



Treat Defects Like Plane Crashes

Alan Willett, Software Engineering Institute

Dayton, Ohio

SPIN

October 18, 2005

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

Sponsored by the U.S. Department of Defense

© 2005 by Carnegie Mellon University



Trademarks and Service Marks

The following are service marks of Carnegie Mellon University.

- Capability Maturity Model IntegrationSM
- CMMiSM
- Team Software ProcessSM
- TSPSM
- Personal Software ProcessSM
- PSPSM

The following are registered trademarks of Carnegie Mellon University.

- Capability Maturity Model®
- CMM®



Flight Quality

Odds of a flight accident or incident: 8 in 1,000,000 take-offs^{*1}

Odds of flying a major airline and becoming a “fatality:”
1 in 3,000,000 passengers^{*2}

Odds of winning a lottery by guessing 5 out of 6 numbered balls correctly (out of a 50-ball drum): 16 in 1,000,000 draws

*1 Source: Federal Aviation Administration

*2 Source: Air Transport Association



Response to Airplane Failures

The following is a overall view of the complex airline failure investigation process.

- Determine the root causes of the failure.
- Inspect the fleet of airplanes for similar defects.
- Fix the fleet.
- Examine the process to determine where the process failed.
- Improve the process.

Source: (<http://www.nts.gov/events/major.htm>)



Common Response to Software Failure

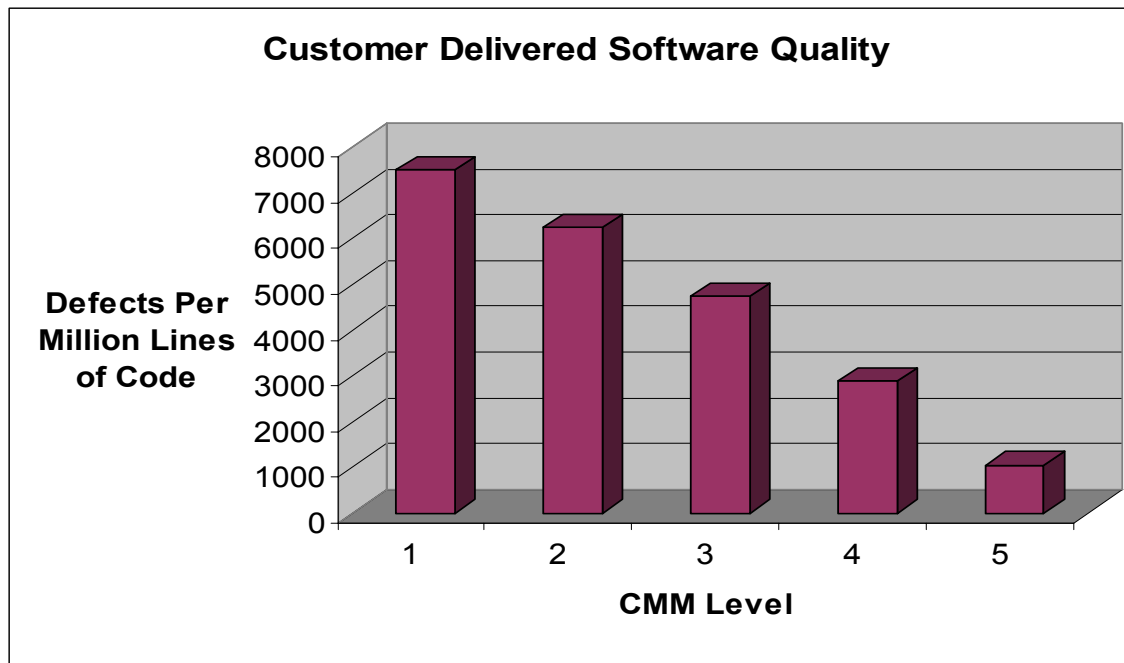
The common response is reactive.

- Triage.
- Fix the critical problems.
- Document work-around.
- Record all the other problems.

Why?



Because There Are So Many Defects!



Source: Capers Jones. *Software Assessments, Benchmarks, and Best Practices*, 2000.

As the size and ubiquity of software systems grow

- the number of software “incidents” become more common
- the concern over the safety of software system rises



The Culture of Defects

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University



What is a Culture?

A culture is a self reinforcing system.

A culture is a group of people sharing a “system of belief”.

The longer the history, the deeper the roots of those beliefs.

For example...



Quotes from Organizational Cultures

“I want to see the most aggressive schedule possible!”

- A program manager at the project planning kick-off.

“Will you be done on time?”

- A Division president (responsible for over 5000 people) calling individual line managers and developers.

“Hurry up and get it to test so we can start finding the defects.”

- A Vice President in charge of a major new product development project repeatedly in status meetings.

“Never slip a date until the day you miss it.”

- A sentence embedded in the project management training at one company.



An Example Large System Project

1,000,000 LOC to drive a real time system.

5 different operating systems, 7 different software languages.

300 software developers.

Product released globally to a large profitable success.

