The Business Case for Improved Production Practices

Steve McConnell

www.construx.com

igda international game developers association

Construx
Delivering Software Project Success

Observations

Many Different Kinds of Software Exist

- Inhouse business systems (e.g., inventory control, phone center management)
- Internet applications (e.g., Amazon, eBay, asp products)
- Desktop applications (e.g., Word, Excel, TurboTax, PrintShop)
- Systems software (e.g., Windows XP, Mac OS)
- Embedded systems (e.g., antilock brakes, pacemakers, ATM machines)
- Aerospace (e.g., avionics software)
- Military systems (e.g., command & control)
- Games (e.g., MVP Baseball, Sims, Half-Life, Halo)

Different Kinds of Software Call for Different Development Practices

- Major Development Considerations
  - Market-timing constraint
  - Consequence of defects (human life, human safety, economic loss, inconvenience)
  - Extent of distribution/ease of re-distribution
  - Number of disciplines involved (i.e., artists, designers, developers, etc.)
  - Level of polish needed

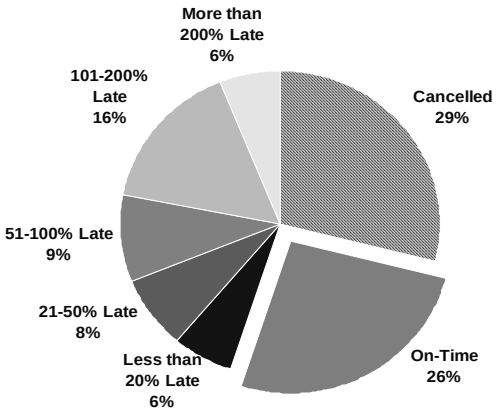
Industry Segments Don't Cross-Pollinate Much

- Many software industry segments are re-inventing the wheel
  - Large projects
  - Multi-disciplinary projects
  - Strong company-culture influences
  - Growth "inflection points"
- Methodologies can be chosen consciously, rather than unconsciously
- There is much that industry segments can learn from one another

State of the Practice

Disappointing Project Outcomes

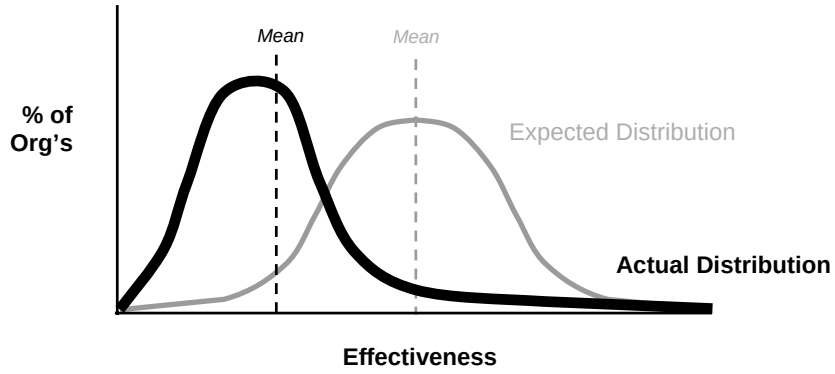
- Average schedule overrun may be as high as 100%
- About one-quarter of all projects are cancelled
- Reasons for these outcomes are complicated



Causes of Disappointing Project Outcomes

- Actual inefficiencies
  - Most projects are run somewhat inefficiently
  - Average developer reads less than 1 professional book/year and subscribes to no professional journals
- Perceived inefficiencies
  - Management & customer expectations are often unrealistic and unachievable
  - Some management and customer actions actually undermine effective project performance

Average Practice is Close to the Worst Practice



10 Tough Questions for Software Executives

1. How much are you spending on software?
2. How do your teams' skills compare to industry averages?
3. How do the capabilities of your organization compare to other, similar organizations?
4. What percentage of your costs arise from unplanned rework?
5. How confident are you that your "buy" decisions should not be "build" decisions?

10 Tough Questions for Software Executives (cont.)

6. What percentage of your projects are on time and on budget?
7. How confident are you that your current projects will perform to their estimates?
8. What percentage of your current projects are most likely to be cancelled?
9. How satisfied (quantitatively) are users of your products?
10. How much (quantitatively) has your productivity improved in the past 12 months?

