The Art & Science of Software Process

Steven Teleki
Program Director, Software Engineering
Industry SOA Accelerators, IBM Software Group
Past Chair, IEEE Computer Society, Austin Chapter

Why Art & Science ?

When asked why he gave the title, *The Art of Computer Programming*, to his famous series of books, Donald Knuth said:

"Science is what we understand well enough to explain to a computer and art is everything else."

Knuth, Donald. Computer Programming is an Art. Communications of the ACM. December 1974.

What is the Goal of a Process?

Make commitments that you can keep.

Produce software on-time and on-budget.

The process serves to organize the participants of software work to create value.

Paraphrasing Peter Drucker

Drucker, Peter F. The Essential Drucker. Harper Business. New York, NY. 2001.

"We cannot solve the problems that we have created at the level of thinking that we have created them."

Albert Einstein

Agenda

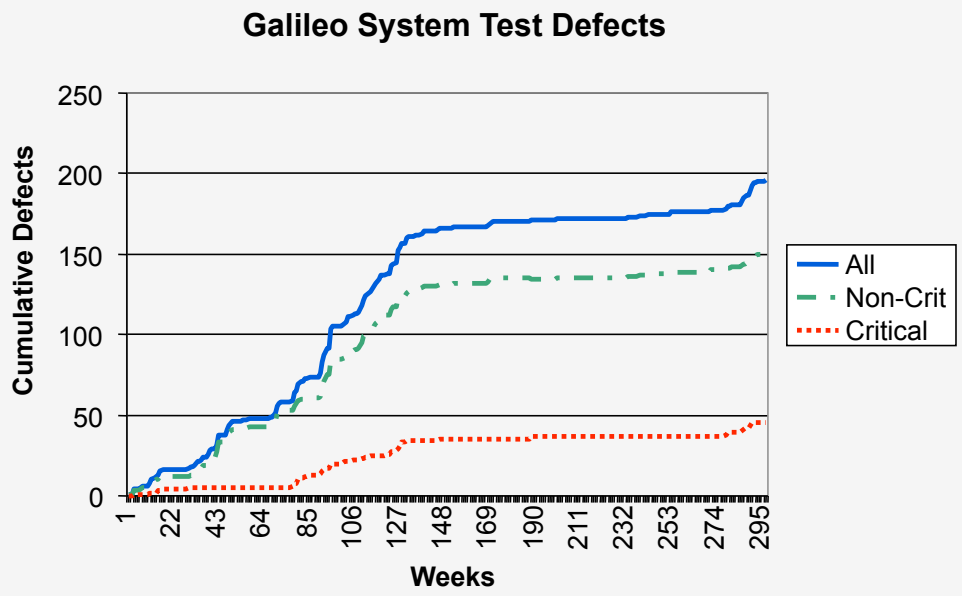
What have practicing developers tried so far?

What is the Art & Science of Software Process?

- ➡ Understand Knowledge Worker Performance
- ➡ Seek Personal Mastery through the Personal Software Process
- ➡ Incorporate Elements of High-Performance Software Development Practice in your work

What Have We Tried?

Tried to get the defects out in testing...



Tried Hiring Young Workers...

"Wanted: Young, skinny, wiry fellows not over 18. Must be expert riders willing to risk death daily. Orphans preferred. Wages $25 per week."

Pony Express advertisement, 1860.

McConnell, Steve. After the Gold Rush. Microsoft Press. 1999.

Tried Offering Better Compensation...

"We realize the skills, intellect and personality we seek are rare, and our compensation plan reflects that. In return we expect TOTAL AND ABSOLUTE COMMITMENT to project success—overcoming all obstacles to create applications on time and within budget."

Software Developer Advertisement, Seattle Times, 1995.

McConnell, Steve. After the Gold Rush. Microsoft Press. 1999.

Developers, Managers, & Gurus just Kept on Trying...

- » PSP/TSP/CMM/CMMI
- » ISO 9001/9000-3
- » FDD (Feature Driven Development) www.featuredrivendevelopment.com
- » Rational Unified Process www.ibm.com
- » SCRUM www.controlchaos.com
- » Extreme Programming www.extremepr
- » OPEN (Object-or and Notation) w

The Winner Is:

Code 'n Fix!

(by a wide margin)

Understand Knowledge Worker Performance

The Lessons of a Long-ago Disaster

What do you know to be important but are unable to measure?

- As of October 1707: **longitude**

Longitude: How far east or west you are?