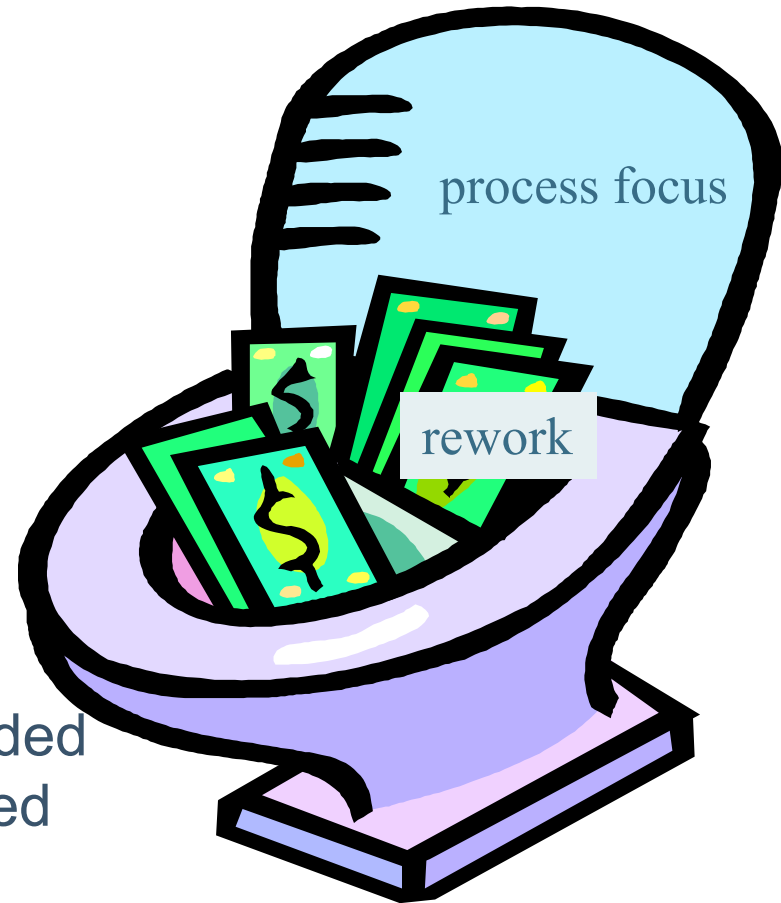


At What Cost? - 1

The project focus was on product excellence.

The process focus was on testing quality in.

The rework level likely exceeded 70% with a cost that exceeded 100 million dollars.





At What Cost? - 2

To handle the over 8,000 defects found in system test, the project

- built an additional building with the sole purpose of testing the software
- hired 100 testers to work on testing the product 24 hours a day, 7 days a week.
- hired 30 software developers whose sole purpose was to chase the trouble reports in test
- hired 20 people to do the paperwork of the defect tracking