ROI

Improved Software Practices are Business's

Last Great Frontier

ROI for Selected Practices

Practice	12-month ROI	36-month ROI
Formal code inspections	250%	1200%
Formal design inspections	350%	1000%
Cost and quality estimation tools	250%	1200%
Long-range technology planning	100%	1000%
Productivity measurements	150%	600%
Process assessments	150%	600%
Management training	115%	550%
Technical staff training	90%	500%

Source: Capers Jones, Assessment and Control of Software Risks, Prentice Hall, 1994.

ROI

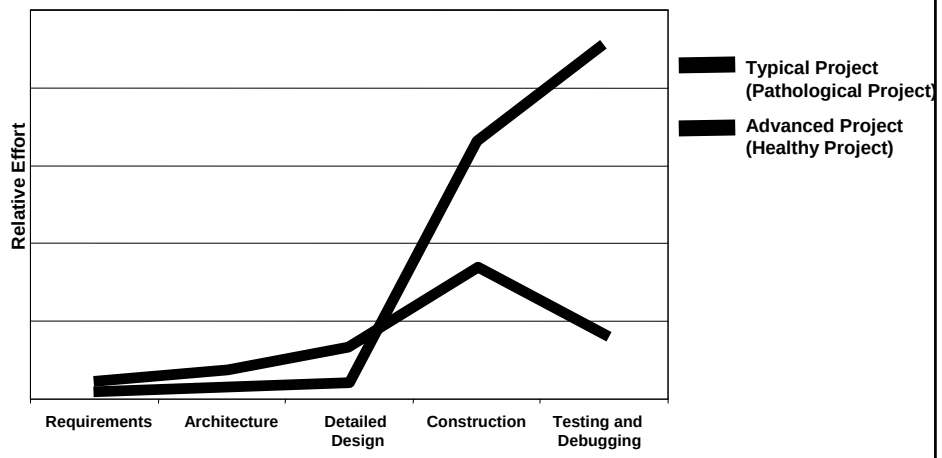
- Improved software practices pay an average ROI of 500% (including false starts), and continued improvement is sustainable for many years
- The best organizations have sustained ROIs of 900% from software improvement initiatives for many years

Source: James Herbsleb, et al, "Benefits of CMM Based Software Process Improvement: Initial Results," Pittsburgh: Software Engineering Institute, Document CMU/SEI-94-TR-13, August 1994.

Where Does the "R" Come From?

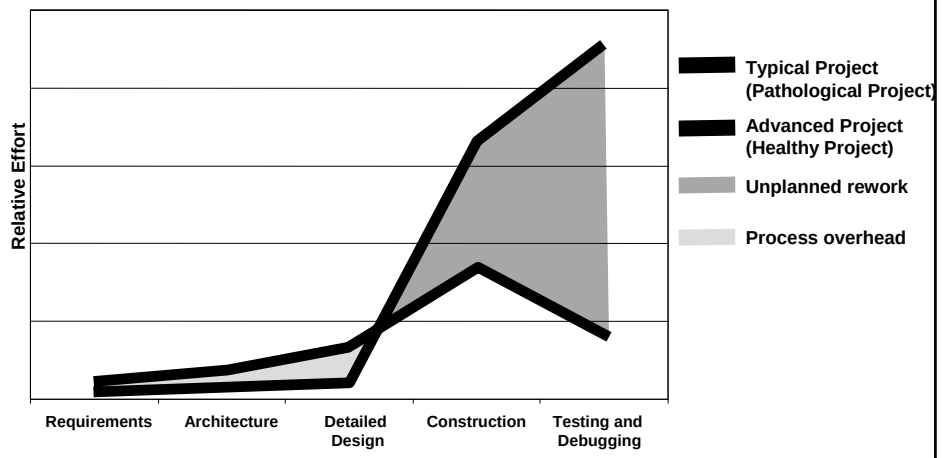
Where Costs Come From:

Lifecycle Cost Profile



Where Costs Come From:

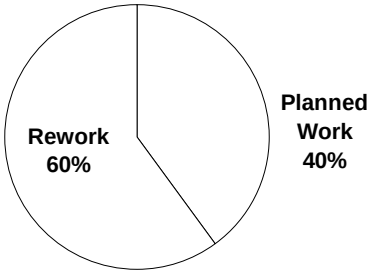
Lifecycle Cost Profile (cont.)



