Shared Defect Detection : The Effects of Annotations in Asynchronous Software Inspection

Chapter 2

Definitions

Chapter 3

- Overview
- Preparation
- Inspection
- Rework
- Follow-up

All participants are invited to attend the overview meeting in which the objectives of the inspections are defined. Roles may be assigned and inspection materials -- such as

process includes planning, overview, preparation, analysis, inspection, rework, and follow-up. The moderator prepares for the inspection by selecting participants and entry criteria. Participants gather in an overview meeting in which the inspection objectives

3.7 Gilb Inspection

Gilb and Graham (1993) developed a comprehensive inspection method. Inspection steps include entry, planning, kickoff, checking, logging, brainstorming, edit, follow-up and exit. While the names of the steps vary from those used by Fagan, they are based on the steps in Fagan Inspection with additional pre- and post-inspection activities that help

3.9 Phased Inspection

head moderator who controls the overall process. Then each team separately conducts its own defect detection using their preferred technique. An additional activity, collation, is performed by the head moderator before the post-inspection activities begin. The

When programs are modeled in terms of abstract operators and data, i.e. as state machines, their correctness can be determined using a set of correctness arguments. The

3.15 Software Inspection Standards

3.16 Meetingless Inspection

Meetings are considered an essential part of successful or effective inspection. Some

projects, an inspection history may be needed. Searching through documents is an inefficient way to produce this history and can lead to an overall decrease in productivity.

4.2 Computer Supported Software Inspection

There have been a number of commercial products and academic prototypes that attempt to provide computer support for software inspection. Computer systems ranging from

4.2.2 Individual preparation

Individual preparation can benefit from computer support in various ways, including:

- reviewing artifacts for defects, computer support by using tools such as source code profilers can help discover certain types of defects
- inspectors can easily access checklists and other documents for referencing
-

In a distributed environment, asynchronous inspection support can eliminate the need for synchronous meetings.

4.2.4 Data collection

Conventionally, data is collected manually. This can be time consuming and error-prone.

support for collaborative activities and to control the inspection process. CSRS (Johnson, 1994) radically departs from conventional inspection techniques by emphasizing asynchrir nna9w[(34)].ir nnaIn

parallel communication and group memory, via Groups Outline Interface. Results show

(Mashayekhi et al., 1993), which is used for creating annotations and a defect list. An

4.6.2 Notes Inspector

ASSIST (tool-based) inspection over paper-based inspection (MacDonald & Miller,

4.7 Screen Captures of Softwano44.7 ection Tools

Figure 5. ICICLE*

Figure 7. S[*] c r u t i n y

Figure 9.

Figure 10.

Chapter 5

A Framework for Software Inspection Research

of each step, inputs to the process, context of inspection and technology are key factors that influence software inspection outcomes. Empirical studies under such frameworks

Chapter 6

Research Design

private review, public review, consolidation, and group review. Documents to be

attention are important factors in performing such a task (Ashcraft, 1989; Kantowitz & Sorkin, 1983). The higher the noise, the more difficult the task is.

It is conceivable that individual inspectors having different levels of expertise may be affected differently by a “contaminated” document. In particular, novice inspectors may inspection. Thus, learning is an essential part of software inspection and it may be hypothesized that shared asynchronous software inspection would induce more learning leading to better inspection.

In addition, the software inspection task, particularly defect detection, is a repetitive

6.3 Experimental Design

6.3.1 Subjects

the target materials were fed into a compiler, but all of them were carefully examined to be free of syntactic errors.

Materials were prepared to test the hypothesis and to examine some search questions. These included determining learning and contamination effects caused by visible annotations. For effect of annotation on learning, two similar defects were located in different positions in a target material. The first one was annotated with true defect description, while the other one was left as-is. Effect on contamination -3.i The

- Identifying documents is effortless

In order to compare effectiveness in T0, which no effectiveness of defect assertion can be estimated, against those of T3 and T7, formulas were constru4.2(on58 provid(re)4.2num we)4.2ulai

