Process Maturity and Inspector Proficiency: Feedback Mechanisms for Software Inspections

Thomas Lee Rodgers
TRodgers@cmi.arizona.edu

Douglas L. Dean
DDean@cmi.arizona.edu

University of Arizona, Tucson, Arizona

Abstract

Recent research suggests that people-related issues account for as much as 50% of explained sources of variation in software code inspections and that process related issues account for less than 30%. This paper examines the impact of two feedback mechanisms (process maturity and inspector proficiency) on software inspections. Results of a survey of thirty-one experienced software developers and follow-up interviews are presented. Key findings include that significant process variations appear to exist within relatively mature organizations and that teams own inspection processes. The paper also addresses the significance and potential of feedback mechanism research. The findings extend to other formal technical reviews and provide insights into managing different-place, different time inspections.

1. Introduction

"Understanding the Sources of Variation in Software Inspections" is the topic of a recent research article by Porter, Siy, Mockus and Votta [13]. They find that about 50% of variation in defect detection relates to people input factors (reviewers, coders and code units) and very little variation relates to process treatment factors (team size, type of session, and repair strategy). The implication is that the "real beef" is in people-related and not process-related factors. While such assertion has intuitive appeal, further definition and exploration of underlying factors is warranted.

This paper explores perceptions of experienced software inspectors concerning two feedback mechanisms, process maturity and inspector proficiency. Similar to control mechanisms in state machines such as speedometers or gauges, feedback mechanisms measure both causes and effects. For process maturity, the focus is at the inspection team level and not at the overall organizational level. For inspector proficiency, the focus is on individual ability and motivation to work effectively during an inspection. Properly defined, the two mechanisms represent major components of people and process related factors. Is it possible to measure and use feedback mechanisms to increase effectiveness and efficiency of inspections? Although this paper does not completely answer this question, it provides a foundation for further exploration.

Recognition of feedback mechanisms should enable better allocation of (people) resources and assignment of (process) tasks. The expected benefits include improved quality, increased productivity, and faster time-to-market. Within distributed contexts, feedback mechanisms should enable different-time, different-place work processes.

2. Previous Research

During the past twenty-two years, much has been written about inspections and various types of formal technical reviews. In a 1976 article within the IBM Systems Journal, Michael Fagan described formal inspections of design and of code that are now referred to as Fagan inspections. Fagan asserted "substantial net improvements in programming quality and productivity " [5]. In subsequent years, over four hundred academic and practitioner articles have been written about inspections. Two teams of widely recognized "how-to" experts are Tom Gilb and Dorothy Graham [7], as well as Robert Ebenau and Susan Strauss [3]. Inspections are now integrated into many software engineering practices including the Capability Maturity Model [10] and Personal Software Processes [9]. Inspections have been widely adopted by world class organizations including AT&T [6] and Hewlett Packard [8]. In short, inspections are widely recognized as a means for increasing software quality.

In spite of technology advances, little has changed concerning format and content of inspections. Fagan inspections are a six-step process consisting of planning, overview, preparation, inspection, rework,
Participants have pre-defined roles including moderator, scribe, inspector, and source author [4]. Although similar to Fagan inspections, Gilb inspections add defect prevention and process improvement objectives [7]. Within IBM, Fagan inspections were the basis for defect prevention efforts [12] and subsequently for orthogonal defect classification [2].

During recent years, inspections remain a topic of continued research interest. Lawrence Votta, from AT&T, suggests that the value of an inspection comes from preparation and that need for face-to-face meetings might be diminishing [20]. Adam Porter, from the University of Maryland, finds that people-input factors are most important [14], and that traditional checklist and ad hoc review approaches are ineffective [15]. Based on industrial experience in Europe, Michiel van Genuchten finds that inspection defect yields improve significantly when using an electronic meeting system [19]. Recently, Philip Johnson, from the University of Hawaii, envisions radical inspection changes on the horizon including method-specific inspections, minimal meetings, defect-correction emphasis, organizational guideline knowledge bases, outsourcing review, computer-mediation, and review mega-groups [11]. Do advances in technology including object-orientation and wider information access call for radical changes in inspections? Technology can enable wider participation in asynchronous settings; however the question remains as to whether the inspection process needs to be reengineered.

Feedback mechanisms are widely recognized within engineering literature and to a lesser extent within the inspection literature. Defect prevention efforts are based on casual analysis which is considered a feedback mechanism [12]. Similar causal analysis form the basis for the second level of Personal Software Processes [9]. Analyzing prior defect patterns helps focus inspections and remove recurring errors.

This paper focuses on two feedback mechanisms that have not been considered in prior research. They were initially recognized in a conference paper and shown in the causal model on the following page [17]. The model suggested that inspection processes are cognitive processes in which people (players) are allocated along with causes in order to identify defects. The two feedback mechanisms both cause and affect the cognitive processes. The conference paper posits that feedback mechanisms help explain sources of variation in inspection processes. This paper explores the assertion by surveying experienced inspectors.

4. Research Objectives

As a beginning to a study of feedback mechanisms, this paper explores process maturity and inspector proficiency. The primary issues are how to define and measure both constructs. Two specific research objectives are considered.

Research Objective #1: Does process maturity significantly affect results of software inspections?