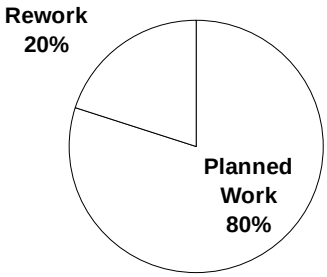
Where Costs Come From:

Activity Breakdown

Cost Breakdown for an Average Project



Cost Breakdown for an Expertly-Run Project



Reduced Cost

- **Improving software practices reduce costs an average of ~35% per year**
- **The improvement is sustainable for several years, and the potential is much higher**
- **The best organizations have sustained cost improvements of 55%+ per year**

Source: James Herbsleb, et al, "Benefits of CMM Based Software Process Improvement: Initial Results," Pittsburgh: Software Engineering Institute, Document CMU/SEI-94-TR-13, August 1994.

Improved Quality

- Poor quality is the single largest cost driver for most projects
- Improved software practices improve quality an average of ~40% per year
- This improvement is sustainable for several years, and the potential is much higher
- The best organizations have sustained quality improvements of 70%+ per year

Sources: Steve McConnell, Rapid Development, Microsoft Press, 1996.

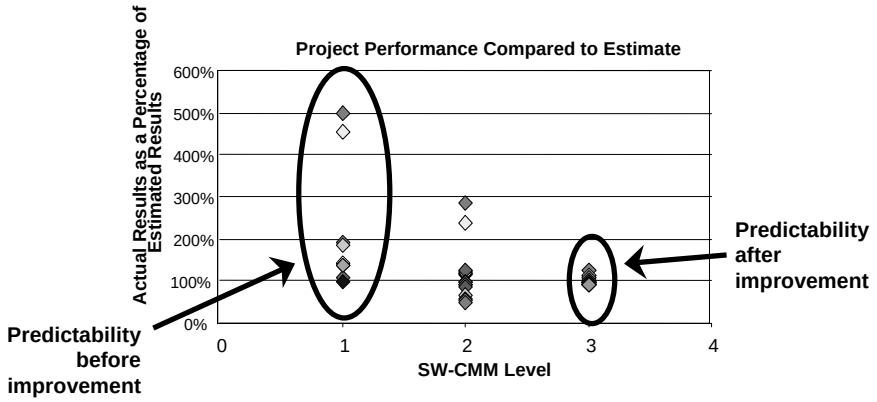
James Herbsleb, et al, "Benefits of CMM Based Software Process Improvement: Initial Results," Pittsburgh: Software Engineering Institute, Document CMU/SEI-94-TR-13, August 1994.

Improved Cycle Time

- Improved software practices shorten schedules an average of ~15-20% per year
- This improvement is sustainable for several years, and the potential is much higher
- The best organizations have sustained schedule improvements of ~20-25% per year

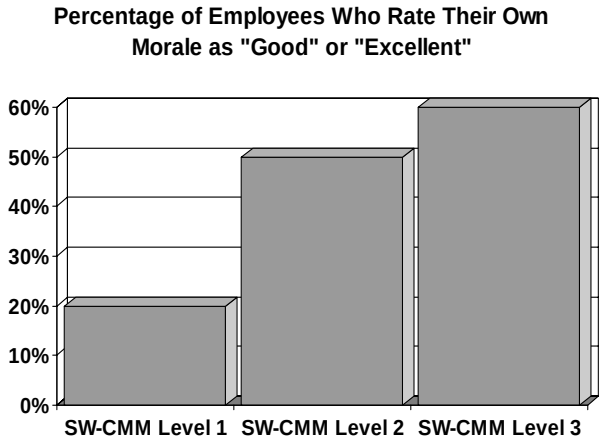
Source: James Herbsleb, et al, "Benefits of CMM Based Software Process Improvement: Initial Results," Pittsburgh: Software Engineering Institute, Document CMU/SEI-94-TR-13, August 1994.

Better Predictability



Source: Dr. Patricia K. Lawlis, Capt. Robert M. Flowe, and Capt. James B. Thordahl. "A Correlational Study of the CMM and Software Development Performance," Crosstalk, September 1995.

