The Art & Science of Software Process

Steven Teleki
Vice President, Software Engineering, Yash & Lujan Consulting, Inc.
Chairman, IEEE Computer Society, Austin Chapter

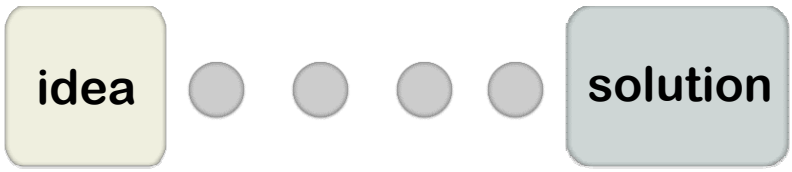
Why *Art & Science*?

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 a Software Process?

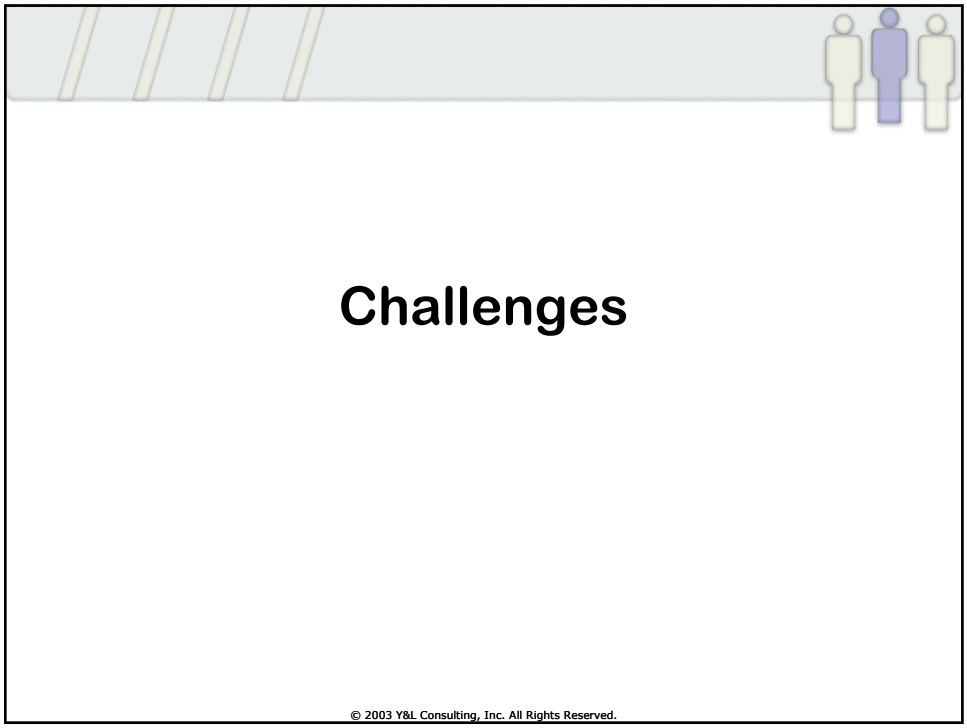


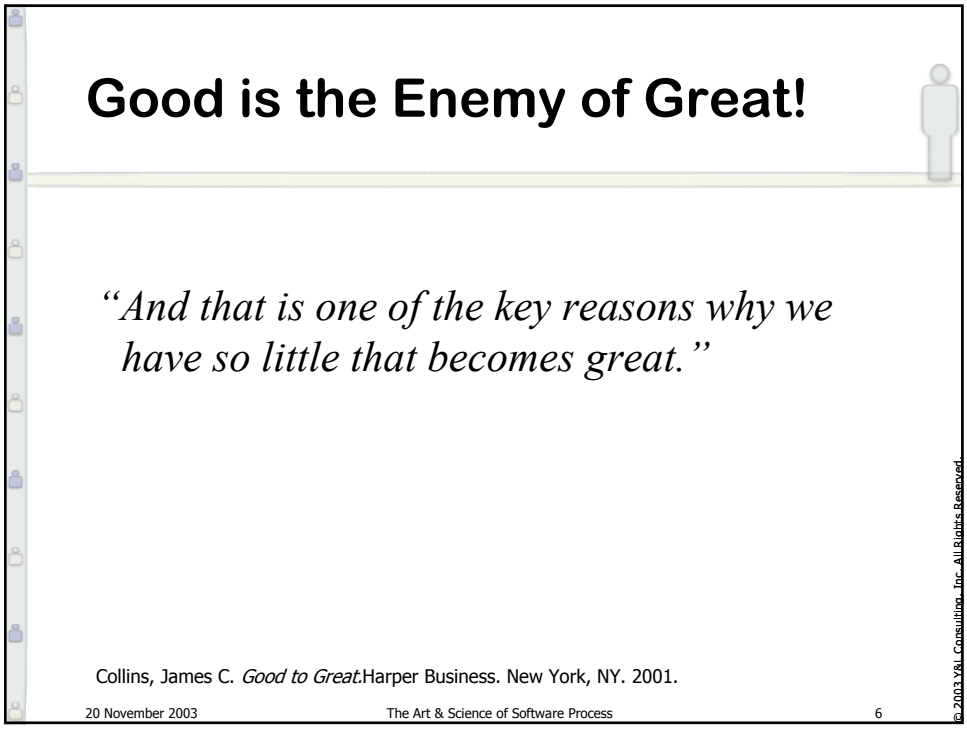
- It starts with a *business need*–an idea.
- It ends with a *solution* that meets the stated need.
- It covers ***all*** activities from start to end.

What is the Goal of the Process?

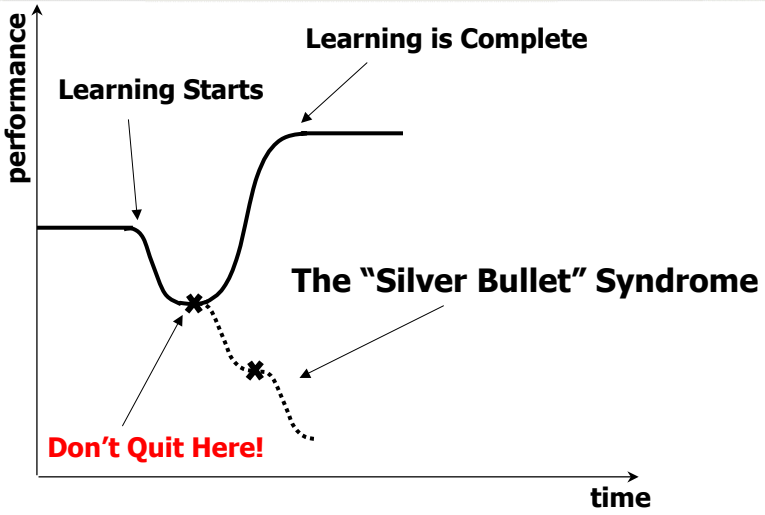
- Make commitments that you can keep.