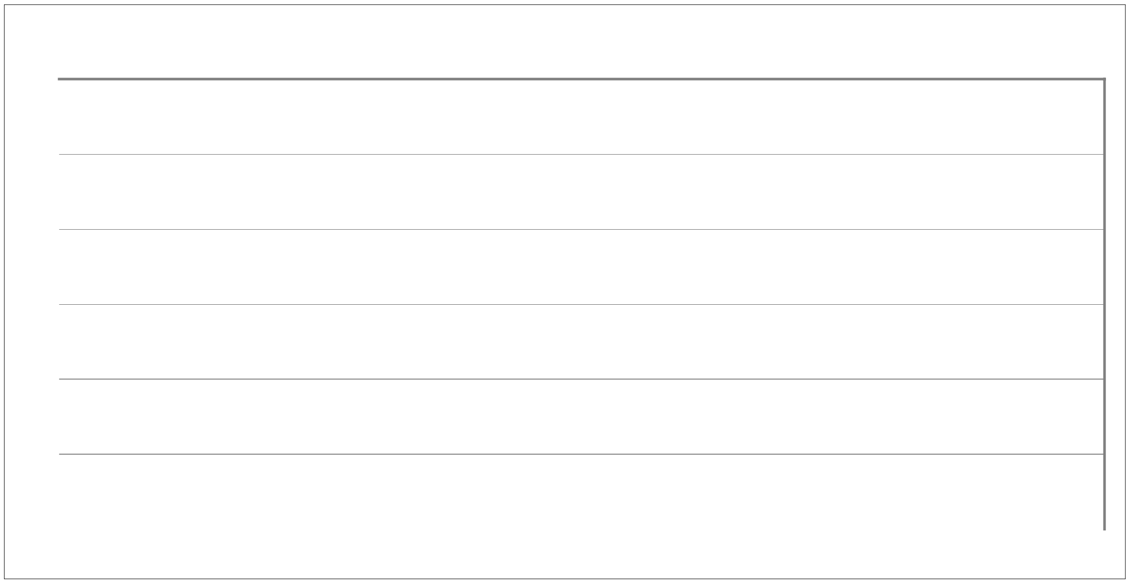
For T0 (private defect detection),

7.5 Inspection Time

Inspection time, which is the time each subject spent on inspecting each target material,

```
                      HOMOGENEOUS
   ORD        MEAN     GROUPS
---------  ----------  -----------
     I        46.852    I
    II        32.852   .. I
   III        24.815   .. I

THERE ARE 2 GROUPS IN WHICH THE MEANS ARE NOT SIGNIFICANTLY DIFFERENT FROM
ONE ANOTHER. (TIME OF 2
```



	T3	**T7**
N	27	
MEAN	87.037	
SD	14.495	
SE MEAN	2.7896	
MINIMUM	50.000	
MEDIAN	100.00	
MAXIMUM	100.00	

	Related Defect	
	Found	Not Found
Annotation		

Defect on Line 90

- Search function for locating strings
-

7.10.8 Validity of the Experiment

inspection: methodology, inspection-support tools, and user behavior. Specifically,

materials. However, studying how these two tasks interact and how inspectors mentally perform asynchronous software inspection would be very valuable for the

```
        default:
          return url;
      }

      return url;
    }

    /*************************************************************************/

```

A.2 Visible Annotations

$\longleftarrow$ *PAGE_WIDTH* $\longrightarrow$

```
                                                                 Page 1
-----------------------------------------------------------------------
------
Title: Software Inspection
ISBN: 0201631814
Price: $49.95  Code B
Year: 1993
Source: Amazon.com

Title: Software Inspection : An Industry Best Practice
ISBN: 0818673400
Price: $38.00  Code B
Year: 1996
Source: Amazon.com

Title: Software Inspection
ISBN: 0201631814
Price: $46.88  Code B
Year: 1993
Source: Fatbrain.com

Title: SOFTWARE INSPECTION HANDBK
ISBN: 0863412254
Price: $26.00  Code B
Year: 1990
Source: Fatbrain.com

Title: Software Inspection Process
ISBN: 0070621667
Price: $47.00  Code B
Year: 1993
Source: Fatbrain.com

5 items found
```

Incomplete Structured Chart (Book_Search)

In inspecting an original document, you simply examine the source code (line-by-line) to detect defects.

A.7 Announcement

[illegible]

A.8 Consent Form

Appendix B : Ques tionnaires]TJ/F14.0634 0 15.4.0634 8.1653 69261.4475

(b) yes, as a class exercise
(c) yes,, weork,

20. Any comments are welcome

C.2.3 Print Module

Line No	T3			T7		
	C	D	M	C	D	M

```
                    TIME
N                     27
MEAN              33.185
171 1T*134 Tm-0.003 Tc(           ATPER       AFPER       AMPER         E1)T        E12 9.0000
SD                21.316
N                                                                  8.000
SE MEAN           4.1023
MINIMUM           9.0000
SD          14.06       9.2276      11.202      14.195      9.1378  8.000
MEDIAN            28.000
SE MEAN     2.7072      1.7758N     2.1558N     2.7318      1.7586 9.0000
MAXIMUM           93.000
MINIMUM     56    9999990     9999990     99999941    9999996     2 8.000
MEDIAN      78    9999990     999999111   9999996 59  9999996 86  2 8.000
MAXIMUM       10        33  9999996 33  9999996 10          10      000

N                     27
MEAN              33.185
```

173j- /F1 Tf11858430 0 11858430 129.8013 713.013Tm0

Variable	Description	Subject Answer

Variable	Description	Subject Answer

Variable	Description	Subject Answer

S	Q11E	Q12	Q13	Q14	Q15	Q16A	Q16B	Q16C	Q19	T1	T2	T3	TM1	TM2	TM3	TPER1	TPER2	TPER3	TPER	CRAT	FRAT	TRAT

how group members like to receive their mail. Clicking one button attaches the document. Tchecl Tm-0.0006mply typ

Johnson, P., & Tjahjono, D. (1997). Assessing software review meetings: A controlled experiment study using CSRS.

Porter, A., & Johnson, P. (1997). Assessing software review meeting: Results of a comparative analysis of two experimental studies. *IEEE Transaction on Software Engineering* ***23***(3), 129-145.

Porter, A., Siy, H., & Votta. L.G. (1998). Understanding the sources of variation in software inspections. *ACM Transaction on Software Engineering and Method* ***7***(1), 41-79.

Porter, A., Siy, H., & Votta, L.G. (1997a). Understanding the effects of developer activities on inspection interval. *Proceeding of 19th International Conference on Software Engineering (ICSE*