Continuous Improvement and Exploratory Testing

Paul Carvalho

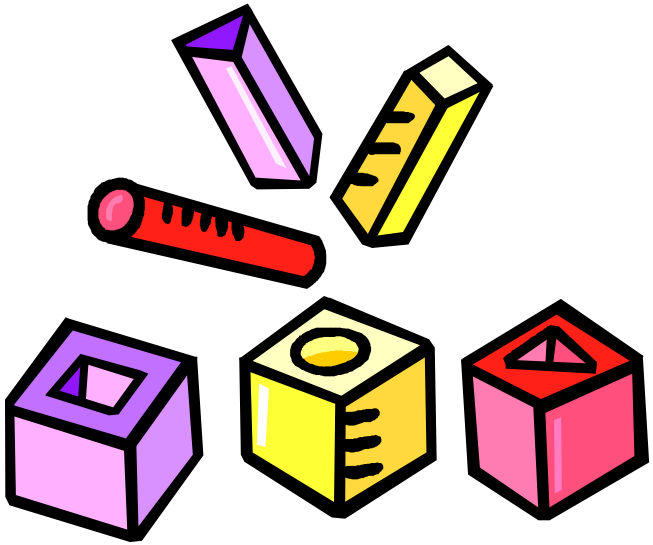
paul@staqs.com

for the Kitchener-Waterloo Software Quality Association

26 April 2006

Agenda

- Continuous Improvement
- Exploratory Testing
- Putting them both together (Case Study)



Why is Continuous Improvement important to you?

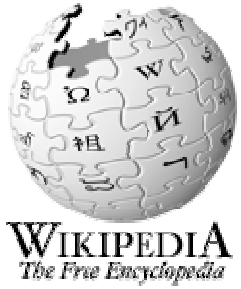
Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots.

So far, the Universe is winning.

- Rich Cook (author)

Continuous Improvement

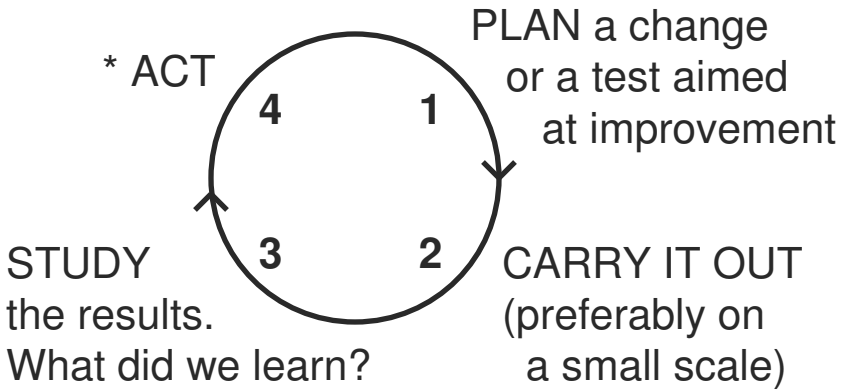
From Wikipedia, the free encyclopedia:



- Continuous improvement is a phrase suggesting that a process or product should always get better as knowledge about it and experience with it accumulates over time.
- It is specifically used in quality systems or management programs such as Total Quality Management, associated with the work of W. Edwards Deming and Walter A. Shewhart.

C.I. simplified: P-D-C-A

- Dr. Deming attributed the basis for the Plan-Do-Check-Act (P-D-C-A) cycle to Dr. Shewart.
- Here's how Deming drew the Shewart Cycle for us:



* ACT. Adopt the change.
or Abandon it.
or Run through the cycle again, possibly under different environmental conditions.

C.I. in an Agile World?