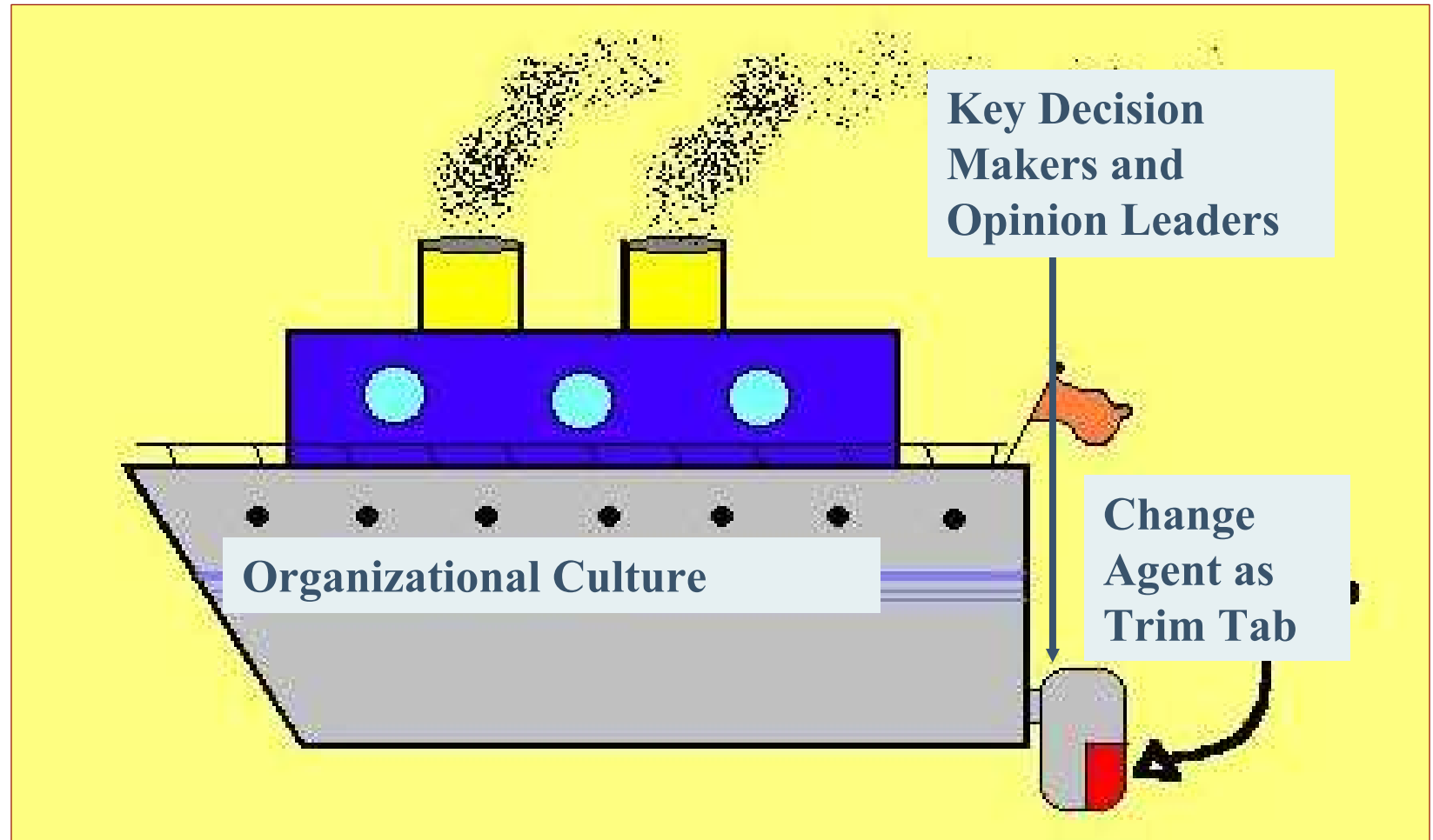


Carnegie Mellon
Software Engineering Institute

So What Do We Do?



Change Agent as Trim Tab





The Role of the Change Agent

In large boats

- the rudder is too large to effectively turn the boat without breaking
- the trim tab is a small part added to the rudder that enables it to turn.

In organizations, the culture is largely created and influenced by key decision makers and opinion leaders

- Developers
- Team leaders
- Managers
- Executives

The change agent can influence the culture to change by helping change the “system of beliefs” of the organizational “rudder”.

To do so, the change agent must be clear about their own system of belief.



My Personal Software ProcessSM Experience

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

**Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University**



Short History Of Alan

Grew up on a dairy farm where measurement was a standard way of running the business.

Alan enters the world of software where measurement was “and the ship date is...”

Alan formally enters the world of software process improvement on an SEPG.

Alan takes the Personal Software Process for Engineers.



The Personal Software Process

The PSP is a process designed for individual use.

The PSP applies to structured personal tasks.

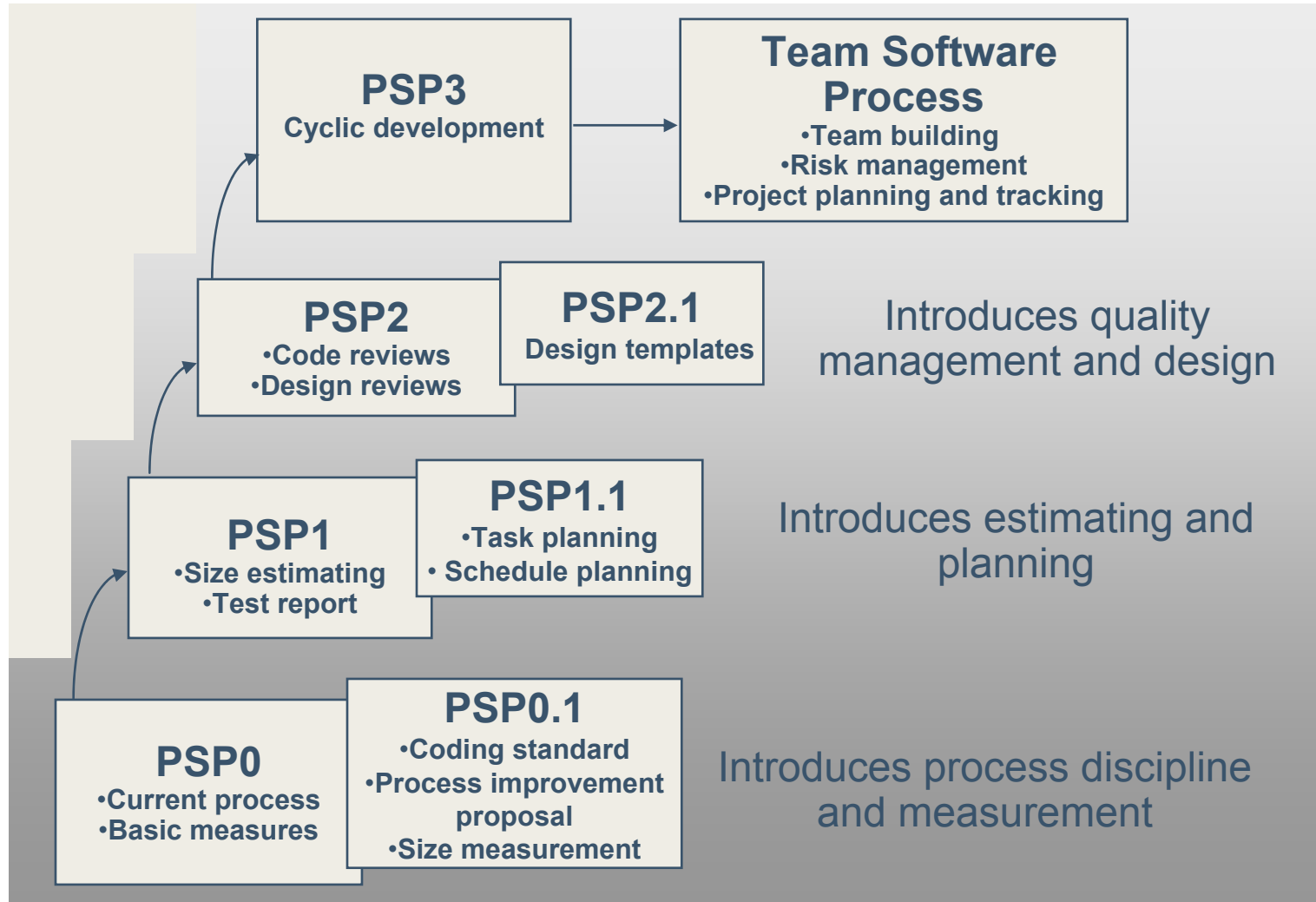
With PSP individual developers use their data to manage and improve their performance.

