Principles of Successful Software Inspections

Dennis Beeson and Tim Olson

World-Class Quality
3082 Hamline Ave. N., St. Paul, MN. 55113
Phone: 612-636-2234
Email: DDBeeson@gte.net, Tim.Olson@worldnet.att.net

Abstract

Software inspections remain the most effective method of early defect detection and removal (e.g. early defect detection 80 - 90%, ROI 7:1 - 12:1). Yet many organizations are unsuccessful at invoking the cultural changes required to implement and sustain an effective software inspection process. So what can an organization focus on to change people's perspective of inspections to develop a quality culture centered around software inspections? This paper will identify some of the essential attributes or principles of software inspections which facilitate in building and sustaining a quality culture. This paper will measure the F/A-18 Software Development Team's inspection process against these principles to determine software inspections effectiveness as well as identify areas for future improvement.

Objectives

The objectives of this paper are to:

- present some common cultural problems associated with software inspections.
- present some successful software inspection data from the F/A-18 Aircraft.
- present an overview of effective principles that are successful when performing software inspections.
- benchmark the F/A-18 Software Development Team's inspection process against inspection principles identified to determine effectiveness and indicate areas for improvement.

In This Paper

The following table describes the title and starting page of each section:

The Positive Impact of Inspections on F/A-18

Background

Since 1987, the F/A-18 Software Development Team (SWDT) at the Naval Air Warfare Center - Weapons Division (NAWC-WD) has been providing system and software engineering maintenance and upgrades on the F/A-18 A/B model aircraft Mission Computer (MC) and Stores Management System (SMS) for the US Navy and Foreign Military Sales (FMS) customers.

F/A-18 Mission Computer Upgrades

The F/A-18 SWDT has undertaken four major upgrades to the F/A-18 aircraft's Mission Computer (MC) Operational Flight Program (OFP). The MCs are the center of the F/A-18's avionics architecture. The MCs are the primary link between the aircrews cockpit display environment and the aircraft's tactical and air vehicle management avionics subsystems.

F/A-18 MC Defect Removal Life Cycle

Figure 1 illustrates the overall impact software product inspections and software process improvement have had on product quality. During a ten year period involving over 5000 inspections, early defect detection and defect prevention have significantly moved the defect removal curve to the left. The majority of product defects are now found in the requirements, design and coding phases. In fact, over 86.6% of all defects are found before testing. The defect removal life cycle curve is also used to demonstrate product maturity to the customer.

