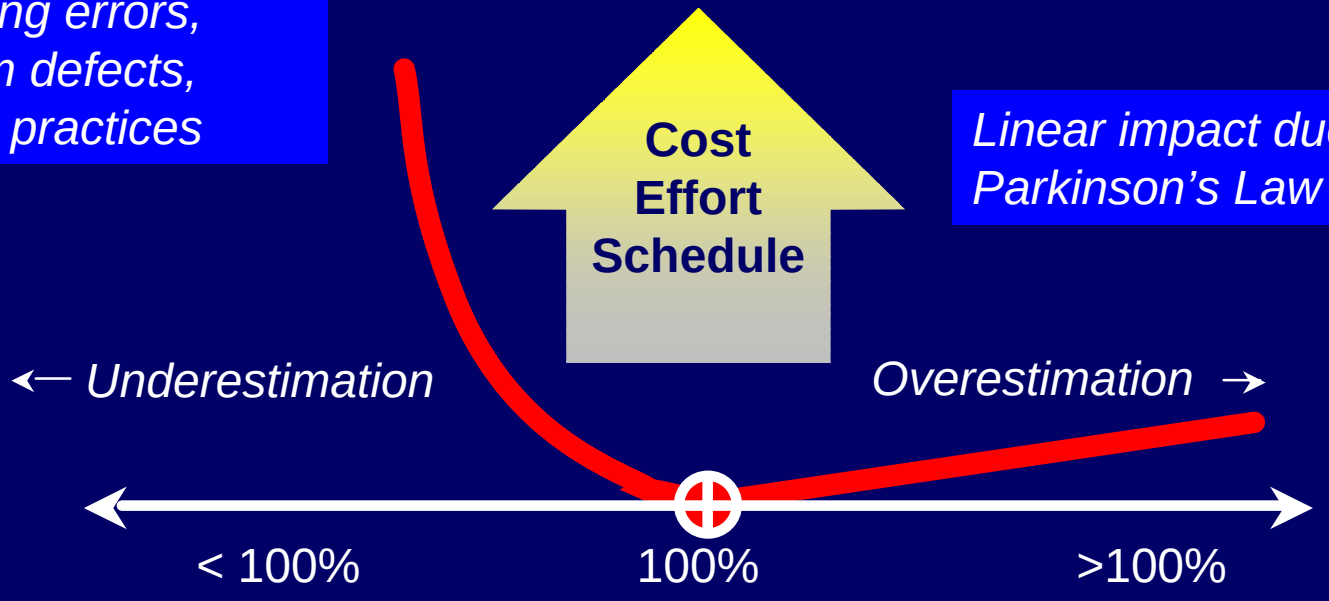


Need for Accurate Estimates

- **Unrealistic/unjustified expectations are a major cause of project problems**
- **State of the art is dramatically better than the state of the practice**