- *Produce quality software on-time and on-budget.*
  - To paraphrase Peter Drucker:
    The process serves to organize the participants of software work to create value.

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


Challenges
© 2003 Y&L Consulting, Inc. All Rights Reserved.


Good is the Enemy of Great!
"And that is one of the key reasons why we have so little that becomes great."
Collins, James C. Good to Great. Harper Business. New York, NY. 2001.
20 November 2003
The Art & Science of Software Process
6
© 2003 Y&L Consulting, Inc. All Rights Reserved.

Learning Takes Time!



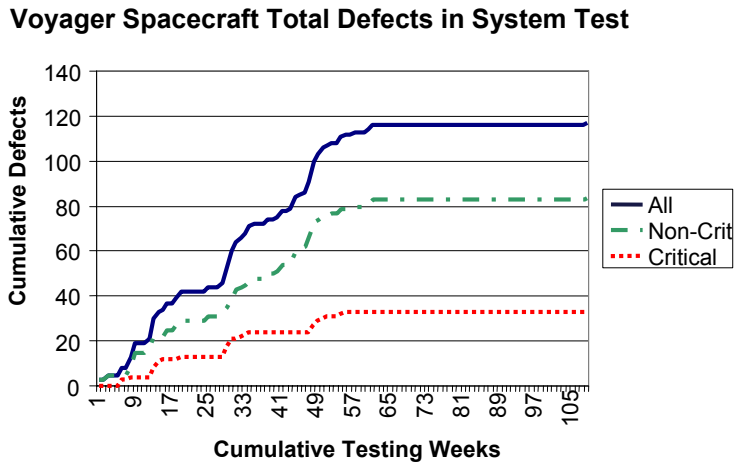
Is Learning Difficult?

