Use of Software Inspection Inputs In Practice:
An Empirical Investigation

Research Area
The research area of this study is “software inspection”.

Aim and Significance
The purpose of this study is to investigate the key factors influencing use of these inputs in software industry. An overall research objective of this research is to understand the use of inspection inputs in practice. This increases our understanding of what, why and how these key factors influence the use of inspection inputs.

Objectives
The objectives of this research are:
1. to identify the common types of inspection inputs that are widely used and which appear to be significant in the software industry, and
2. to investigate the key factors influencing the use of these inputs.

Research Problems
1. What are the common types of inspection inputs that are widely used and which appear to be significant in the software industry?
2. What are the key factors influencing the use of these inputs?

Literature
Software inspection was originally introduced by Fagan (1976). The inspection process essentially includes six major steps: planing, overview, individual preparation, group inspection meeting, rework and follow-up (Fagan, 1976; 1986). Recent empirical researchers mainly focus on the defect detection process (Porter, 1995; Basili, 1996; Johnson and Tjahjono, 1998; Biffl, 2001). The normative literature (Kim, et al., 1995; Gilb & Graham, 1993; Freedman & Weinberg, 1995) also documents the inputs and outputs of the inspection process. In this proposal, we use these inputs and outputs to develop a conceptual model by classifying into two types of inputs: explicit inputs and implicit inputs (see figure 1). According to this EIIO (Explicit and Implicit Inputs-Outputs) model, we define five explicit inputs as follow:

1) software artifacts submitted for inspection (e.g. requirements document, design, code, test cases, or any other software artifact) (Laitenberger, 2000; Weller, 1993; Kelly, 1992),
2) supporting documents in the domain application and software development standards adopted by the software organization (Kotonya and Sommerville, 1998; ANSI/IEEE, 1989). This includes business reports which provide domain business information supporting the system being built.

1 Typical defect detection process consists of individual preparation followed by group meeting
3) previously inspected software documents relating to the software artifact (e.g. a related document for a design inspection may be a requirements document) (Vliet, 2000; ANSI/IEEE, 1989),
4) inspection aids, including checklist (Ackerman, 1989; Fagan, 1976; Gilb and Graham, 1993; Chernak, 1996; NASA, 1993), scenarios (Porter and Votta, 1994, 1995; Gough et al., 1995; Fusaro et al., 1997) and perspectives (Basili et al., 1996; Laitenberger, 2001; Shull and Basili, 2000), and
5) prescriptions documents for organizing and conducting inspection including instruments used during inspection (e.g. inspection procedure and structure (Gilb & Graham, 1993; Strauss & Ebenau, 1994; Porter and Votta, 1997), and defect forms for recording defect (Freedman & Weinberg, 1990)).

Implicit inputs include reviewers’ expertise, and behavioral aspects (norms, beliefs, values etc) (Nonaka, 1995) of reviewers. In fact, Sauer et al (2000) theorizes that expertise is a key driver of inspection performance. The defect detection process consists of individual preparation and a group meeting (Fagan, 1976, 1986; Votta, 1993; Basili, 1996). The typical defect detection duration is between two and three hours (Fagan, 1976; Doolan, 1992; Kelly et al., 1992; Porter, 1994; Johnson and Tjahjono, 1997). At the completion of defect defection, there are two types of outputs: the inspected software artifact, and quantitative outcome such as defect information recorded in defect forms.

Figure 1: A conceptual ‘EIIO’ Model

©Copyright. All rights reserved by Author.
Page 2
To our knowledge, there is no published evidence on what and how inspection inputs are used during defect detection. Neither are there consistent prescriptions of what inputs are necessary for the effective inspection of different software artifacts. For example, in a code inspection, are the requirements and design documents both present during defect detection? Are inspection outputs affected if only one of these documents is present or available? Are different inputs optimal for the inspection of different software artifacts? Laboratory experiments conducted mainly manipulate the following factors:

1. process structures\(^2\) (Fagan, 1976; Parnas and Weiss, 1985; Bisant and Lyle, 1989; Martin and Tsai, 1992; Knight and Myers, 1993);
2. use of inspection meeting\(^5\) (Votta, 1993; Johnson and Tjahjono, 1998; Porter et al., 1995),
3. reading techniques\(^4\) (Gilb and Graham, 1993; Fagan, 1976; Porter and Votta, 1994; Fusaro et al., 1997; Basili et al., 1996; Laitenberger, 2000);
4. use for roles in meeting support\(^5\) (Fagan, 1976; Ackerman et al., 1989; Russell, 1991; Land, 2000);
5. team size (Fagan, 1976; Owen, 1997; Bisant and Lyle, 1989; Porter, 1997; Kelly, 1992);
6. computer support\(^6\) (Johnson, 1997; Mashayekhi et al., 1993; Murphy and Miller, 1997; Vermunt et al., 1998), and
7. re-inspection\(^7\) (Ebenau and Strauss, 1994; Emam and Laitenberger, 2001; Eick et al., 1992; Votta, 1993; Wohlin and Runeson, 1998; Biffl, 2001).