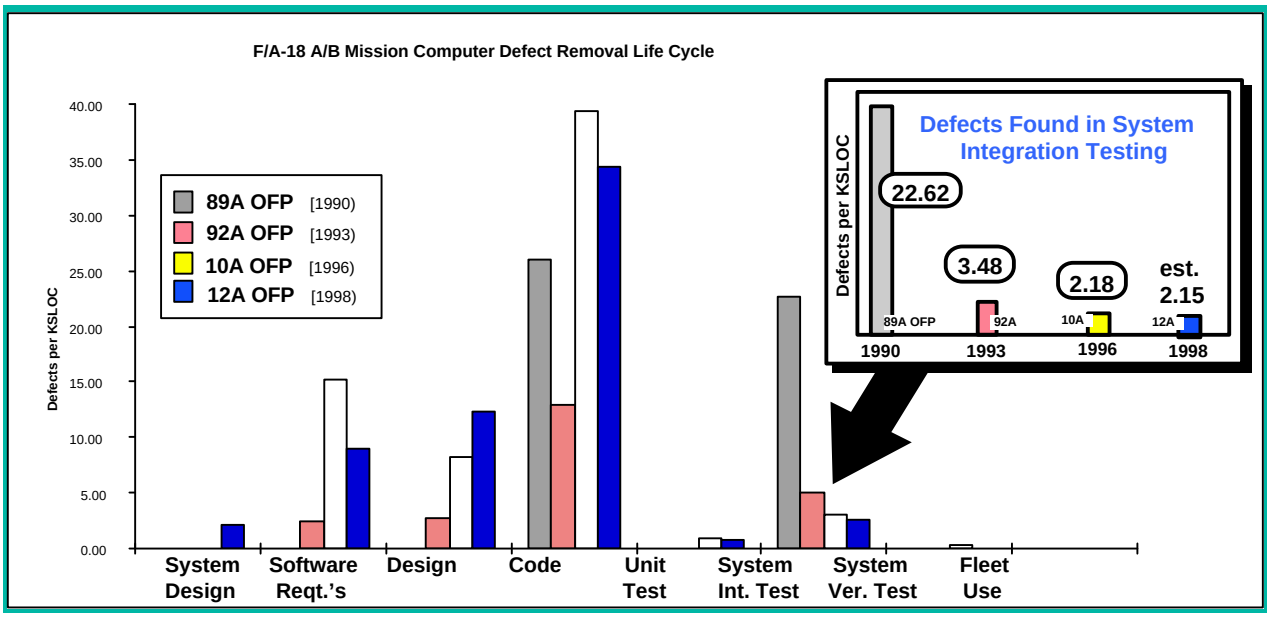


Figure 1: F/A-18 A/B Mission Computer Defect Removal Life Cycle

Benchmarking the F/A-18 Inspection Process

World-Class Software Benchmarks

Benchmarking the F/A-18's inspection process data against a world-class level. Over the last 10 years, the F/A-18 SWDT has progressed from an average performing SEI CMM Level 1 organization to comparing favorably against world-class software organizations. Table 1 characterizes current performance of various world-class organizations to the F/A-18 SWDT current performance capability.

Measurement	World-Class Benchmark*	F/A-18 Software Development Team
Quality		
Inspection Defect Removal Efficiency	80%-90%	86.6%
Post-Release Defect Rate	.01 per KSLOC	.01 per KSLOC
Cost		
Total Cost Savings	$7.5-$45 Million	$14.4 Million ($ 3.6M per major update)
Inspection Cost	$2,500 on Average	$1,500 on average
Return on Investment (ROI)	7:1 - 12:1	7:1
Schedule		
Schedule / Cycle Time	Reduced 10-25% per yr.	Reduced 9% per year
Productivity	Doubled in 3 years	Increased 62% in 3 years

Table 1 World-Class Software Benchmarks *derived from World-Class Quality - Timothy G. Olson copyright 1995 - 1996

Principles of Successful Software Inspections

Principles of Software Inspections

To fully understand how to optimize software inspections to promote team building and improve individual learning it was necessary to have a clear description of the core attributes or principles that make software inspections successful from a people prospective. Only after these principles were identified was it possible to make the necessary process improvements. Research and benchmarking of software inspections best practices were successful in identifying the following principles found in most effective inspection processes:

Principles	Description
Leadership	Management should provide resources and be an active participant in communicating, mentoring, and building the organizations quality culture. Facilitate the team in setting clearly stating mission, goals, and objectives centered around quality, quality measurement, and quality improvement.
Quality Culture	Foster commitment to designing in quality. Develop an understanding of the quality expectations, values, and priorities of the immediate and final customers.
Responsibility	Foster responsibility for the quality of the end product
Process Ownership	Team participation in process definition and process change mechanisms.
Defect Prevention	Foster commitment to learning from past defects.
Communication	Foster open honest communication supported by effective meeting facilitation. Understand the strength and weaknesses of self, team, and organization and use this diversity to optimize effectiveness. Operate organization with integrity, making decisions based on what is truly best for product quality and the organization.
Feedback	Give feedback on individual defects found, overall product quality, status of defect prevention (e.g. common defect trends identified, changes to data driven checklists).
Defect Analysis	Analysis and tracking of defect density per development phase and determining criteria for reinspection.
Agreement	Management, engineering, suppliers, and immediate and final customers should effectively review and agree to product plans (e.g. schedule, resources, staffing, quality objectives, etc..).
Defined Process	Fully communicate what is expected of management, engineering, suppliers, and immediate and final customers (e.g. what, how, when, were, why).
Training	Effectively train people in inspection purpose, roles, process, facilitating meetings.
Defect Identification	Formal mechanism for documenting, categorizing, and dispositioning defects. Defect identification involves gathering defect and associated metrics (e.g. size, effort, cost, time, rework). Defect identification is usually supported by data driven checklists.
Accountability	Formal mechanism hold developers, reviews, and moderators accountable for fulfilling their role in the inspection process.

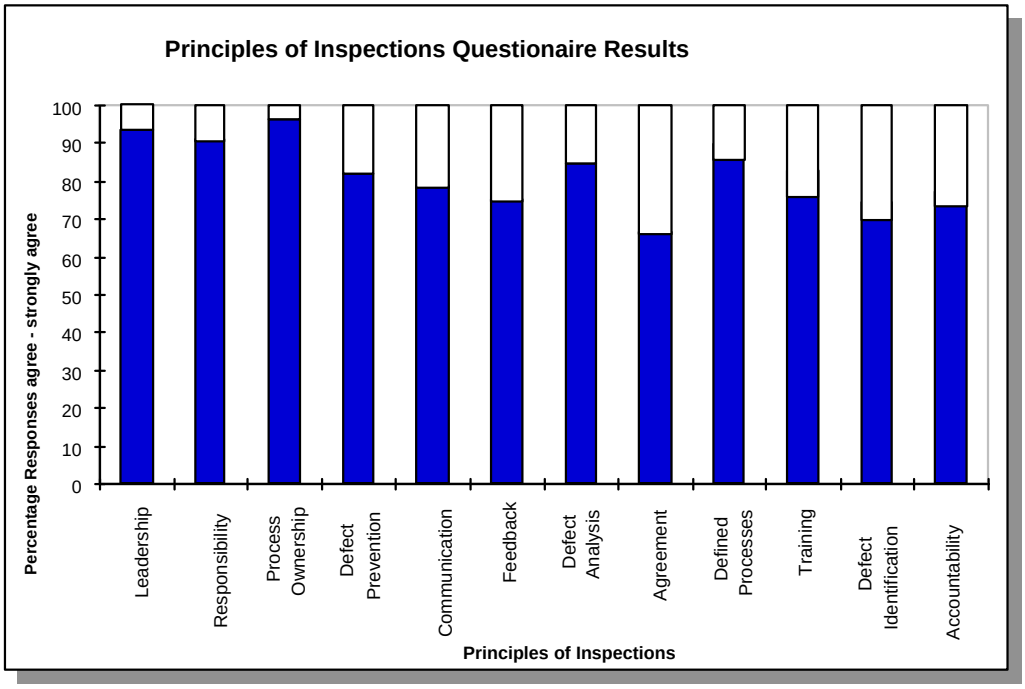
Measuring the Principles of the F/A-18 Inspections