Enhanced Morale



Source: James Herbsleb, et al. "Software Quality and the Capability Maturity Model," Communications of the ACM,June 1997, pp. 30-40.

ROI Examples

Organization	Results
Boeing Information Systems	Estimates within 20%, $5.5 million saved in 1 year
BDN	**ROI 300%**
CSC	65% reduction in error rates
Harris ISD DPL	90% defect rate reduction; 2.5x productivity gain
Hewlett-Packard SESD	**ROI 900%**
Hughes	$2 million annual reduction in cost overruns

ROI Examples

Organization	Results
IBM Toronto	90% reduction in delivered defects, 80% reduction in rework
Motorola GED	2-3X productivity improvement, 2-7X cycle time reduction, **ROI 677%**
Philips	**ROI 750%**
Raytheon	**ROI 770%**
Siemens	90% reduction in released defects
Schlumberger	4X reduction in beta test bugs

ROI Examples

Organization	Results
Telcordia	Defects 1/10 industry average, customer satisfaction increased from 60-91% over 4 years
Texas Instruments – Systems Group	90% reduction in delivered defects
Thomson CSF	**ROI 360%**
US Navy	**ROI 410%**
USAF Ogden Air Logistics Center	**ROI 1900%**
USAF Oklahoma City Air Logistics Center	**ROI 635%**
USAF Tinker Air Force Base	**ROI 600%**

Other Benefits

- **Direct ROI is from better *operational efficiency***
- ***Indirect ROI may be greater***
  - **Predictability--product launch, changes in business practices, etc.**
  - **Inter-group coordination**
  - **Cost control**
  - **Risk Reduction**

ROI Implications

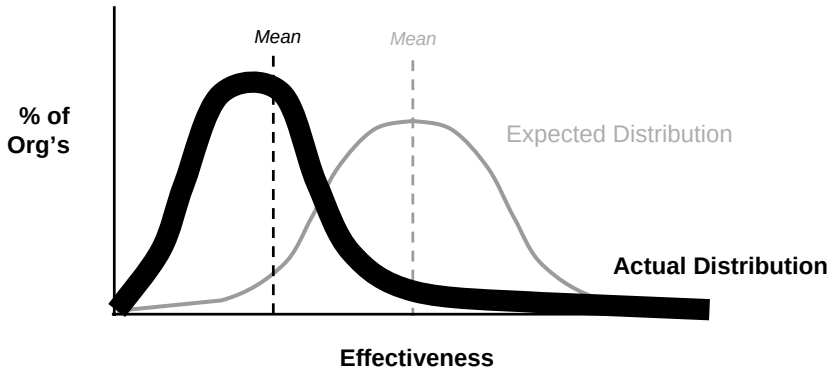
- Size of investment required varies
- Payback periods vary
- Not all investments are possible initially--some larger ROIs are dependent on previous investments with smaller ROI
- Best starting point depends on organizational specifics

What Prevents Companies from Seizing This Opportunity?

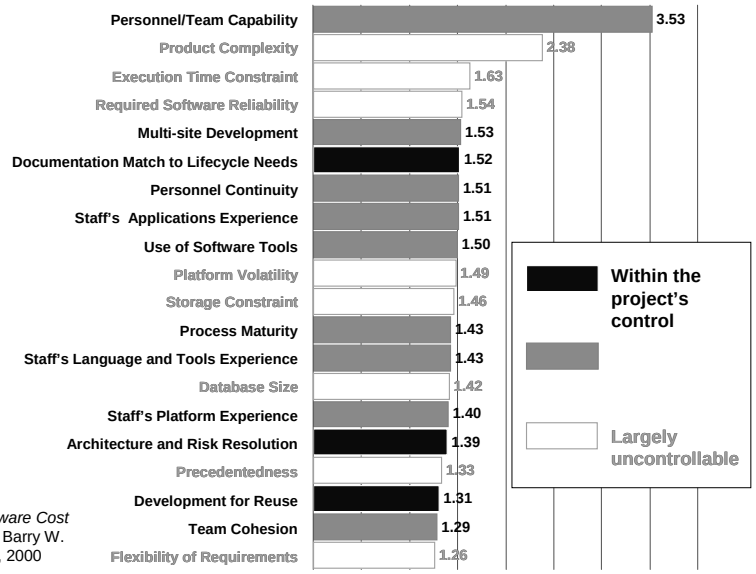
Procrastination:

“Hard work often pays off over time, but laziness always pays off now”

Most Software Professionals Have Not Seen Software Development at its Best



Where's the Leverage?



