

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

The Art & Science of Software Process

Steven Teleki
Director, Software Development, Webify Solutions, Inc.
Past Chairman, IEEE Computer Society, Austin Chapter

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 1

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 2

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

The World As We Know It...

Look around you, from:

- » mobile phones to thermostats,
- » cars to toothbrushes,

just about every device has a computer in it.

But,... our methods to create the software are not keeping up with the increasing demand.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 3

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

***“We cannot solve the problems
that we have created
at the level of thinking
that we have created them.”***

Albert Einstein

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 4

StevenTeleki The Art & Science of Software Process
Home | Situation | **Complication** | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 5

StevenTeleki The Art & Science of Software Process
Home | Situation | **Complication** | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 6

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

What Have We Tried?

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 7

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Tried to get the defects out in testing...

Voyager Spacecraft Total Defects in System Test

Cumulative Testing Weeks	All	Non-Crit	Critical
1	0	0	0
9	10	5	2
17	35	20	10
25	45	30	15
33	70	45	25
41	80	55	25
49	105	75	30
57	115	80	35
65	118	85	35
73	118	85	35
81	118	85	35
89	118	85	35
97	118	85	35
105	118	85	35

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 8

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Tried adding new features while fixing defects...

Days	Enhancement Gap	Defect Gap
1	5	2
3	8	3
5	10	4
7	12	5
9	15	6
11	18	7
13	22	8
15	25	9
17	28	10
19	32	11
21	35	12
23	40	15
25	45	20
27	50	25
29	55	35
31	65	50
33	80	75
35	100	110
37	120	150
39	140	200

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 9

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 10

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 11

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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

- » PSP/TSP/CMM
- » ISO 9001/9000-3
- » FDD (Feature Driven Development)
www.featuredrivendevelopment.com
- » Rational Unified Process www.rational.com
- » SCRUM www.controlchaos.com
- » Extreme Programming www.extremeprogramming.org
- » OPEN (Object-oriented Process, Environment, and Notation) www.open.org.au
- » **Code 'n Fix ☺**

WINNER !
(by a wide margin!)

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 12

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 13

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Understand Knowledge Worker Performance

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 14

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 15

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

What is Your Software Development Performance?

Have you been thinking about it before?

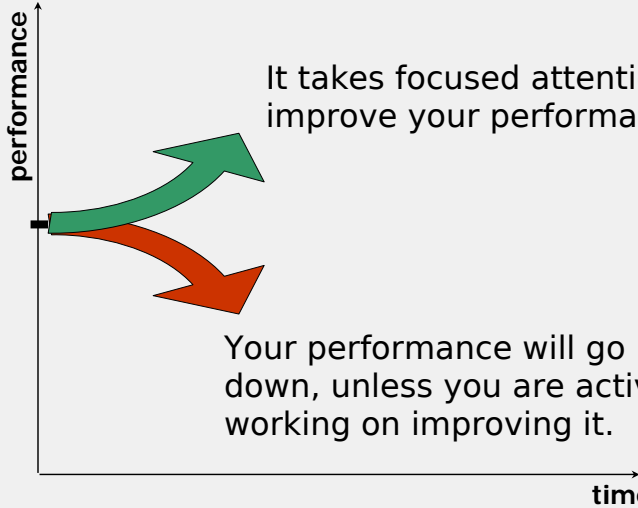
- » Do you know your "batting average?"

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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 16

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Wake up: There is NO Constant Performance!



It takes focused attention to improve your performance.

Your performance will go down, unless you are actively working on improving it.

performance

time

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 17

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 18

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 19

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Heed the Age-old Wisdom

"In every human activity, the Law of the Farm governs."

"There is no cramming on the farm."

Covey, Stephen R. *First Things First*. Free Press. 1996.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 20

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Seek
Personal Mastery through
the Personal **Software** Process

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 21

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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*

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 22

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Stand out from the crowd!

“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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 23

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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*

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 24

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 25

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 26

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 27

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 28

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 29

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 30

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 31

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 32

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Differentiate Work Phases

Relative Effort on a Typical vs. an Advanced Project

The graph shows relative effort on the y-axis and project phases on the x-axis. The x-axis categories are Requirements Analysis, Architecture, Detailed Design, Construction, and Testing and Debugging. The 'Typical Project' (red line) starts low, remains relatively flat through Architecture and Detailed Design, then rises sharply during Construction and continues to rise through Testing and Debugging. The 'Advanced Project' (blue line) starts slightly higher than the typical project, rises through Architecture and Detailed Design, peaks during Construction, and then declines during Testing and Debugging.

Project Phase	Typical Project (Relative Effort)	Advanced Project (Relative Effort)
Requirements Analysis	Low	Low-Mid
Architecture	Low-Mid	Mid
Detailed Design	Mid	Mid-High
Construction	High	Peak
Testing and Debugging	Very High	Declining

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 33

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Define Your Processes

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 34

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 35

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 36

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

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!

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 37

StevenTeleki The Art & Science of Software Process
 Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Use the Process Building Block

Defects

Defects

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 38

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Conclusion

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 39

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Embrace 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.

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 40

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | 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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 41

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Become Familiar with Other Domains

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

—Steve Jobs, President, Apple Computer

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 42

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Remember...

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

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 43

StevenTeleki The Art & Science of Software Process
Home | Situation | Complication | What Have We Tried? | Knowledge Worker | Personal Mastery | High-Performance | Conclusion

Want to be Agile?

knowledge
+ understanding

= speed

2 May 2005 © 2002-2005 Steven Teleki. All rights reserved. 44

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

Mario Andretti
race car driver

Your Letters Are Welcome!

Steven Teleki
1605 Amelia Drive
Cedar Park, TX 78613

teleki@computer.org

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