- 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.

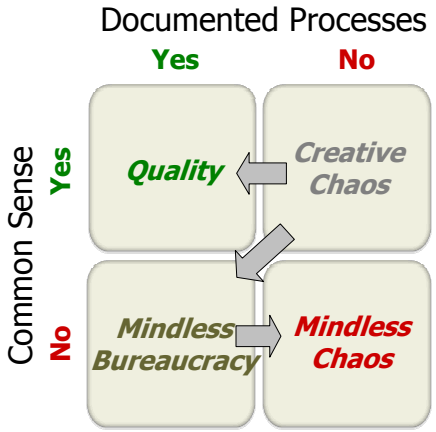
Defect Removal Pattern



Simple ROI Calculation

Current	Averages		Example	
			20,000	LOC
	5	defects/KLOC	100	defects
	10	hours/defect	1,000	hours
	125	$/hour	125,000	$
	10	LOC/hr	2,000	hours
	20	hours/week	100	weeks/dev
	4	developers	25	weeks/team
Future	**Averages**		**Example**	
			20,000	LOC
	1	defects/KLOC	20	defects
	10	hours/defect	200	hours
	125	$/hour	25,000	$
	10	LOC/hr	2,000	hours
	20	hours/week	100	weeks/dev
	4	developers	25	weeks/team
Savings			800	hours
			100,000	$

How Can We Get Quality?



From Mark Paulk, with thanks to Sanjiv Ahuja, President and COO of Bellcore.

Execution vs. Enactment

- Execution: carrying out a process without much thinking or judgment.
  "Unencumbered by the thought process."
  - A computer executes a program.
- Enactment: carrying out a process with understanding of each step and using the process as a guide.
  "If the map and the terrain don't match, trust the terrain."

Thanks to Click & Clack, *The CarTalk Guys* on National Public Radio.

Everything Seems Crazy at First!

"We should do something when people say it is crazy. If people say something is 'good,' it means someone else is already doing it."

» Hajime Mitarai, president, Canon

Peters, Thomas J. *The Circle of Innovation, You Can't Shrink Your Way To Greatness*. Vintage Books. New York, NY, 1997.

Organizational Expectations

"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.

What Changed In 140+ Years?

"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.

A Disaster off the Scilly Isles

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

- Longitude: How far east or west you are?
- Guessing average speed, or dropping a log over the side of the boat and measuring time of travel from bow to stern.
- October of 1707, Admiral Clowdisley Shovell
- 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.*
- An **understanding** of your performance is the basis of making good estimates.

The Stockdale Paradox

Retain faith that you will prevail in the end, regardless of the difficulties.

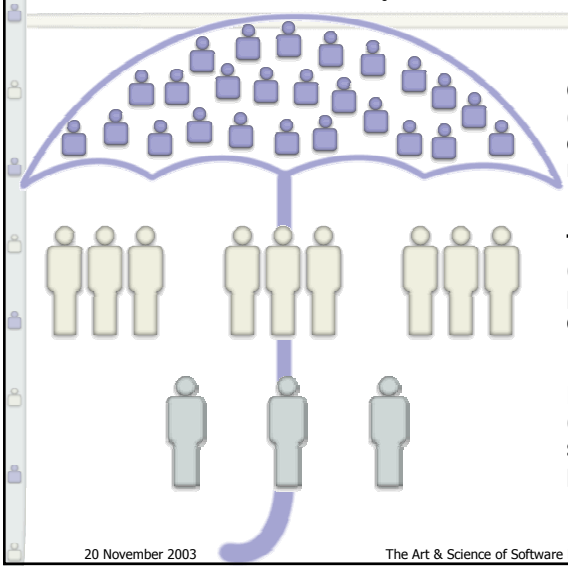
AND at the same time

Confront the most brutal facts of your current reality, whatever they might be.