Source: Adapted from *Software Cost Estimation with Cocomo II*, Barry W. Boehm, et al, Prentice Hall, 2000

Organizational Focus

- Few factors are readily within the control of a single project
- Few factors are totally outside the control of both the project and the organization
- Leverage for improved software practices is mostly at the organizational level rather than the project level

What Prevents Companies from Seizing the Opportunity?

- Fear of "killing the goose that's laying the golden eggs"
- Successful small projects cause complacency--leading to unsuccessful large projects
- Can't see how to apply lessons from other industry segments
- Too much time spent fighting current fires to prevent future fires
- Many companies <u>are</u> seizing the opportunity!

Focus on Low Hanging Fruit

- Lots of proven practices are available (though they might have been proven in other industries)
- Risk of not using these practices is substantially higher than of using them

Low Hanging Fruit (year first available)

- Project planning and management practices
  - Automated estimation tools (1973)
  - Evolutionary delivery (1988)
  - Measurement (1977)
  - Productivity environments (1984)
  - Risk-management planning (1981)
- Requirements engineering practices
  - Change board (1978)
  - Throwaway user interface prototyping (1975)
  - JAD sessions (1985)

Low Hanging Fruit (cont.)

- Design practices
  - Information hiding (1972)
  - Design for change (1979)
- Construction practices
  - Source code control (1980)
  - Incremental integration (1979)
- Quality assurance practices
  - Branch-coverage testing (1979)
  - Inspections (1976)
- Process improvement
  - SW-CMM (1987)
  - Software Engineering Process Groups (1988?)

Where to Start

- Generalities:
  - Requirements, project planning, project tracking, quality assurance, configuration management subcontractor management
  - These are the SEI CMM-SW's Level 2 KPAs
  - Construx's consulting experience bears this out as a generality
- Specifics vary greatly
- Significant results are achievable!

References

Barry W. Boehm, et al, *Software Cost Estimation with Cocomo II*, Prentice Hall, 2000

James Herbsleb, et al, "Benefits of CMM Based Software Process Improvement: Initial Results," Pittsburgh: Software Engineering Institute, Document CMU/SEI-94-TR-13, August 1994.

James Herbsleb, et al. "Software Quality and the Capability Maturity Model," *Communications of the ACM*,June 1997, pp. 30-40.

Humphrey, Watts, 2001. *Winning with Software: An Executive Strategy*, Reading, Mass.: Addison Wesley, 2001.

Johnson, Jim, 1999. "Turning Chaos into Success," Software Magazine, December 1999, pp. 30-39 [Johnson is Chairman of the Standish Group]

Capers Jones, *Assessment and Control of Software Risks*, Prentice Hall, 1994.

Herb Krasner, "Accumulating the Body of Evidence for The Payoff of Software Process Improvement – 1997,"

Dr. Patricia K. Lawlis, Capt. Robert M. Flowe, and Capt. James B. Thordahl. "A Correlational Study of the CMM and Software Development Performance," *Crosstalk*, September 1995.

Standish Group, The, 1994. "Charting the Seas of Information Technology," Dennis, MA: The Standish Group, 1994.

Krasner, Herb, "Accumulating the Body of Evidence for the Payoff of Software Process Improvement – 1997," November 19, 1997. Unpublished paper.

Steve McConnell, *Rapid Development*, Microsoft Press, 1996.

Steve McConnell, *Professional Software Development*, Addison-Wesley, 2004.

Neil C. Olsen, "Survival of the Fastest: Improving Service Velocity," *IEEE Software*, September 1995.

Lawrence H. Putnam, "Linking the QSM Productivity Index with the SEI Maturity Level."

Software Engineering Institute,"Process Maturity Profile of the Software Community 2000 Year End Update," March 2001.

van Solingen, Rini, "The Cost and Benefits of Software Process Improvement," *Proceedings of the Eight European Conference on Information Technology Evaluation, September 17-18, 2001*.

Contact Info

- **Steve McConnell**

Construx Software
11820 Northup Way, Suite E200
Bellevue, WA 98006
(425)636-0100
stevemcc@construx.com