- How does C.I. fit with Agile projects?
- Let's review the Manifesto for Agile Software Development:

Individuals and interactions over processes and tools

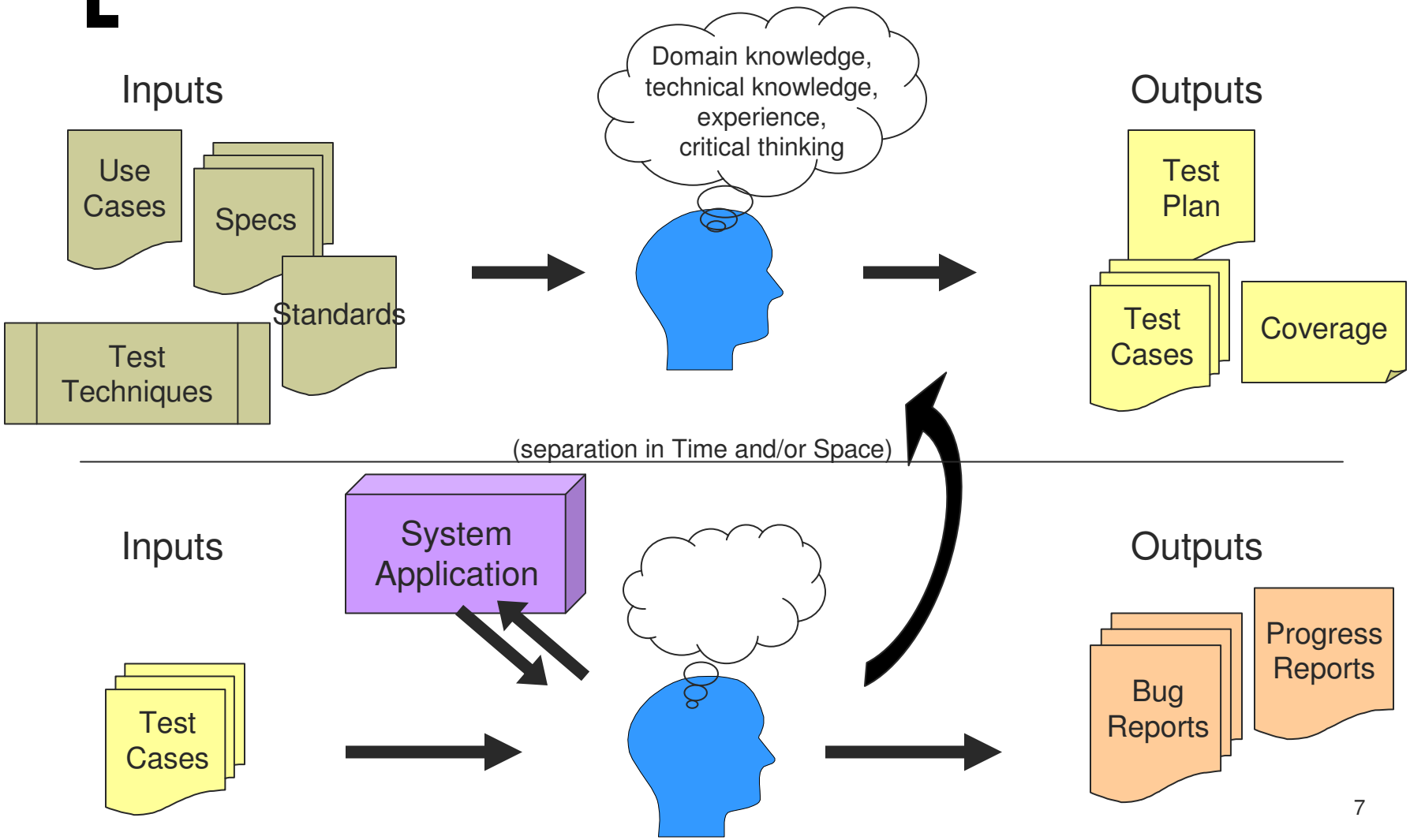
Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