Collins, Jim. *Good to Great.* Harper Business. NY, NY. 2001.

Approaches

A Comprehensive Approach to Process Improvement



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

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

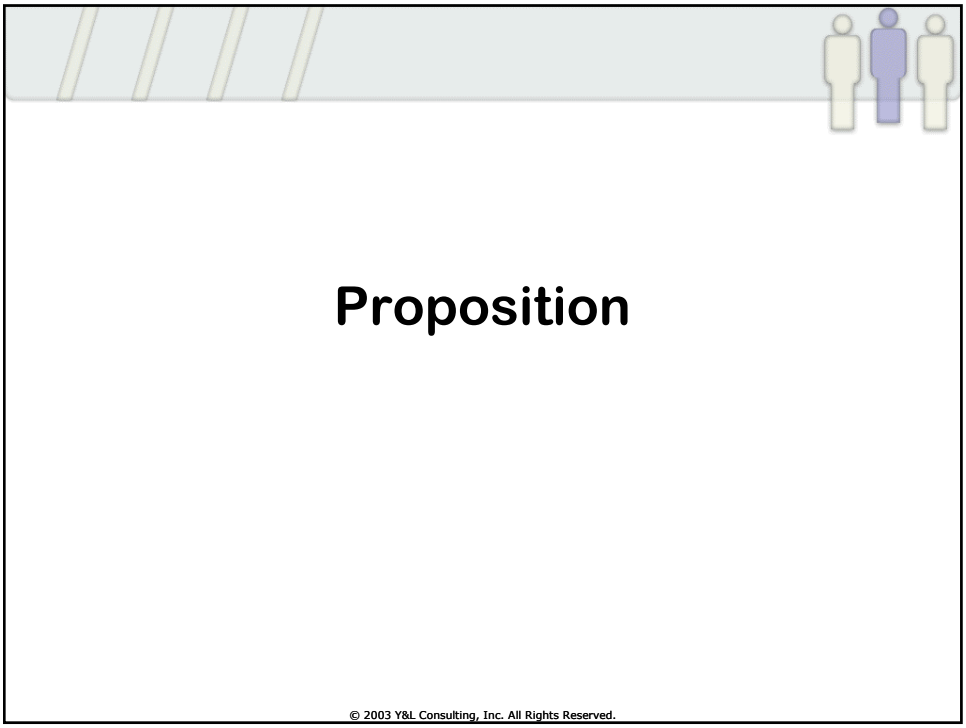
Personal Software Process (PSP): Focuses on individual skills and discipline; entirely personal.