- Admiral Clowdisley Shovell misjudged longitude.
  - 4 warships and 2,000 lives were lost

Buckingham, Marcus, Curt Coffman. First, Break All The Rules. Simon & Schuster. NY, NY. 1999.

What is Your Software Development Performance?

Have you been thinking about it before?

» Do you know your **"batting average?"**

Software Development Performance is the complexity of all activities that an individual or team does in order to create software.

Figure Out How You Acquire Knowledge

0th Order of Ignorance: Lack of Ignorance.
You know.

1st Order of Ignorance: Lack of knowledge.
You know the question. Uncertainty.

2nd Order of Ignorance: Lack of awareness.
This is a real problem: not only you don't know the answer, you don't even know what the question is.
Ambiguity.

3rd Order of Ignorance: Lack of process.
You don't have a process to find out what it is that you don't know.

4th Order of Ignorance: Meta Ignorance. **You don't know about the orders of ignorance. You are past this.** ☺

Armour, Phillip G. The Five Orders of Ignorance. Comm. of the ACM. Vol.43. No.10. Oct. 2000.

Create Appropriate Learning Environments

Crawl, walk, run!

An accomplished walker doesn't think about the mechanics of the steps anymore.

Learning Dilemma

We learn best from experience but we never directly experience the consequences of many of our most important decisions.

Senge, Peter. The Fifth Discipline. Pg. 23. Currency Doubleday. New York, NY. 1990.

"In every human activity the law of the farm governs."

"There is no cramming on the farm."

Stephen R. Covey

Seek Personal Mastery through the Personal Software Process

Personal Software Process

Personal

- It is *your* process. If there is something that you don't like, then *you* need to change it!

Software

- A personal process applied to software development.

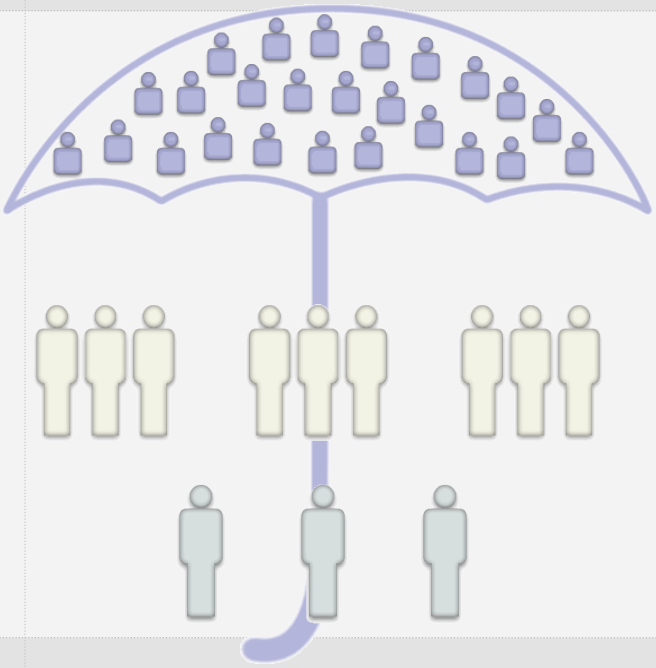
Process

- *"A series of actions, changes, or functions bringing about a result."*

Anybody who creates a deliverable that could have defects can benefit from a personal process.

Humphrey, Watts S. A Discipline for Software Engineering. Addison-Wesley. Reading, MA. 1994.
Excerpted from The American Heritage® Dictionary of the English Language

Team Success Hinges On The Individual



Capability Maturity Model (CMM): Focuses on the organization's capability; management actions.

Team Software Process (TSP): Focuses on team performance; product development.

Personal Software Process (PSP): Focuses on individual skills and discipline; entirely personal.

Focus on Your Talents

What is a **talent?**

"A talent is a recurring pattern of thought, feeling, or behavior that can be productively applied."

Every role performed ***at excellence*** requires talent.

"Michelangelos of housekeeping."

Buckingham, Marcus, Curt Coffman. First, Break All The Rules. Simon & Schuster. NY, NY. 1999.
Peters, Thomas J. The Circle of Innovation. Random House. New York, NY. 1997.

Make Your Non-Talents Irrelevant

"People don't change that much. Don't waste time trying to put in what was left out. Try to draw out what was left in. That is hard enough."
– wisdom from great managers

Team up with people with complimentary talents.