This strategy provides

- better estimating, planning, and tracking
- protection against over-commitment
- a personal commitment to quality
- personal involvement in process improvement



The PSP Course Sequence

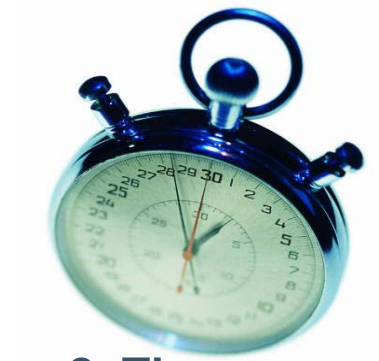




PSP Measurement Framework



1. Size



2. Time



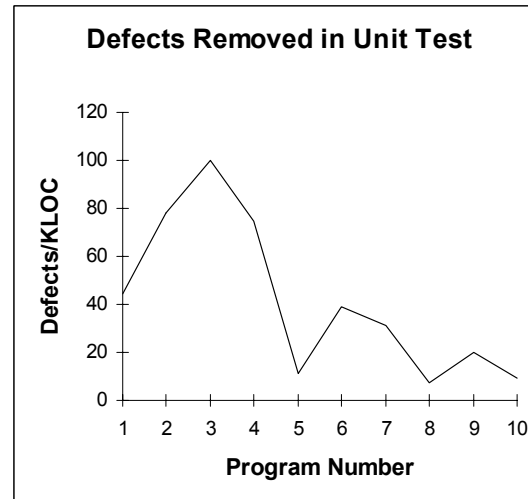
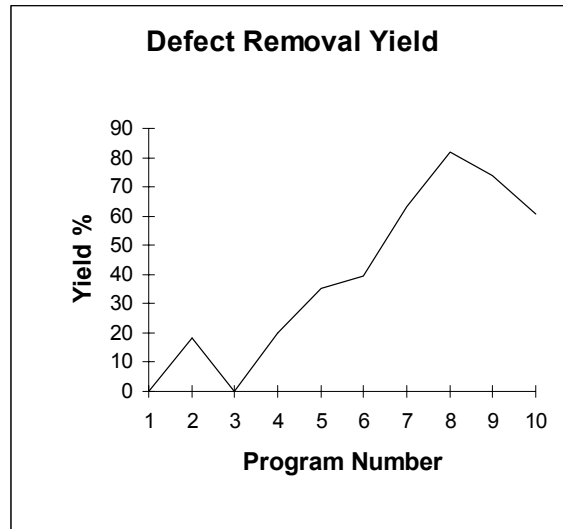
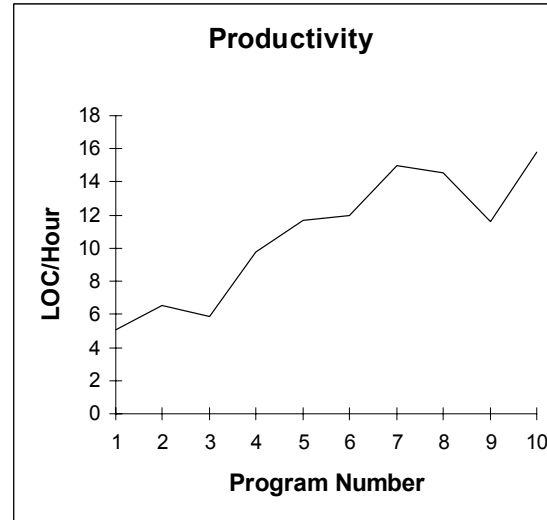
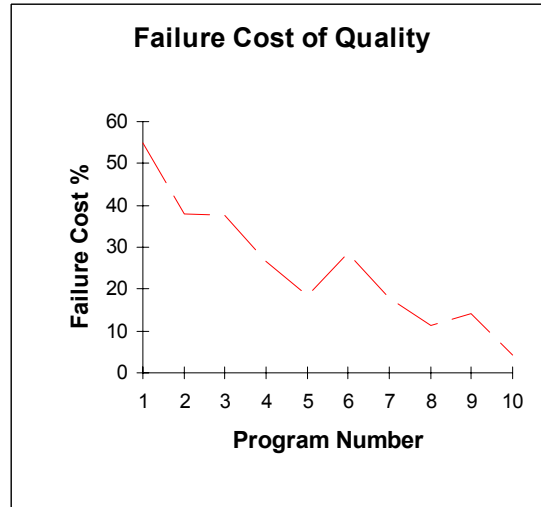
3. Defects