Effect of Estimation Accuracy

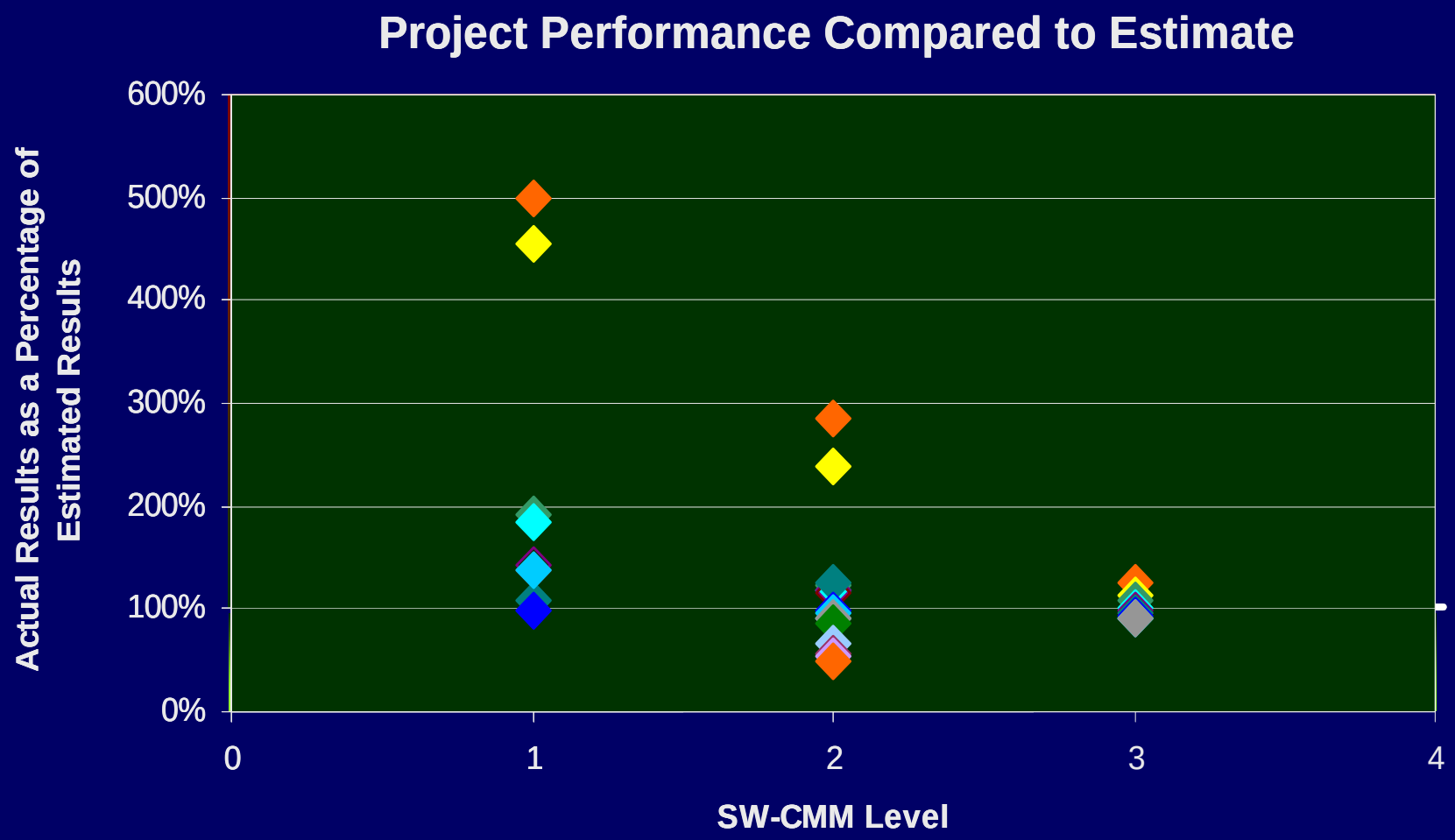
Non-linear impact due to planning errors, upstream defects, high-risk practices