Buckingham, Marcus, Curt Coffman. First, Break All The Rules. Simon & Schuster. NY, NY. 1999.

Your Are In Charge!

Think of yourself as:

Me, Inc.

Even if you happen to be on somebody's payroll at the moment!

Peters, Thomas, J. Brand You 50:Fifty Ways to Transform Yourself from an "Employee" into a Brand that Shouts Distinction, Commitment, and Passion. Knopf/Random House, 1999.

DISTINCT … OR EXTINCT!

"If there is nothing very special about your work, no matter how hard you apply yourself, you won't get noticed and that increasingly means you won't get paid much, either."

Michael Goldhaber, Wired

Incorporate
Elements of High-Performance Software Development Practice in your work

Prototype, Prototype, Prototype

"At Sony the mean time to prototype is an astonishing five days. Competitors take several months, at best, to do the same."

Peters, Thomas J. The Circle of Innovation. Random House. New York, NY. 1997.

Separate Research from Development

Research

- Inventing something new, that has never existed.
- It can only be time limited.

Development

- Use existing technology, or implement an invention.
- Can be planned & scheduled; it has been done before.

Library research and *learning* can be planned.

Define and Capture Context

What is Context?

» Everything that is said, done, drawn, or written during the software development process.

How much Context do you need?

» Just enough to always know where you are with the work and to know what to do next.

Plan Your Work

Why?

» The plan is the basis of commitments.

» To be successful you must be able to make commitments that you can meet—**at a profit**.

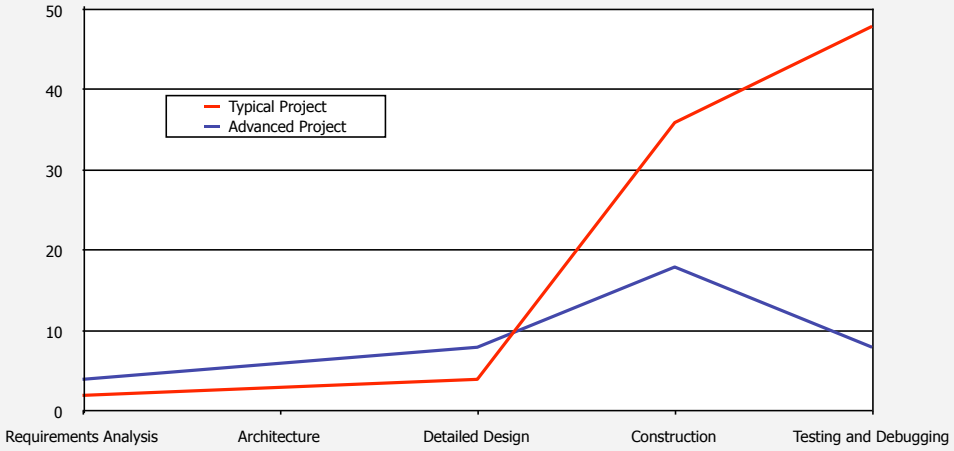
What is a plan?

» It is the amount of work that needs to be done to achieve the desired outcome.

How?

» Plan in detail. Task length: 45-90 minutes.