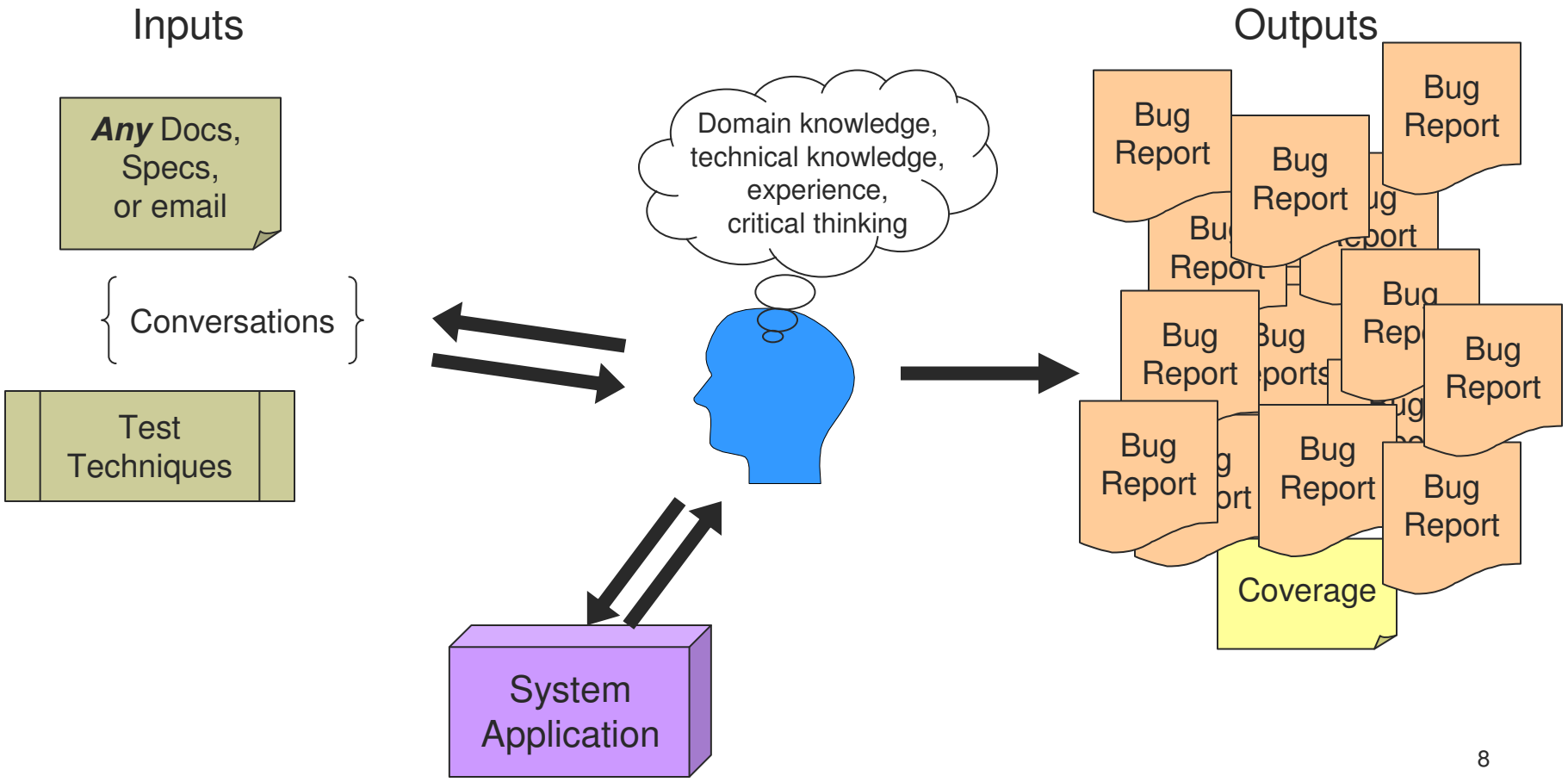
While there is value in the items on the right,
we value the items on the left more.

Factory School of Testing



Exploratory Testing (ET)

Simultaneous Learning, Test Design, and Test Execution.



What are your ET Concerns?