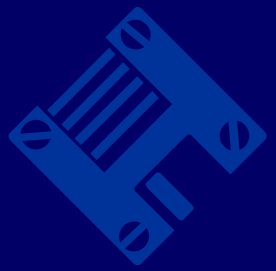
Linear impact due to Parkinson's Law

Target as a Percentage of Nominal Estimate

Improved Estimation



From a set of U.S. Air Force projects



Accurate Estimation

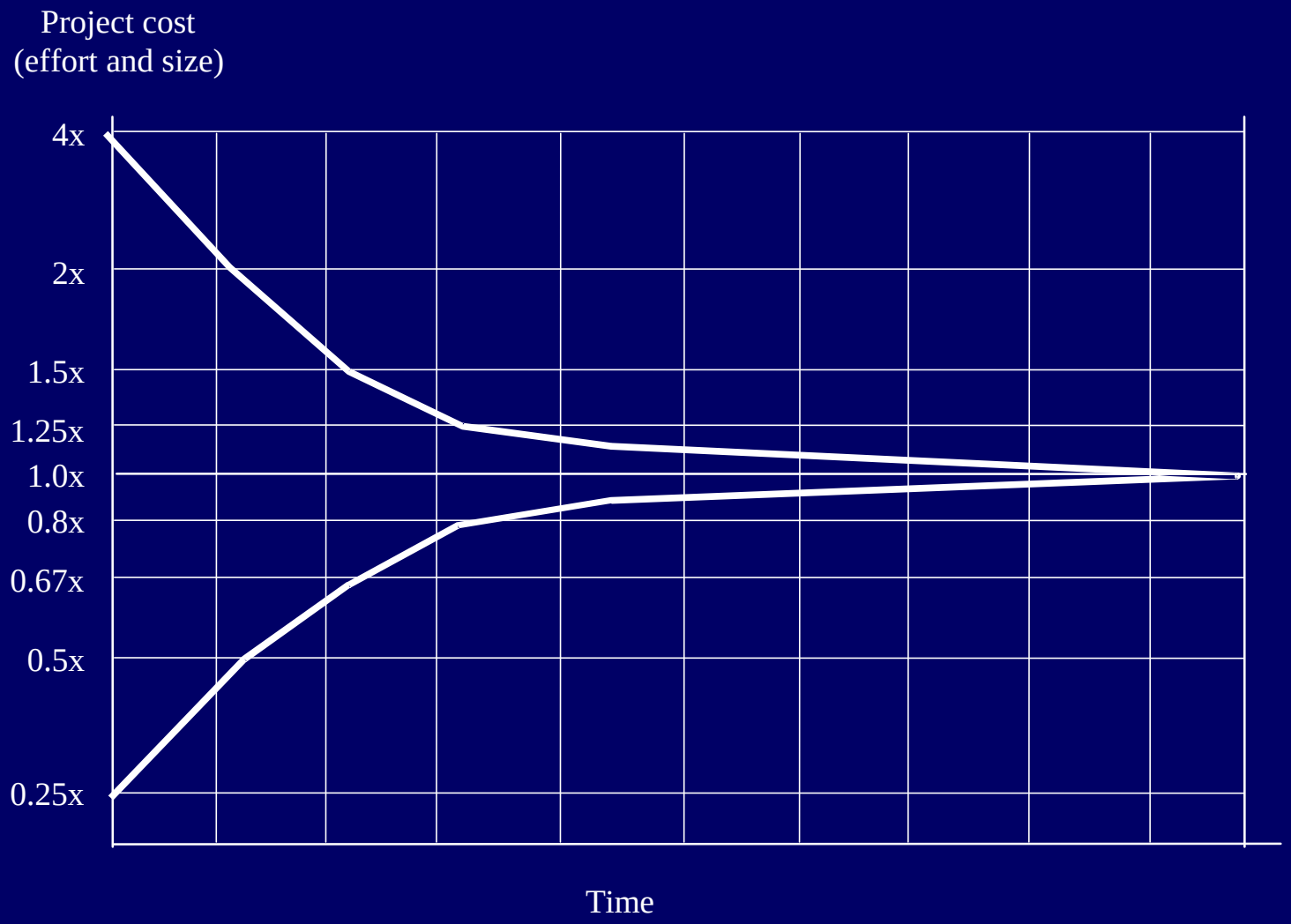
- **Estimation is a specialized technical skill**
- **Treat creation of the estimate as a mini-project**
- **Plan to re-estimate periodically**


