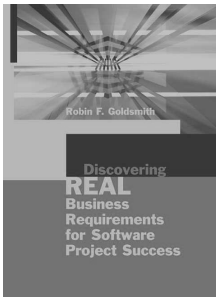
7 Low-Overhead Software Process Improvement Methods



Robin F. Goldsmith, JD
GO PRO MANAGEMENT, INC.
SYSTEM ACQUISITION & DEVELOPMENT
QUALITY/TESTING
PRODUCTIVITY
BUSINESS ENGINEERING
TRAINING
22 CYNTHIA ROAD
NEEDHAM, MA 02494-1412
INFO@GOPROMANAGEMENT.COM
WWW.GOPROMANAGEMENT.COM
(781) 444-5753

What's It Take to Win a 600-Mile Sydney-Melbourne Ultra Marathon?

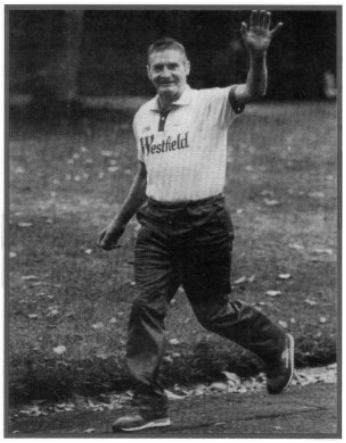
Special shoes? (swoosh?)

Smooth stride?

Youth?

How About a 61-Year-Old Shuffling in Rubber Gumboot Galoshes?

Cliff Young
c. 1997



What's the message for process improvement?

Evolution

Revolution

Objectives

- Differentiate process improvement from any particular formally-defined improvement approach
- Describe how to identify the *REAL* (as opposed to presumed) software process and apply proven techniques for improving it consistently (evolution) and creatively (revolution)
- Suggest some "good" practices that often achieve marked improvements without high overhead

People Often Equate "Process" with CMM and/or Mandated Busywork

"We need process…"

"We don't have time for process…"

"We don't need no stinkin' process…"

Process and process improvement existed long before CMM, ISO, Six Sigma, etc.

CMM Assessors Recognize It Too

"The emphasis has often been on passing various 'tests,' such as the ISO certification and SEI CMM levels. If you are taking one of these routes, you may be wondering if you are actually improving anything--or just learning to jump through new hoops for the purpose of an audit. Process improvement doesn't have to be academic, or solely focused on documentation. It can, and should, be used to solve real problems and make real gains."

Neil Potter and Mary Sakry
"Measuring Process Improvement"
Software Testing & Quality Engineering
November/December 2000

"Many—perhaps most—software process improvement efforts fail"
Mark Paulk, Keynote
International Conference on Software Process Improvement, November 2002

Common Approaches to Software Process Improvement (SPI)

Overhead

- Big initiatives, formal procedures and falderal
  - Process imposition (assumes "correct" ways), e.g., CMM, TQM, and Six Sigma
  - More process acceptance, e.g., ISO 9000, TickIT
- Apply proven process improvement methods to software--identify and address one's own process
- Implement specific "good" practices (Pareto 80-20 approach)

① *Generic* **Real** *Process Improvement*

Presumed
Defined, documented *Different*

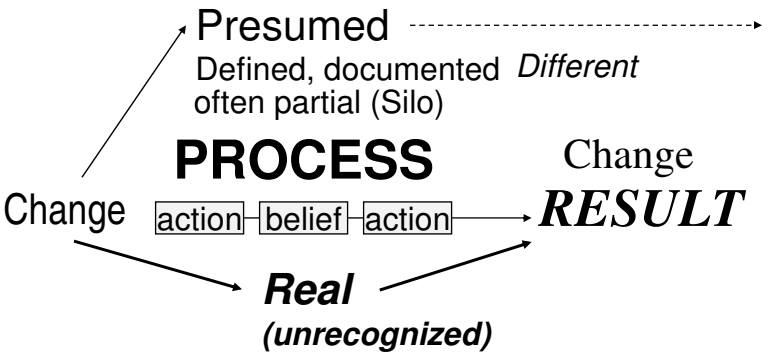
PROCESS

action – belief – action → ***RESULT***

Real
(unrecognized)