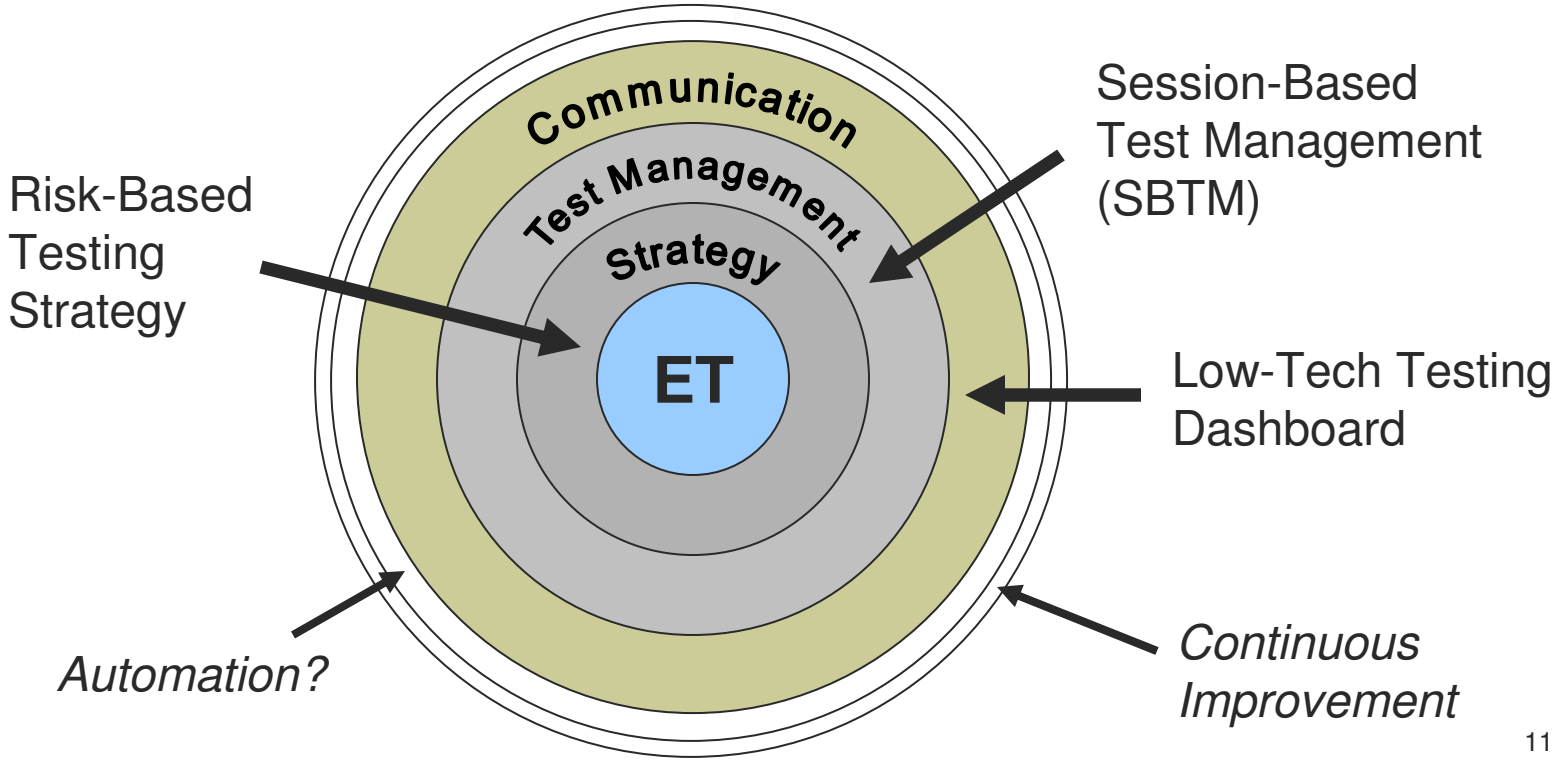
- informal, no structure
- ad hoc, improvised
- not repeatable (no documented test cases)
- no process
- no paper trail (audit-ability)
- no traceability

Is that *all* there is to ET?

Testing is like an onion. It has layers.

Layers of Testing

- Exploratory Testing is the **heart**, but ***not*** everything there is to it.



Agile C.I. = *Lessons Learned..*

- During and after each project, spend some time thinking about your Testing Approach:
  - What worked well?
  - What are some of the gaps?
  - What additional Testing ‘layers’ can we or do we need to add?
- P-D-C-A is the *wrapper* around an entire testing project (or some aspect of it)
  - If you don’t start with a Plan how do you know what to *Study* or *Act* upon after the project?