Execution

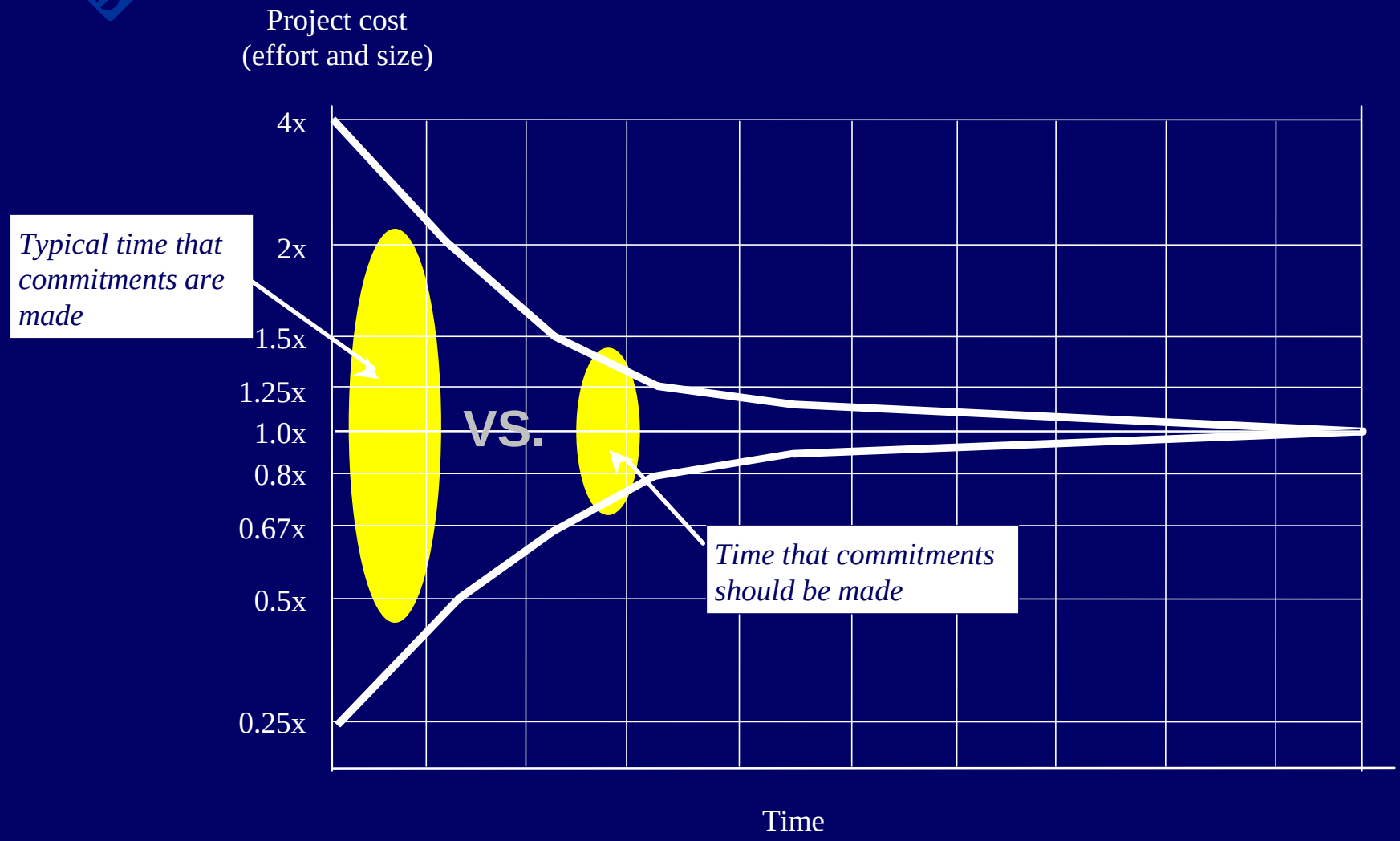
Key #6

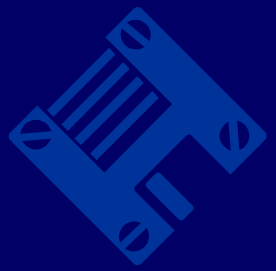
Two-Phase Budgeting

Estimate Refinement



Two-Phase Estimation and Budgeting





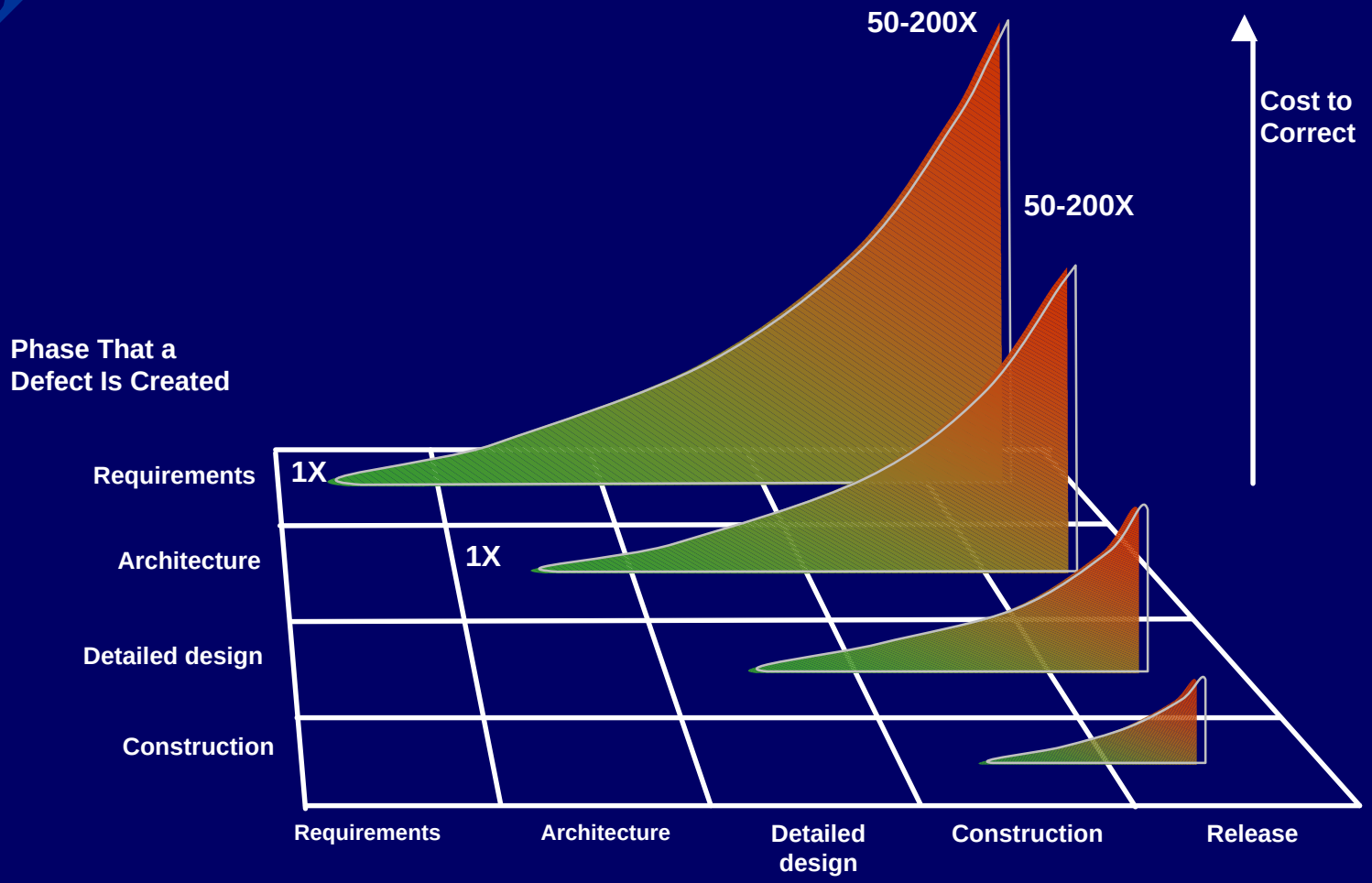
Benefits

- Delays commitment until time when it can be meaningful
- Forces activities to occur upstream that should occur upstream
- Helps set realistic expectations for all project stakeholders
- Improves coordination with non-software groups
- Improves execution by putting plans on more informed basis
- Can be more than two phases