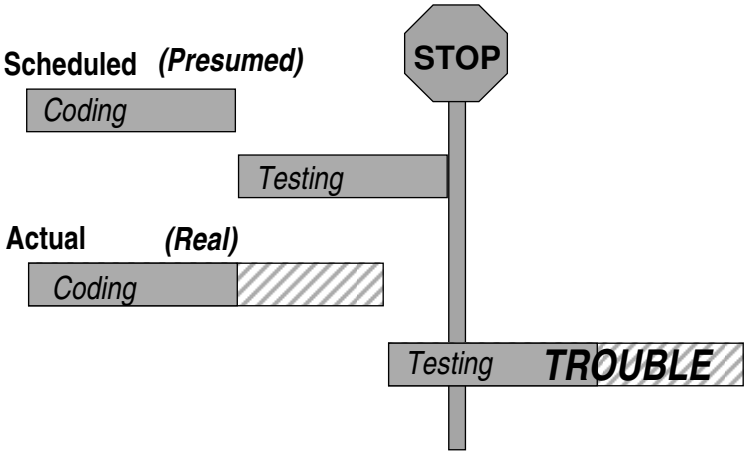
A result is the inevitable outcome of the process followed, regardless of whether it is intended, desired, or even recognized
Knowing your process enables predicting your results

To Change Results, Change Process



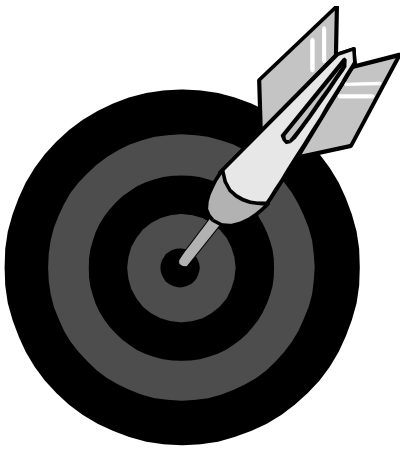
To see the need for and effects of changes, we need to measure at key process points from the beginning to the full end result.

Typical Testing Experience



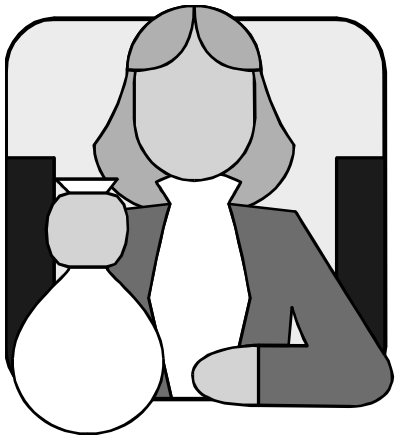
Effective Managers Focus on Results Ineffective Ones Focus on Activities

- Include result measures in process, so they are taken routinely
- Map and measure all the actions/factors on the way to the result
- Remove or replace
  - No/low value-added
  - Error-prone, delays
  - Resource intensive



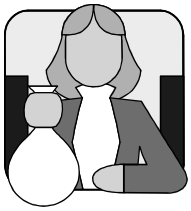
How to Tell the **REAL** *Process Bank Example*

- Sought project management training
- Presenting problem: "Too many Change of Scope requests"
- Presumed problem: Each request requires high executive sign-off, considered a bad thing



How to Tell the **REAL** *Process Bank Example (cont.)*

- ✓ Virtually every project had a Change of Scope request
- ✓ Virtually every Change of Scope request was approved
- ✓ Change of Scope requests incurred no actual negative consequences



So, what's their "real" process?