Case Study: ET & C.I. applied

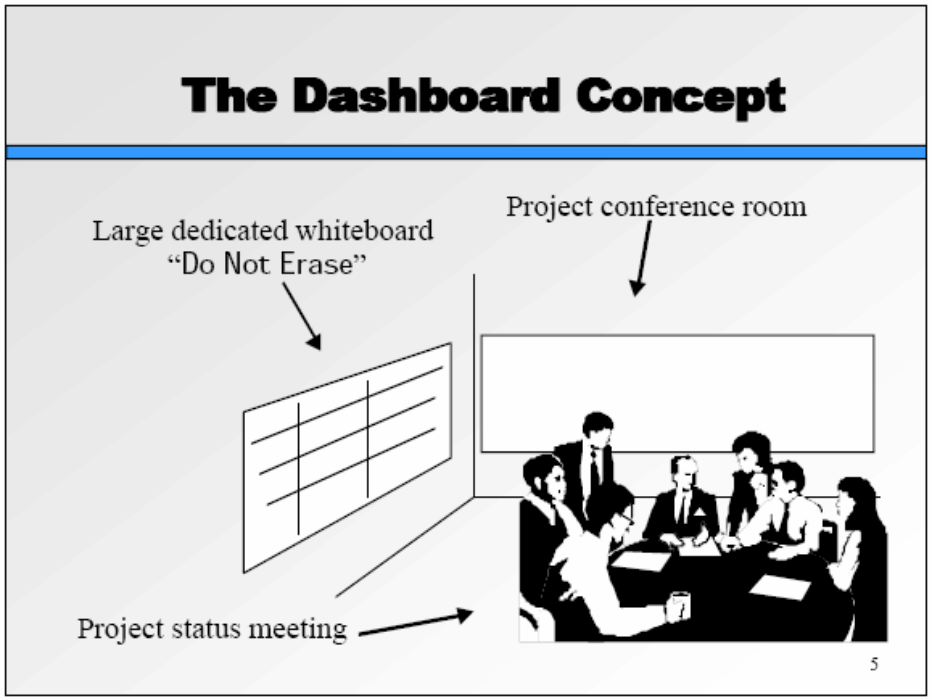
Project	Software Testing Processes
Jan 2004: ■ Major release ■ Minor release	■ Exploratory Testing (ET) ○ Satisfice Heuristic Test Strategy Model ○ Heuristic Risk-Based Testing
April 2004: ■ Major release ■ Maintenance release	■ ET ○ Heuristic Test Strategy Model ○ Risk-Based Testing ■ Session-Based Test Management (SBTM)
September 2004: ■ Maintenance release ■ Major release	■ ET, Heuristic TSM and RBT ■ SBTM ■ Low-Tech Testing Dashboard (LTTD)

SBTM in a nutshell

- Session-Based Test Management provides a *framework* for measuring and managing exploratory testing
- Each “session” has:
  1) Charter – a clear mission for the session
  2) Time Box – focussed effort for fixed duration (e.g. 90 mins)
  3) Reviewable Result – a scannable session sheet
  4) Debriefing – observation check, coaching opportunity
- Metrics generated based on the approved session reports

The Low-Tech Testing Dashboard

Testing Dashboard				Updated: 2/21 Build: 38
Area	Effort	C.	Q.	Comments
file/edit	high	1		
view	low	1+		1345, 136
insert	low	2		
format	low	2+		automatio
tools	blocked	1		crashes: 1
slideshow	low	2		animation
online help	blocked	0		new files n
clipart	none	1		need help t
converters	none	1		need help t
install	start 3/17	0		
compatibility	start 3/17	0		lab time is
general GUI	low	3		



(Slides from a James Bach presentation titled "A Low-Tech Testing Dashboard")

Lesson Learned: Improving the LTTD

- The Low-Tech Testing Dashboard didn't identify *progress* through each of the major testing areas granularly enough for our Project Manager
- We modified it to include a column to indicate progress through estimated sessions:

Project: CoolName					Last Updated: Aug. 16
Area	Sessions Required: Est. / Rem.	Effort	C.	Q.	Comments
File/Print	5 / 2	High	2	😐	7329, 7345

In Summary

- Exploratory Testing is agile, and *maybe* even Agile
- Exploratory Testing and Continuous Improvement are both *exploratory* and *continuous* processes
- Continuous improvement can be as heavy (i.e. CMM, TMM, TPI, etc.) or as light a process as you want.

References

- http://www.agilemanifesto.org/
- http://www.contextdriventesting.com/
- http://www.satisfice.com/
  - Heuristic Risk-Based Testing (article)
  - Session-Based Test Management (Test Methodology)
  - Low-Tech Testing Dashboard (presentation)
- http://www.testinglessons.com/
  - Lessons Learned in Software Testing (book)
- http://www.staqs.com/