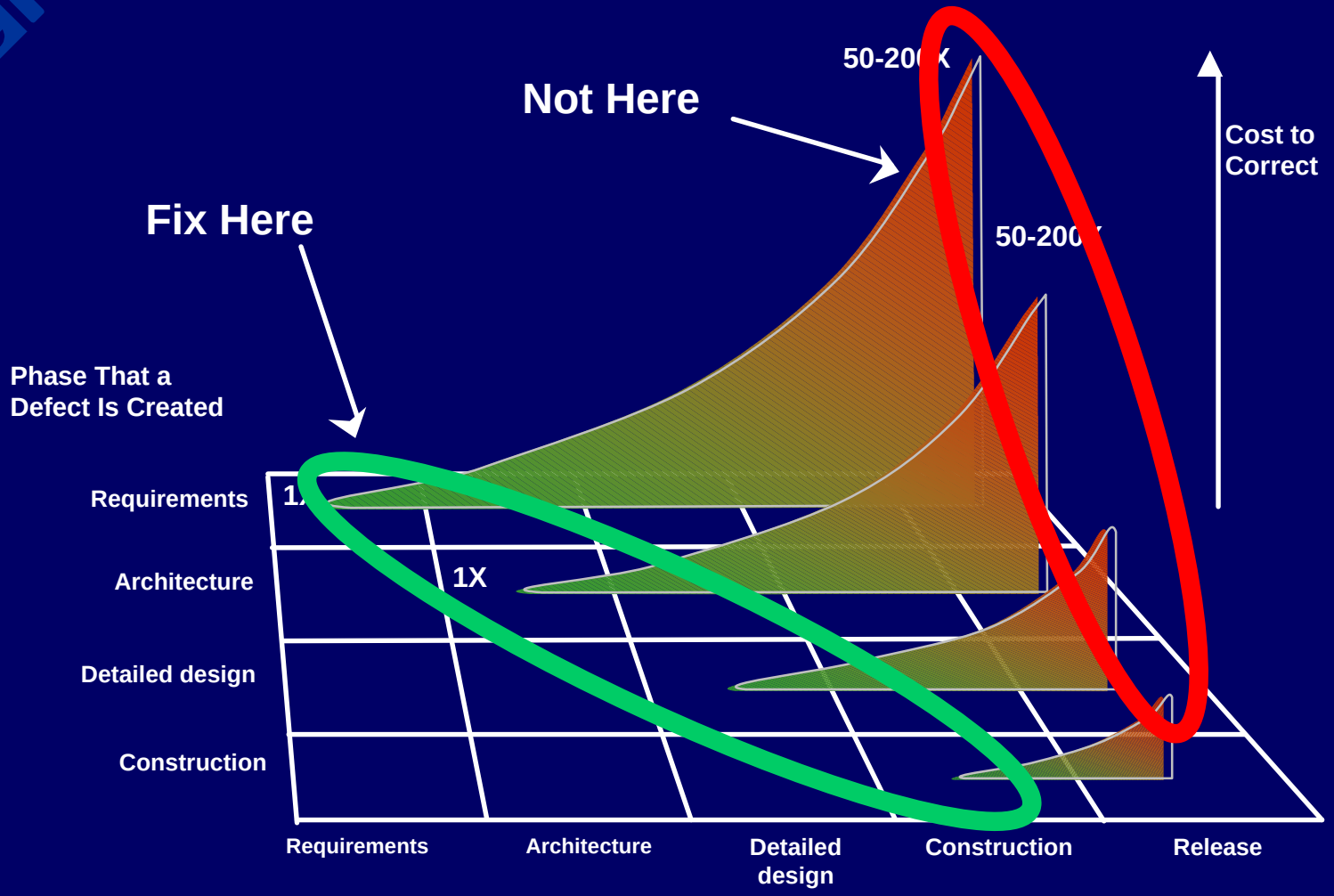
Key #7

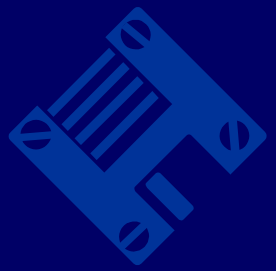
A Focus on Quality

Defect Costs Increase the Longer Defects Stay in Process



Fix More Defects Earlier!



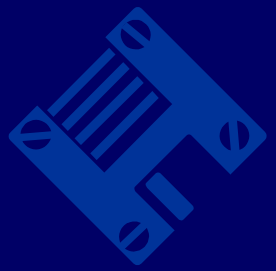


Why Focus on Quality?

- For most projects, unplanned defect correction work is the largest cost driver (40-80% of total)
- Can focus on quality for sake of economics (as above)
- Can focus on quality for sake of quality (not needed nearly as often)
- Quality must be planned into the project; it can't just be tacked onto the end
- Changes are destabilizing, and it's better to eliminate defects early

Key #8

Technology Expertise

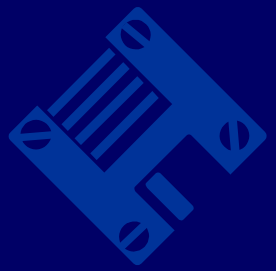


Technology Expertise

- **Many projects suffer because of poor adoption of new technology**
- **“New technology” = high risk**
- **Success depends on having technology expertise, which you don’t have the first time you deploy a new technology**

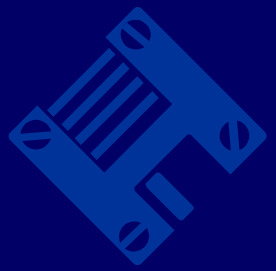
Key #9

Active Risk Management



Role of Risk Management

- About as many projects fail as are delivered on time
- More than 50% of projects show their problems during initial development
- About 25% show their problems during initial planning
- Active risk management keeps small problems from turning into big, project-killing problems