People sensed intuitively that "they" wouldn't support the time, effort, accountability to manage effectively

② *"Soft Side" of* **Real** *Process*

- Cultural beliefs
  - The way we do things around here
  - What can and cannot be said/done or it's your head
- Management practices
  - Hiring, firing, promoting--what's really rewarded
  - Understanding of the work
  - Walk the talk commitment to quality
- Execution skill, diligence, and improvement

How to Tell the **REAL** *Process Outsourcing Example*

- Sought assistance learning how to "manage outsourcer's performance"
- Presenting problem: Outsourcer failed to deliver, may even have gone out of business entirely
- Presumed problem: Picked the wrong outsourcer

How to Tell the **REAL** *Process Outsourcing Example (cont.)*

- ✓ Engaged outsourcer to build a system
- ✓ Outsourcer failed to deliver after several years
- ✓ Blamed the outsourcer
- ✓ Engaged a different outsourcer to build the system
- ✓ New outsourcer failed too
- ✓ Several other outsourcings also failed

So what's their **REAL** *process?*

The **REAL** *Process Outsourcing Example (cont.)*

- ➔ Client executive assumes he knows what is needed and picks vendor personally, failing to adequately define business requirements, especially how to do it "our way"
- ➔ Outsourcer doesn't know what to deliver
- ➔ Constant pressure, unclear changes, and bickering about "not our way" until outsourcer gives up
- ➔ Client blames outsourcer [*repeat process*]
- ➔ Executive attends 2 hours of 2-day software acquisition training, gets promoted; dedicates group to acquisitions, apply training's formats but don't develop content skills

③ *"Good" Practice: Training*

Essential for meaningful improvement, but too often:

- Effectiveness of training is measured by survey (popularity poll) or test (maybe worse?)
- Manager doesn't know the lessons of course or recognize applicability to him/her
- Nobody is accountable for using the lessons, so they aren't used; and instructor is blamed

Wrong time, wrong folks—"They won't let us use it"

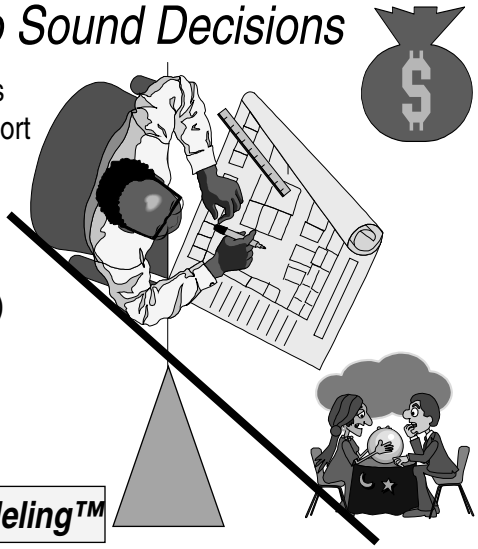
④ *"Good" Practice: Measurement and Estimation*

- Techniques like Function Points add repeatability
- Tools backed by project databases address needed variables
- Initiatives tend to die *("I've got 5000 Function Points, now what?")*

Presumed: Return on Investment (ROI) Up-Front Leads to Sound Decisions

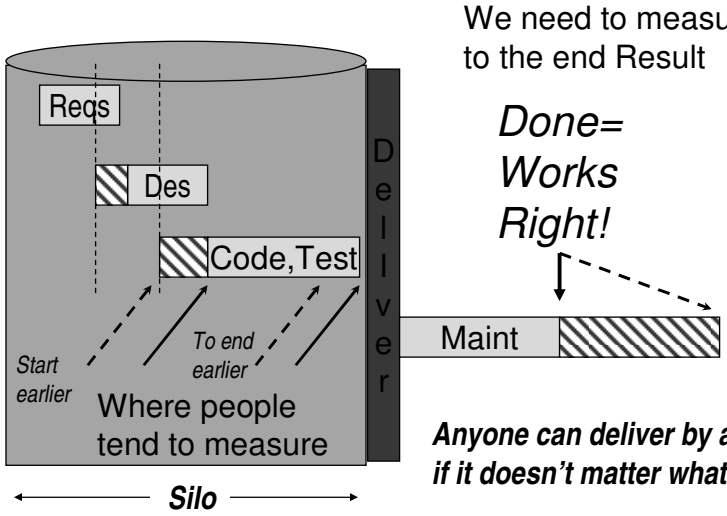
REAL

- Seldom matched to actuals
- Often "justification" to support prior decision
- Costs are underestimated
- Benefits are overestimated
- Inadequate definition of REAL requirements (value) or linkage of solution to meeting requirements
- Extensive calculations for tangibles get trumped by unquantified intangibles