4. Schedule

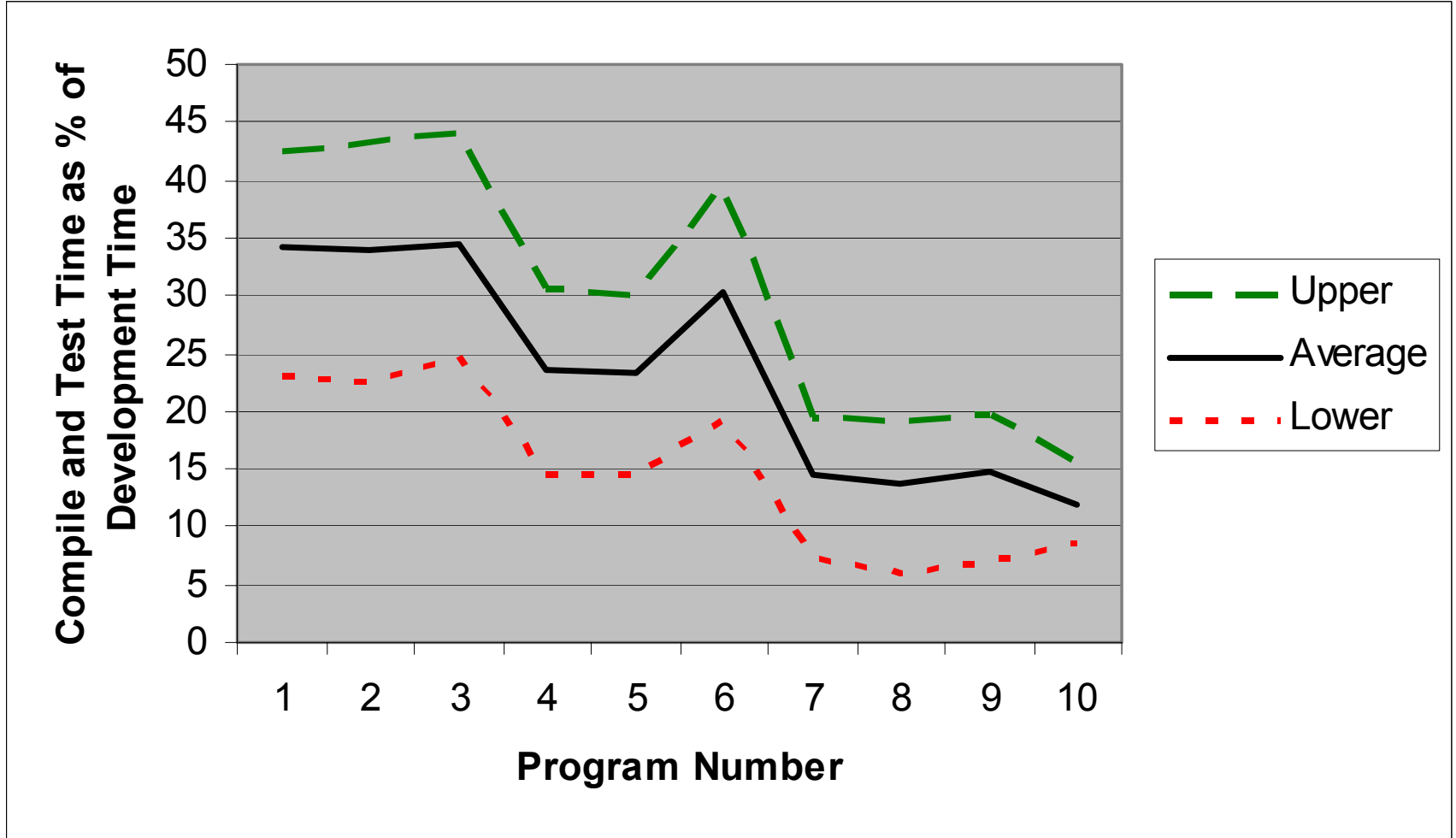


A Sample of Alan's Results





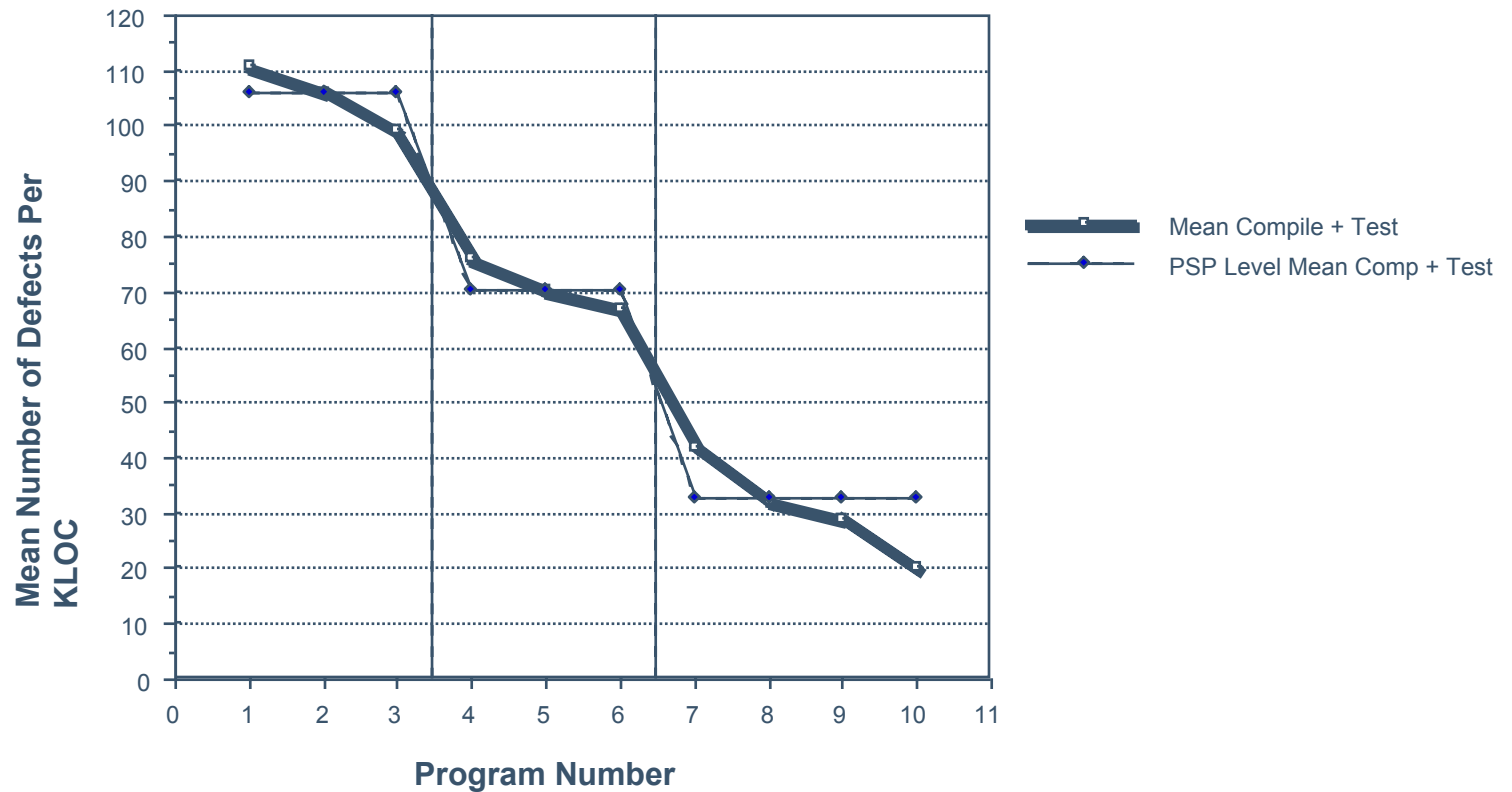
Compile and Test Time – 810 Engineers