We are especially interested in determining what key input factor(s) would significantly influence inspection output(s) in a typical 2 staged software defect detection process, i.e. individual preparation followed by group meeting (Fagan, 1976). The current inspection literature is lacking in empirical evidence on what inputs are important because there is little manipulation of this variable. Where inputs are manipulated the results are conflicting and inconsistent (Porter and Votta, 1998; Cheng and Jeffery, 1996; Biffl, 2001; Basili et al., 1996; Land 2001). For instance, Cheng and Jeffery (1996) found that the average net meeting gain is greater than average net meeting loss (i.e. net gain of the group meeting is approximately 12%) while Porter and Votta (1998) found that the net meeting gain rates are not much different from 0 (i.e. average net meeting gain is \(-.9 \pm 2.2\)). Hence what and how inputs are used in software inspection is still an open question. Practitioners’ reports (Davis, 1982; Pendlebury, 2001; Mathiassen, 2000) already show that requirements documents are often lacking or missing during later stages of software development. Different input requirements directly affect how the inspection is organized. Thus the study of inspection inputs is interesting and important for both researchers and practitioners.

\(^2\) How should the inspection process be organized and conducted.  
\(^3\) This refers to the controversy regarding the need of inspection group meetings for all types of inspection.  
\(^4\) The technique allows inspectors to follow the inspection aid in maximizing defect detection (e.g. checklist)  
\(^5\) Inspection team members can be assigned to different roles (e.g. moderator, author, reader, inspector etc.)  
\(^6\) This refers to computer-supported defect detection.  
\(^7\) This refers to the technique to capture remaining defects.
**Methodology**

We will employ the *survey* method to address the following research questions.

1. What are the common types of inspection inputs that are widely used and which appear to be significant in the software industry?

2. What are the key factors influencing the use of these inputs?

The overall objective of this study is to investigate the common types of inspection inputs in software industry practice and to identify the key factors influencing use of these inputs. We are particularly interested to investigate the following sub-questions:

1. What are the common inputs?
2. What are the key factors influencing the use of these inputs?
3. How do practitioners use these inputs during defect detection?
4. Why are these common inputs used?
5. Why are these common inputs important?

There are 12 stages in this survey plan (see table below) and we are currently working on stage 5 according to this plan.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Exploratory study</td>
<td>Investigate the common types of explicit inputs. This includes:</td>
</tr>
<tr>
<td></td>
<td>1. Secondary data analysis (Literature review)</td>
</tr>
<tr>
<td></td>
<td>2. Key information survey (FtF interview with experts in the industry)</td>
</tr>
<tr>
<td>2. Hypotheses formation</td>
<td>Define variables, objectives, questions and hypotheses</td>
</tr>
<tr>
<td>3. Research design</td>
<td>Cross-sectional survey</td>
</tr>
<tr>
<td>4. Research method</td>
<td>Web-based self-administered questionnaire</td>
</tr>
<tr>
<td>5. Questionnaire design</td>
<td>Mixture of open-ended and closed-ended questions to investigate implicit and explicit inputs, respectively</td>
</tr>
<tr>
<td>6. Pre-test</td>
<td>1. Pilot study on the paper (test the instruments)</td>
</tr>
<tr>
<td></td>
<td>2. Pilot study on the web. (test the research method and instruments)</td>
</tr>
<tr>
<td>7. Sampling</td>
<td>1. Randomly selected sample by using clustered and/or stratified sampling techniques.</td>
</tr>
<tr>
<td></td>
<td>2. Target population – we are particularly targeting software firms under computer service category in Australia and worldwide. According to the Australian Bureau of Statistics (2000), there are 14731 businesses under computer services category at the end of June 1999.</td>
</tr>
<tr>
<td></td>
<td>3. Sample frame: 300 software firms will be selected in Australia (The Business Who’s Who of Australia, 2001)</td>
</tr>
<tr>
<td></td>
<td>4. Sample Units: number of sample units to be confirmed, it will be approximately 100 or 10% sample units of total population would be</td>
</tr>
</tbody>
</table>

8 Categorized size of firms can be based on Australian Bureau of Statistics.
considered significant (Alreck, 1995)

<table>
<thead>
<tr>
<th>8. Data gathering</th>
<th>Through CAESER(^9) and ISERN(^10) in Australia and around the world.</th>
</tr>
</thead>
</table>
| 9. Data analysis  | Software: SPSS  
1. Descriptive analysis (mean, mode, medium, percentage, range, standard deviations)  
2. Correlation (Spearmen rank-order\(^11\))  
3. Comparison (Mann-Whitney U test\(^12\), chi-square\(^13\), t-test and analysis of variance (ANOVA)) |
| 10. Draft Report  | The final report includes:  
1. Research Focus (objectives and contributions)  
2. Research Methodology  
3. Research Instruments  
4. Sample (target population, sample size)  
5. Psychometrics (reliability and validity)  
6. Result and Analysis (appropriate tables, charts will be generated for reporting)  
7. Discussion (further analysis if needed)  
8. Conclusion (comments, recommendations for further study) |
| 11. Post-survey investigation | In-depth interview with the selected samples to discuss the results from the survey. |

**Expected Outcomes**

Major outcomes from this research include:

1. A conceptual model to identify inspection inputs. This will be developed from the literature review.
2. A survey to identify common and the key factors influencing the use of inputs for software inspection in the industry.
3. Multi-case analysis to investigate why the key factors influence the use of inputs
4. A modified conceptual model based on empirical data.

**Contributions**

The major contributions expected from this work are threefold. Firstly, this work increases our understanding of how the typical inspection process uses inputs in practice. Secondly, this work...
will identify the key factors influencing the use of inputs. Thirdly, a conceptual model helps to understand software inspection from inputs-outputs perspective.
9. References


CAESER (Centre for Advanced Software Engineering Research), University of New South Wales in Australia. URL: http://www.caeser.unsw.edu.au/ (access on 13 Oct. 2001)