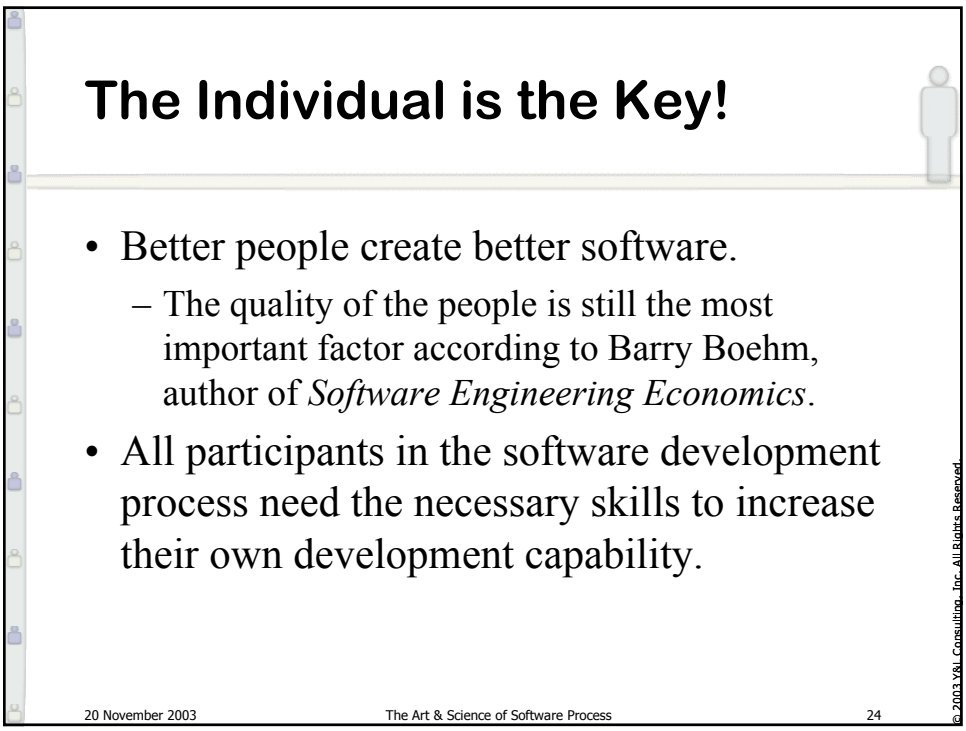
Other Approaches

- ISO 9001/9000-3
- FDD (Feature Driven Development) www.featuredrivendevelopment.com
- Rational Unified Process www.rational.com
- SCRUM www.controlchaos.com

More Approaches

- Extreme Programming www.extremeprogramming.org
- OPEN (Object-oriented Process, Environment, and Notation) www.open.org.au
- **Code 'n Fix ☺**


Proposition
© 2003 Y&L Consulting, Inc. All Rights Reserved.


The Individual is the Key!
• Better people create better software.
– The quality of the people is still the most important factor according to Barry Boehm, author of *Software Engineering Economics*.
• All participants in the software development process need the necessary skills to increase their own development capability.
20 November 2003
The Art & Science of Software Process
24
© 2003 Y&L Consulting, Inc. All Rights Reserved.

Personal Mastery (Personal Process)

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

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*

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*

LOW PRODUCTIVITY KILLS!

"However low its wages, a business […] is unlikely to survive, let alone prosper, unless it measures up to the standards set by the leaders in its field, anyplace in the world."

Peter F. Drucker

Why Focus on Yourself?

- You are special!
- You are the same person at home & at work.
- 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.

Work To 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 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

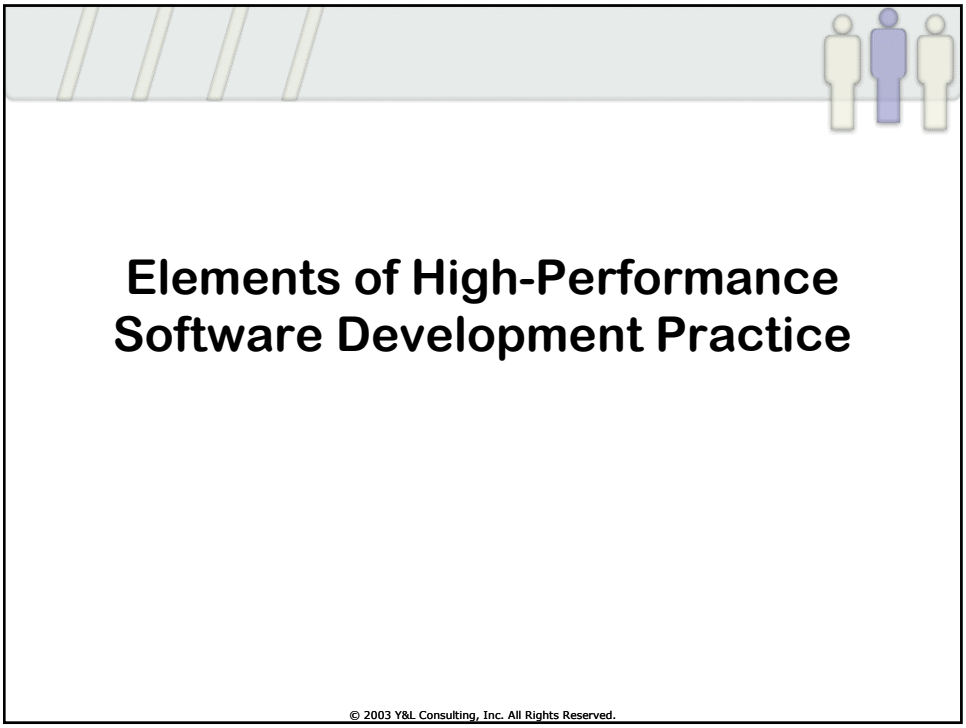
- Team up people with complimentary talents.