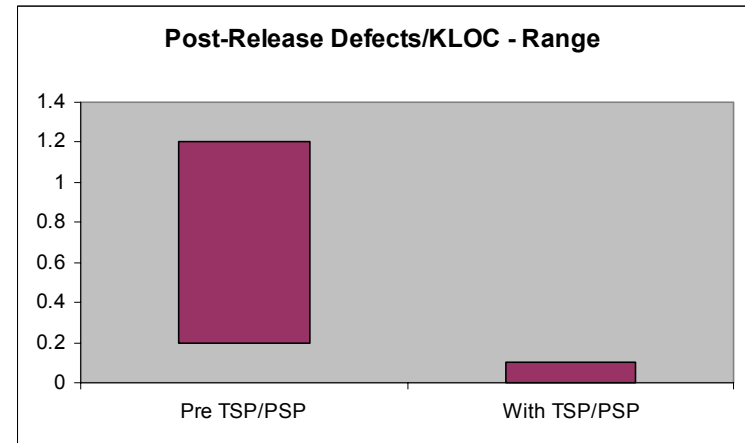
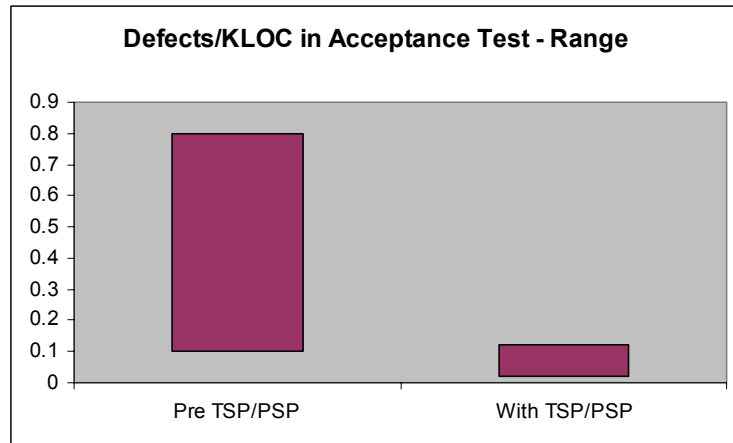
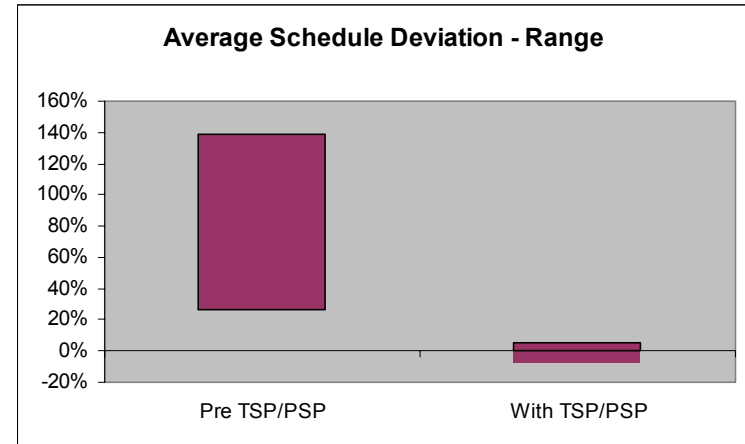
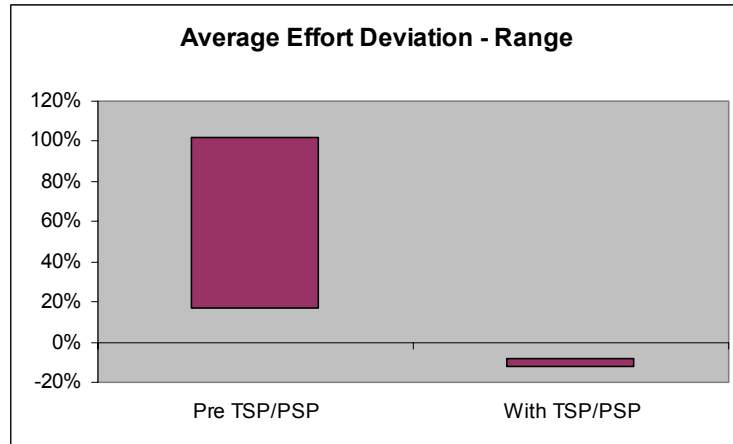
PSP Quality Results