Consider ROI Value Modeling™

The "We don't have time" Fallacies

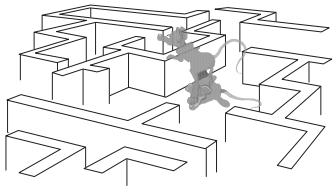


Anyone can deliver by a deadline if it doesn't matter what they deliver

⑤ *"Good" Practice: Formal Technical Reviews, Especially Inspections*

- The single technique most identified with CMM
- Authorities agree is the single most effective economical way to find errors
- Yet, seldom used and often fails of own weight and cultural resistance

Inspection and Walkthroughs Are Both Group Technical Reviews



- Walkthroughs are step-by-step simulations
  - Usually at reading out loud pace
  - Often informal (weaker)
- Inspections are guided by a common errors list
  - Often share findings from individual preparation at silent reading pace (so cover more)
  - Usually formal (stronger)

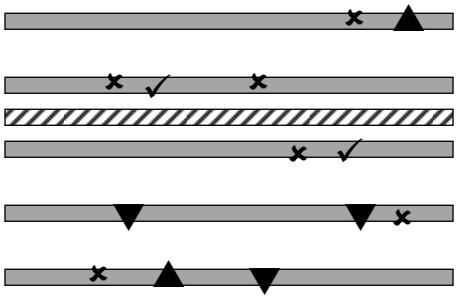
Which Takes Longer?

Testing
Debugging ✓

Finding ✓
Fixing

Which Takes Longer?

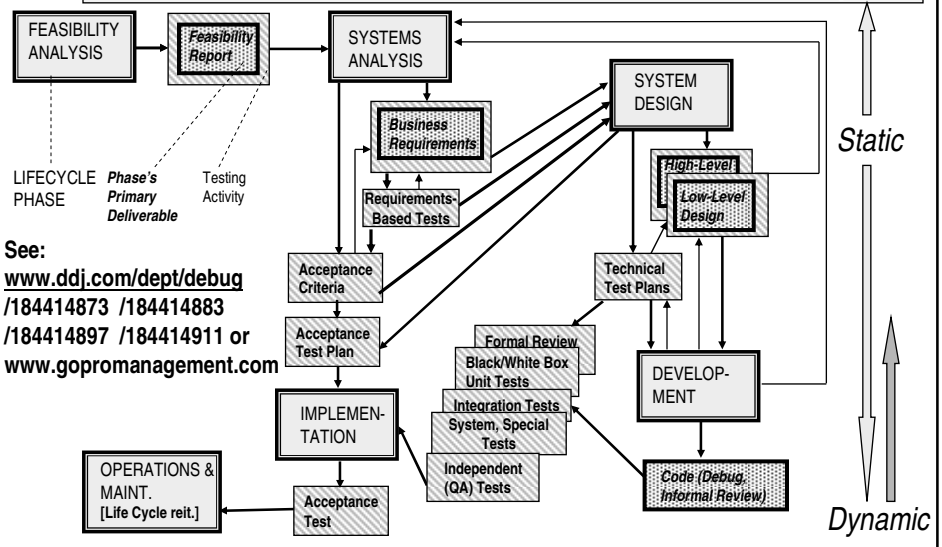
Testing
Debugging ✓ Finding ✓ *How long to find?*
Fixing



Reviews find in one pass, no start-stop:

- × Several at a time, not onesy-twosy
- ✓ Selective perception
- ▼ Issues list prompts to see problems
- ▲ And be more acute
- ▨ Find omissions

⑥ "Good" Practice: Proactive Testing™



See:
www.ddj.com/dept/debug
/184414873 /184414883
/184414897 /184414911 or
www.gopromanagement.com

Key Proactive Testing™ Concepts

Find WIIFMs

- Intertwine testing with each development deliverable
- Plan before acting at any point/level
  - Define acceptance criteria at start, keep independent
  - Prepare test plans/designs during Design, promote reuse
  - Prioritize by level, from full choices, to avoid wasted effort
- Let testing drive development to
  - Enable developers to find more problems themselves
  - Selectively do important testing early to prevent rework
  - CAT-Scan™ to double-check and double-check

⑦ *"Good" Practice: Requirements*

- System can't be better than its requirements
- 10-100-1000 to 1 benefit of getting requirements right early
- Industry still hasn't learned how to discover
  - Over-relies on Use Cases
  - Continues to blame users
  - Gives inadequate time

Testers Need Requirements, Right?