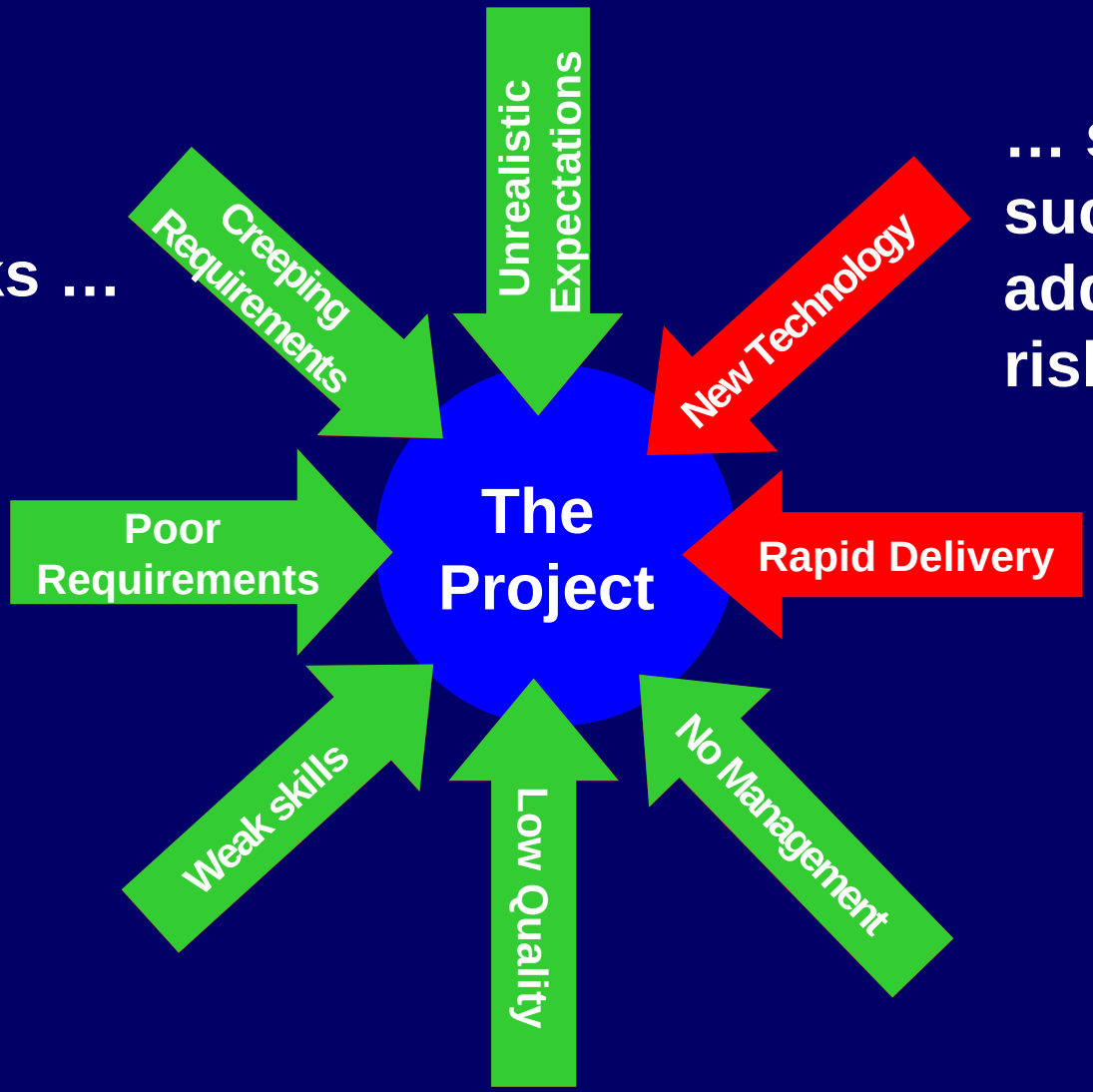


Relationship to Business Risk Taking

- **"We're not afraid of risk, and we don't try to avoid it; we're an internet company"**
  — *former dot com, now out of business*
- **Perception is that high energy companies take risks**
- **Reality is that most companies are beset by risks from all sides—they aren't *choosing* which risks they take**
- **Key to success: Manage non-strategic risks so that you can take strategic risks**

Example of Managing Risks

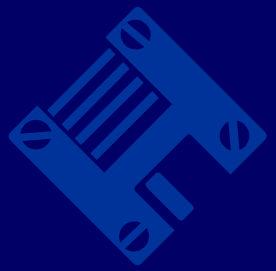
Manage these risks …



… so you can successfully address these risks

Key #10

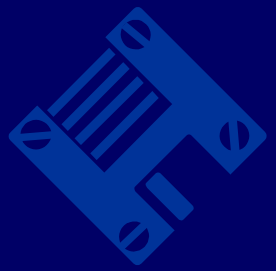
Remember, Software Is Created By Humans



Human Factors

Take care of your people

- **Staffing**
- **Training**
- **Motivation/morale**
- **Work environment**



Summary

Q: What are the most exciting/promising software engineering ideas or techniques on the horizon?

A: I don’t think that the most promising ideas are on the horizon. They are already here and have been here for years but are not being used properly.

— David L. Parnas

Construx

Delivering Software Project Success

Contact Information

Services

- **Software Projects**
- **Consulting**
- **Seminars**

Software Resources

- ***sales@construx.com***
- ***www.construx.com***