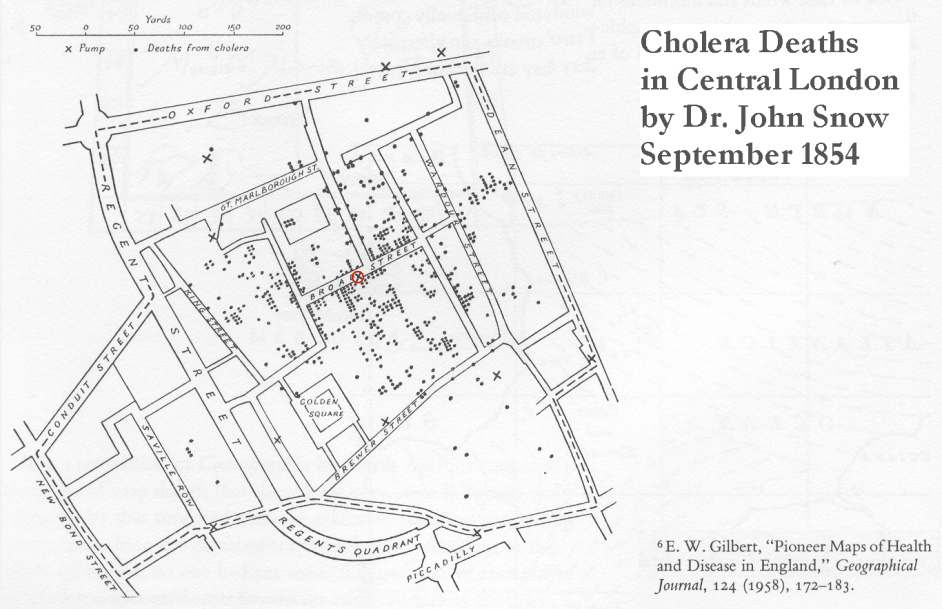
Differentiate Work Phases



Advanced software development approaches require more work during the early stages of the project to eliminate enormous amount of unnecessary work in the later stages of a project.

McConnell, Steve. After the Gold Rush. Microsoft Press. 1999.

Represent Your Data to Make the Most Sense of It



[6] E. W. Gilbert, "Pioneer Maps of Health and Disease in England," *Geographical Journal*, 124 (1958), 172–183.

Tufte, Edward R. The Visual Display of Quantitative Information. Graphics Press. 1983.

Project Data—Plan vs. Actual

	Hours	% Over	Weeks	% Over
Plan 1	118	211	2	50
Plan 2	244	50	3	0
Actual	367		3	

Measurement is only meaningful on a defined process or plan.

Track Effective On-Task Time (EOT)

The time effectively spent on project work.

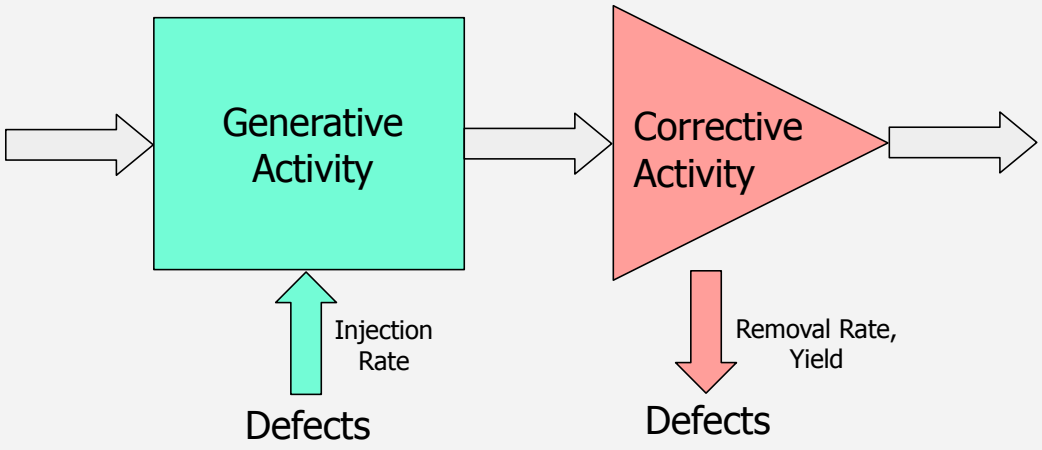
Doesn't include:

- Reading email (usually even if it is project related)
- Meetings (except well-defined project meetings)
- Lunch time, breaks, phone conversations, etc.

Measure your EOT per week.

- Best organizations in the world get 20+ hrs/week.
- You may only get about 3-5 hrs/wk the first week. You should get up to 15 hrs/wk in a few weeks.

Use the Process Building Block



Plan for Quality Results

You must change your process to change your results!

- ✓*What is insanity? Doing the same thing over and over and expecting a different result!*

You know that you will put the defects in, might as well plan to remove them.

Understand what it really *takes to do things!*

Conclusion

"You can be sure our plan was perfect. It's just our assumptions were wrong."

Ken Olsen
Founder & CEO
DEC (for 35 years)
1991

Remember…

- ✓ If you want different results, you must change the way you act and think.
- ✓ The shift from manual work to knowledge work changes the game: think talent.
- ✓ You need to know your own performance.
- ✓ **Brand out from the crowd.**

"Expose yourself to the best things humans have done and then try to bring those things into what you are doing."

Steve Jobs
President & CEO
Apple, Inc.

"If things seem under control, you are just not going fast enough!"

Mario Andretti

race car driver

Your Letters Are Welcome!

Steven Teleki:

- » teleki@computer.org

Visit: http://steven.teleki.net/

- » Software Development Reading List
- » Slides from the talk