Problem: Users don't give technical people their time for defining requirements.

Cause: Lack of user education about why their time and involvement is needed.

We've had this problem for perhaps 50 years. Maybe we're trying to fix the wrong (presumed) cause.

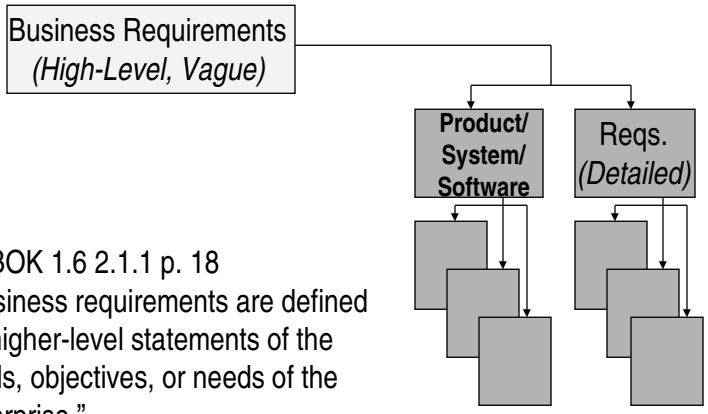
Could there be a different, REAL, cause? What?

For instance: Maybe what IT people think of as user requirements are not perceived that way by users and thus are not something they value spending time on.

➲Two Types of Requirements:

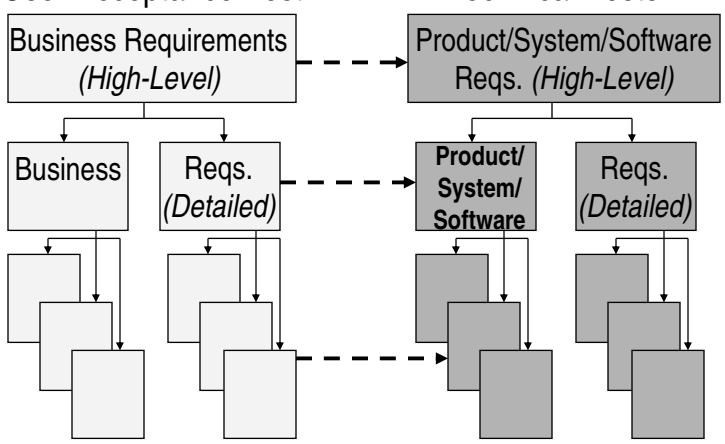
Business/User	Product/System/Software
• Business/user language & view, conceptual; *exists* within the business environment • Serves business objectives • ***What*** business results must be delivered to solve a business need (problem, opportunity, or challenge) and provide value when delivered/satisfied/met **Many possible ways to accomplish**	• Language & view of a *human-defined product/system* • **One of the possible ways *How*** (design) presumably to accomplish the presumed business requirements • Often phrased in terms of external functions each piece of the product/system must perform to work as designed (Functional Specifications)