F/A-18 Software Team

Over the last ten years the F/A-18 Software Development Team has training approximately 50 software engineers in formal inspections. Most have never used formal inspection methods before working on the team. As they progress in knowledge and understanding of inspections they move up in their level of commitment to the teams product quality goals and buy-in to the inspection process. The principles of software inspects need to be effective and in place to protect against loosing buy-in or commitment, issues of non-compliance, or to assist in gaining enough trust in the team and the inspection process to move to a higher level of buy-in or commitment.

Questionnaire

A survey was conducted of the F/A-18 Software Development Team in order to measure the buy-in and commitment to the software inspection principles. The table below shows the results:



Summary

Achieving measurable results using software inspections requires understanding fundamental principles, and then tailoring those principles to practice. These principles must then become part of an organization's day to day business.

References

References

The references used for this presentation are:

- [Covey 91] S. R. Covey, Principle Centered Leadership, New York, NY: Simon & Schuster, 1991
- [Senge 90] P. M. Senge, The Fifth Discipline, New York, NY: Currency Doubleday, 1990
- [Curtis 95] Curtis, Bill, et al. People *Capability Maturity Model* (CMU/SEI-95-MM-02). Pittsburgh, PA: Carnegie Mellon
- [Deming 95] W. E. Deming, Out of Crisis, Cambridge, MA: MIT Center for Advanced Engineering Study, 1995
- [Beeson 98] D. D. Beeson and T. G. Olsen, "Benchmarking F/A-18 Software Inspection Data", SEI 1998 Conference Proceedings, 1998.
- [Barnard 94] Barnard, J. and Price, A. "Managing Code Inspection Information", IEEE Software, March 1994.
  [Billings 94] Billings, C., et al. "Journey to a Mature Software Process", IBM Systems Journal, vol. 33, no. 1, 1994.
- [Ebenau 94] Ebenau, B. and Strauss, S., *Software Inspection Process*. McGraw-Hill, 1994.
- [F/A-18 96] F/A-18 MC/SMS Software Processes; February 12, 1996.
- [F/A-18 97] F/A-18 Systems Engineering Process Guide; August 9, 1997.
- [Fagan 76] Fagan, M. "Design and Code Inspections to Reduce Errors in Program Development", IBM Systems J., no. 3, 1976. pp 182-210.
- [Fagan 86] Fagan, M. "Advances in Software Inspections", IEEE Transactions on Software Engineering, July 1986
- [Gilb 93] Gilb, T. and Graham, D. *Software Inspection.* Addison-Wesley, 1993.
- [Grady 94] Grady, R. and Van Slack, T. "Key Lessons In Achieving Widespread Inspection Use", IEEE Software, July 1994.
  [Herbsleb 94] Herbsleb, James, et al. "Benefits of CMM-Based Software Process Improvement: Initial Results", CMU/SEI-94-TR-13, 1994.
- [Humphrey 89] Humphrey, W. S. *Managing the Software Process*. Reading, MA: Addison-Wesley Publishing Company, 1989.
- [Olson 94] Olson, Timothy G., et al. "A Software Process Framework for the CMU/SEI-94-HB-01, 1994.
- [Olson 96] Olson Timothy G., "World-Class Software Inspections", SEI 1996 SEPG Conference Proceedings, 1996.
- [Paulk 93] Paulk, Mark C., et al. *Capability Maturity Model for Software, Version 1.1* (CMU/SEI-93-TR-24). Pittsburgh, PA: Carnegie Mellon University, 1993.
- [O' Hara 97] F. O'Hara, Achieving maximum benefits from formal reviews/inspections - strategies and case studies, proceedings of EuroSTAR'97, Edinburgh and SPI'97, Barcelona, 1997.