Defects Per KLOC Removed in Compile and Test





TSP Performance Comparison*



*From a study of 18 TSP projects in four organizations conducted in 2000

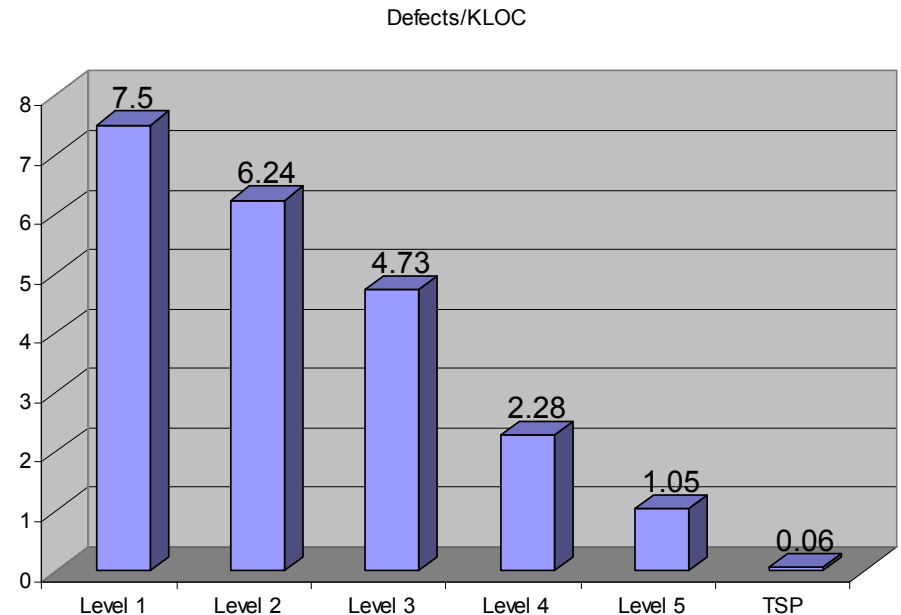


TSP Product Quality

In an analysis of 20 projects in 13 organizations, TSP teams averaged 0.06 defects per thousand lines of new or modified code.

Approximately 1/3 of these projects were defect-free.

For comparison, these results are substantially better than high maturity organizations.



Source: CMU/SEI-2003-TR-014.



Alan's Expectations of Excellence

1. Work as a team - Those that do the work, should plan the work, own the process, improve the process.
2. Build a product the customer will truly appreciate.
3. Quality is key.
4. Create a plan that makes an “as sure as possible” commitment.
5. Variance Happens – understand variance and use data and process to minimize.
6. Minimize late surprises.
7. Continuous improvement (Customer Satisfaction, Quality, Predictability, Productivity)



Treat Defects Like Plane Crashes Reprise

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

**Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University**



Quality Attitude Response to Failure

Fix the product.

Improve the process.



Fix the Product

Determine the defect behind symptom discovered in test.

Determine if other parts of the system have this defect.

Analyze the component or fix defect-density map to identify defect-prone modules.

Determine the fix strategy.

Execute the fix strategy.



Improve the Process

Determine the failures in the software process that allowed the defect to escape into test (or beyond).

Generate improvement proposals to prevent future escapes.

In the context of other improvement proposals, determine the highest leverage improvements.

Plan the improvements.

Execute the improvement plan.



The Story of the “Time” Defect

The following is an example of a problem discovered by the customer and a team’s response to that problem.

This story is based on a number of real experiences that were merged to allow the best illustration of the concepts in the short time we have.



What is the Problem?

The symptoms

- Database records that should have had the same recorded time had different time stamps.
- When client and server were in different time zones, the records were recorded as different times.

The defect: A software component was using local time instead of Greenwich Mean Time (GMT).



Where Else is the Problem?

A search was done to determine if other parts of the system had the same problem.

Ten other components were discovered with this defect.



Consider Overall Product Quality

The team analyzed the fix defect-density map to identify defect-prone modules.

The team determined that one of the modules had exceeded the acceptable quality level.

The team scheduled that module for a complete design and code reinspection.