Even Requirements "Experts" Think the Difference is Detail

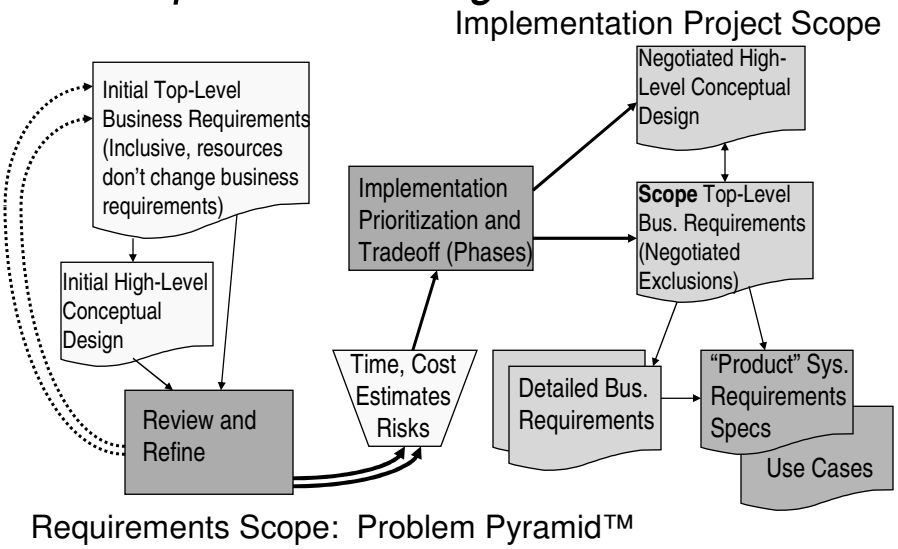


BABOK 1.6 2.1.1 p. 18
"Business requirements are defined as higher-level statements of the goals, objectives, or needs of the enterprise."

When Business/User Requirements Are Detailed First, Creep Is Reduced



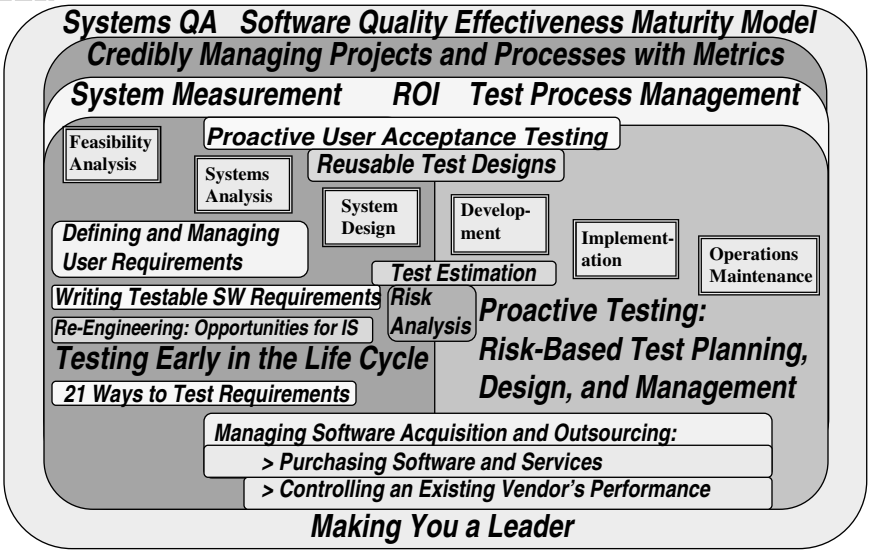
Requirements Negotiation



Summary

- The terms "process" and "process improvement" pre-date and are not exclusive or equivalent to any particular improvement approach
- For reliably consistent and creative improvement, we must identify and address the *real* (rather than presumed) software process
- Merely instituting "good" practices often achieves improvements without excessive overhead

Go Pro Management, Inc. Seminars--Relation to Life Cycle



Robin F. Goldsmith, JD

robin@gopromanagement.com (781) 444-5753

www.gopromanagement.com

- President of Go Pro Management, Inc. consultancy since 1982, working directly with and training professionals in business engineering, requirements analysis, software acquisition, project management, quality and testing.
- Previously a developer, systems programmer/DBA/QA, and project leader with the City of Cleveland, leading financial institutions, and a "Big 4" consulting firm.
- Degrees: Kenyon College, A.B.; Pennsylvania State University, M.S. in Psychology; Suffolk University, J.D.; Boston University, LL.M. in Tax Law.
- Published author and frequent speaker at leading professional conferences.
- Formerly International Vice President of the Association for Systems Management and Executive Editor of the *Journal of Systems Management*.
- Founding Chairman of the New England Center for Organizational Effectiveness.
- Member of the Boston SPIN and SEPG'95 Planning and Program Committees.
- Chair of BOSCON 2000 and 2001, ASQ Boston Section's Annual Quality Conferences.
- Member ASQ Software Division Methods Committee.
- Member IEEE Std. 829 for Software Test Documentation Standard Revision Committee
- Admitted to the Massachusetts Bar and licensed to practice law in Massachusetts.
- Author of book: ***Discovering REAL Business Requirements for Software Project Success***