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

Why Become Great?

- It is no harder to be great then to be mediocre. It takes **clarity & focus**.
- In the search of meaning when you find it, you will become great.
- For its own sake.

Collins, James C. *Good to Great.* Harper Business. New York, NY. 2001.



Defined Process

- A process is defined if it is:
  - Written down;
  - Has enough detail that it can be enacted repeatedly producing the same or very similar outcome.
- A process must be defined for any measurement to be meaningful.

Benefits of a Defined Process

- A defined process enables you to understand, monitor, and improve your performance.
  - Record data on your process.
  - Analyze the collected data.
- Use the data to improve what you do!

Planning

- 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.

Prototyping Culture

"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.

Research vs. 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.

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.

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.

Estimation

- Size (e.g. KLOC for code): estimate only this!
  - Calculate time, schedule, & defects based on size.
- Time (project hours)
  - Calculate time based on past productivity data.
  - Estimate productivity if past data is not available.
- Schedule (map project hours to calendar days)
  - Schedule is the time available for project work.
- Defects (e.g. Defects / KLOC)
  - Estimate defects using past defect injection data.

Quality Planning

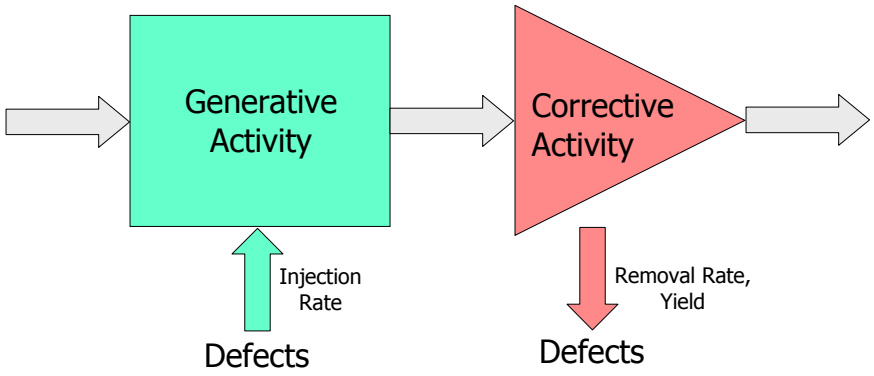
- You must change your process to change your results!
  - *What is insanity? Doing the same thing over and over and expecting a different result!*
- You know that you will put the defects in, might as well plan to remove them.
- Calculate the ROI of all activities!

Broken Windows & Software?

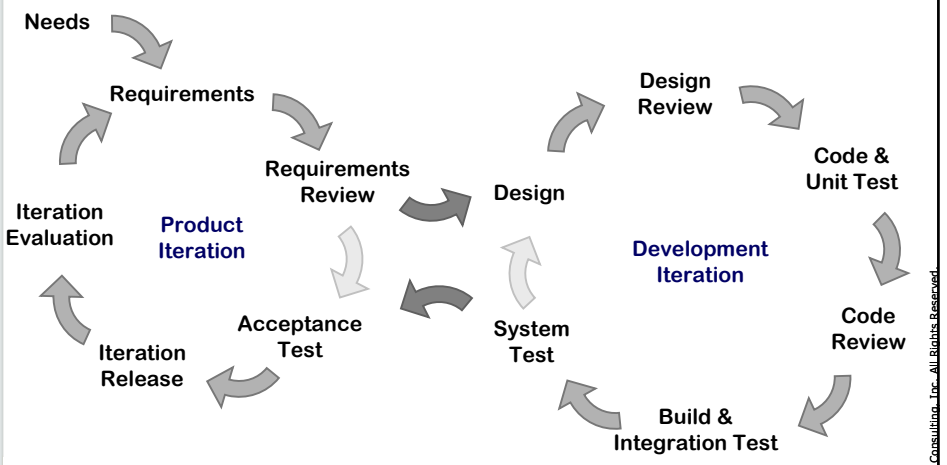
- The brainchild of criminologist James Q. Wilson and George Kelling.
  - Crime is the inevitable result of disorder.
  - If one window broken, soon more will be broken.
- Applies to software equally well…
  - One defect is followed by other defects.
  - NOT Microsoft Windows.