Determine the Fix Strategy

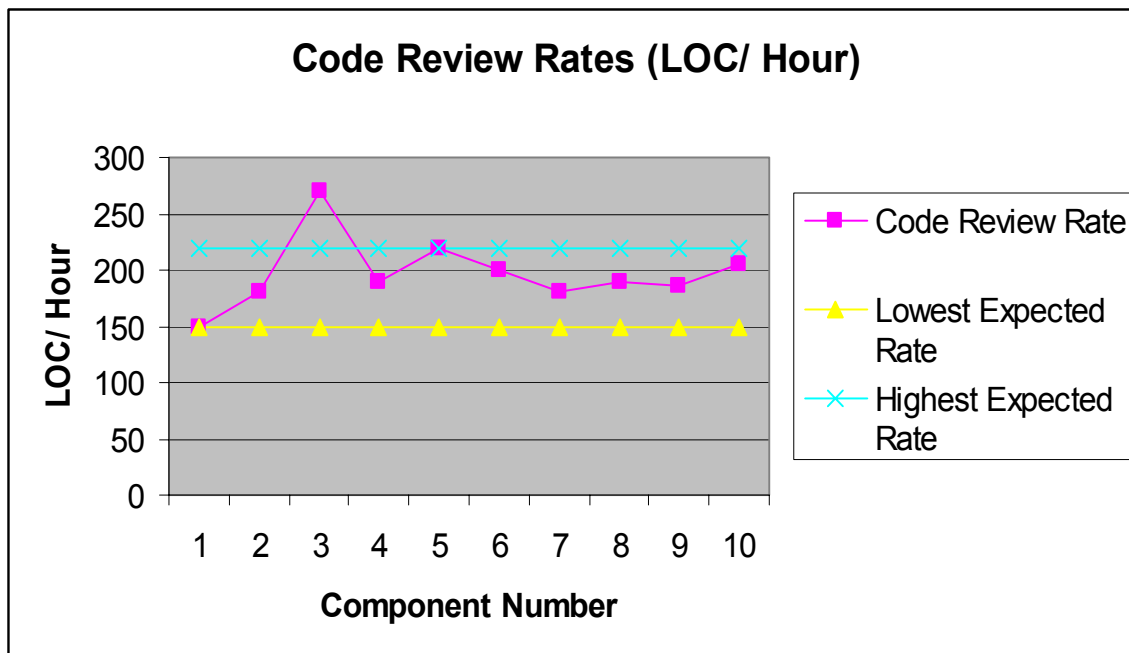
The team decided to fix all of the defects in a single release.

The code would be inspected after the changes were made.



Where Did the Process Fail? -1

The team examined the review rates.



Module 3 (which had been scheduled for reinspection) was the only module that exceeded the highest expected review rate.



Where Did the Process Fail? -2

The team looked at the failed defect-removal phases.

- Personal design review / design inspection
“GMT” was not part of the design standard or the design review checklist.
- Personal code review / code inspection
“GMT” was not part of code standard or the code review checklist.
- Unit test
No tests were conducted across time zones.
- System test
No tests were conducted across time zones.



Where Did the Process Fail? -3

Key findings

- Only a small set of people on the project understood the “GMT rule.”
- Design standards and code standards did not address the “GMT rule.”
- Inspections only caught the problem when the inspectors were people who understood this rule.



Improve the Process

The team generated a number of improvement proposals.

- Determine other implications of a “global” product that may have been missed.
- Update design / code standards to include GMT.
- Update design / code checklists to include GMT.
- In unit and system tests, simulate different time zones in testing.
- Update new engineer training to include design standards.



Prioritize Improvements

In the context of other improvement proposals, determine the highest leverage improvements.

- What do we improve right now?
- What should be deferred?



Improvement Priorities

What do we improve right now?

- preliminary investigation of “global” implications
- update design / code standards and checklists
- investigate how to simulate “time zones” for system tests
- “lunch and learn” on GMT

What should be deferred?

- simulate different time zones in unit testing



The Rest of the Story

The team quickly planned and executed the recommended immediate steps.

Implementation of these steps completely eliminated the escapes to system test and to customers.



Summary

Zero defects discovered by customers should become the norm.

This should be expected by both the software development community and customers.

Take steps towards the “zero-defect” goal. I suggest

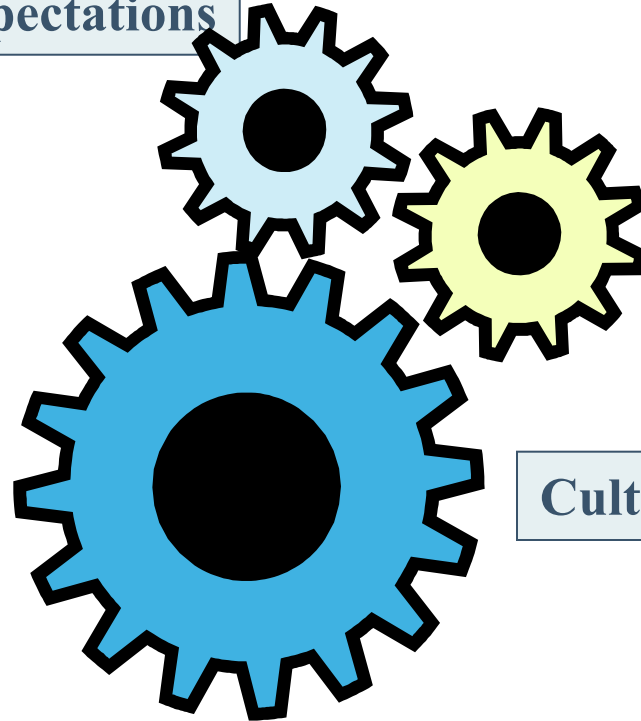
- taking a set of defects and leading a team of people through the response process
- providing training in the “zero-defect mindset”

The goal of zero delivered defects is possible. I have seen many TSP teams achieve this outcome.



Our Challenge

Change Agent Expectations



Cultural Leaders

Cultural Reinforcements

Can we make these all work together?



Carnegie Mellon
Software Engineering Institute

Thank You!



Contact Information

Alan W. Willett
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Email: awillett@sei.cmu.edu

Phone: 607-592-7279