Gladwell, Malcolm. *The Tipping Point: How Little Things Can Make A Big Difference*. Pg. 141. Little, Brown, and Company. New York, NY 2000.

Process Building Block

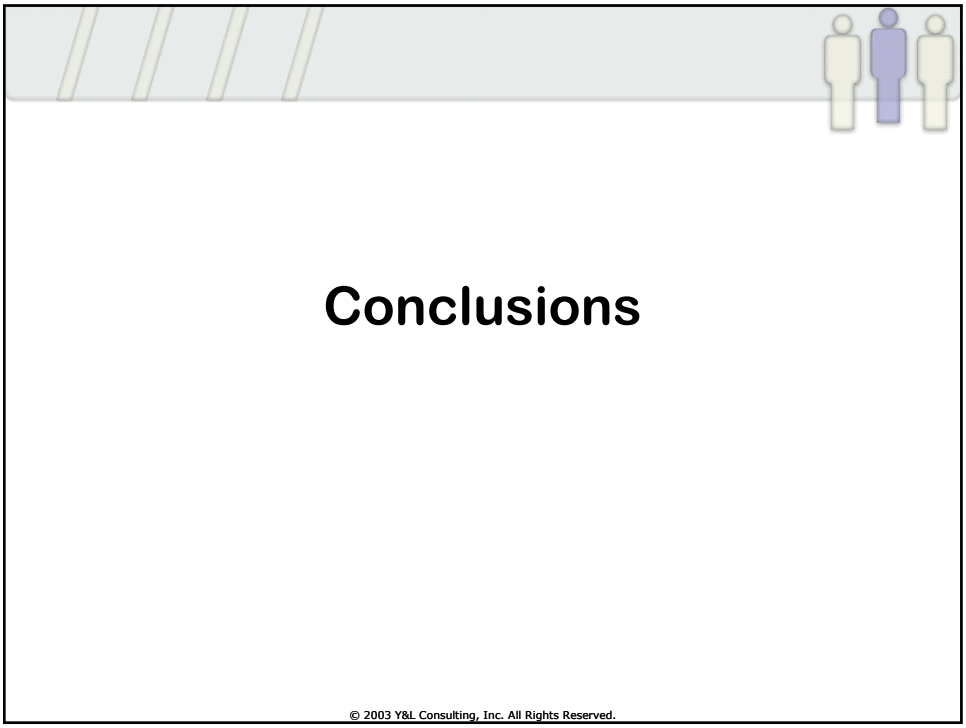


Smooth Iterations



Data Analysis

- Collect data for a reason! If you never look at the data you collected, then don't collect it!
- Data can tell you:
  - Where your time goes?
  - What did you really work on?
  - What was forgotten from the plan?
  - What was extra?
  - Where do you excel?
  - Where can you improve?
  - … and many more things!



Is it *Art & Science*?

- It is both *Art & Science* and it will be.
- As we understand more about:
  - software development,
  - people,
  - processes,
  - relationship to other domains

…we will always evolve part of the *Art* into *Science* and we will continually discover new *Art* that we need to master...

Ongoing Process Improvement

- You need to **know** and understand what your process **is** before you can improve it!
- In workplaces where people **understand** their process and follow it, they invariably think of ways to improve what they do.

- *Improvement isn't possible if your process doesn't change; "working hard" doesn't work.*

Your Theories Lead You

*The way you work depends on your **thinking**!*

- You live with your personal (software) process.
- Getting another personal process, means you have to change the way you *think* and *work*.
- You should know your performance!
- It is possible to create defect free code.

Closing Quote

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

—Mario Andretti, race-car driver

Thank You!

Contact Information

Steven Teleki, Y&L Consulting, Inc.
1431 Greenway Drive, Suite 830
Irving TX 75039

teleki@acm.org or teleki@ylconsulting.com

For a software development reading list please visit:
http://pseng.net/reading/