Maturity is the condition of being fully-grown or developed. Inspection processes can be considered more mature as they become established, predictable, and ultimately managed. Process maturity does not necessarily imply formality or predictability. Formality is unnecessary if participants understand their roles and expectations. A process can be mature in the sense of being well established, yet ineffective and lack predictable outcomes. For example, a mature process would be one in which the participants know exactly how many defects they are expected to find and are given opportunity to manipulate the results.

Three distinct over-lapping phases of process maturity are shown in the following graph. The first is a process definition phase in which a team forms a consensus about how and what to do during the process. During this first phase, the team is learning and establishing the process. The second phases focuses on productivity and doing the process effectively. The third phase focuses on proficiency and process efficiency. During this later phase, the team realizes that the process must be more selective because the issues are more complex.
An alternative method of representing process maturity is using a framework similar to the Capability Maturity Model. [10] which posits five progressive levels: ad hoc, repeatable, defined, managed, and optimized. Although intended for organizational assessment of key process areas, CMM levels can be adopted to inspection processes. Level one, ad hoc, applies to new or infrequent use of inspections within a professional business environment. Level two, repeatable, refers to inspections conducted on a somewhat regular basis. Level three, defined, refers to repeatable inspections in which the process, standards and defect types are well understood. Level four, managed, refers to defined inspections in which the number of major issues to be found is anticipated based on experience. Level four, optimized, refers to managed inspections which is or can be tailored to find specific types of defects or the results of which can be manipulated by the reviewers. In part, this paper explores whether such a classification can be intuitively perceived or predicted based on inspection practices.

Research Objective #2: Does inspector proficiency significantly affect results of software inspections? If so, should inspector proficiency be assessed formally or informally?

Proficiency refers to the ability to do something well. Inspector proficiency refers to the ability to find defects in another person’s work. This paper explores perceptions concerning whether inspector proficiency is important and how acceptable it is to formally record proficiency information.

Prior knowledge of inspector proficiency has value when inspection teams are formed. For example, assume use of a scenario based code inspection in which three inspectors are assigned different roles as suggested by Porter, Votta, and Basili [13]. The first reviewer focuses on data type consistencies including coding and documentation standards; the second focuses on incorrect functionality; and the third focuses on ambiguity or missing functionality. Each assignment is progressively more difficult and may suggest assigning individuals with different skill levels to each role. The first inspector might be a junior programmer for which the review process serves as a learning experience. The second inspector might be a programmer or analyst with proficiency in matching required functionality with existing code. The third inspector should be the most proficient and capable of discovering unexpected and ambiguous defects. Is it necessary to identify the levels of inspector proficiency formally? Is it sufficient to formally identify proficient inspectors with recognized areas of expertise? This paper explores whether perceptions are more acceptable than formally recorded performance.

Broader research objectives are to identify feedback mechanisms within primary processes and to determine their impact on collaborative processes. Although not precisely defined, primary processes are integrated within normal and recurring practices. Michiel van Genuchten used the term “primary process” to illustrate technology diffusion of collaborative systems [18]. First generation collaborative systems focused on research interests and established the importance of factors such as anonymity, process losses and process gains. Second generation focused on widespread organization diffusion by consultant and facilitators using commercially available software such as Ventana’s GroupSystems and IBM’s Lotus Notes. Third generation will focus on primary processes that support workgroups engaged in recurring activities. Genuchten identified software inspections as a primary process and a rich area for exploring collaborative issues. This paper explores whether two feedback mechanisms are important within the inspection process. If significance is established, a broader question is whether feedback mechanisms have importance for other collaborative primary processes.

5. Method

This paper is based on observation of experienced inspectors through survey and qualitative interviews. The findings also help develop a theoretical foundation for the study of collaboratively enabled software inspections.

The three-part survey is available on request. The first section covers background information including years of relevant experience and number of inspections during the last year and over the career. The second section covers process maturity related issues. The third section covers inspector proficiency related issues.

Three experienced inspectors reviewed a preliminary version of the survey. Based on their suggestions, changes were made, including replacing the term “reviewer” with “inspector” and adding question choices such as an expectation to "learn the inspection practice of the company".
This survey is primarily an exploratory devise. Most questions are open-ended or contain open-ended answer choices. Only one question (#10) measures a single construct (process maturity using a CMM framework). All other questions explore perceptions about aspects of the two primary constructs, process maturity and inspector proficiency.

The process maturity section contains twelve questions. Question #8 solicits the type and team composition of the participant’s last inspection. Question #10 solicits an assessment of the process maturity using a CMM framework. Questions #18 and #19 solicit open-ended discussion about the impact and variability of process maturity. The remaining eight questions cover nine different aspects: (1) expected accomplishments, (2) type of inspection, (3) inspector roles, (4) process formality, (5) volume inspected, (6) team composition, (7) type of defects, (8) number of anticipated defects originally expected to be found, and (9) number of defects actually found. For question #17 (number of defects) choices are given to indicate that the quantities are unknown.

The inspector proficiency section contains five questions. Questions #23 and #24 solicit open-ended discussion about past performance and formal recording of proficiency information. The other three questions use seven point Likert scales to assess perceptions. Question #20 asks how important various factors are to the inspection proficiency construct. Question #21 asks whether specific metrics indicate proficiency. Question #22 asks whether perceptions about proficiency should be considered when inspectors are selected and assigned to inspections. Most question choices are closely associated and labeled with similar wordings between the three questions. The choices are loosely categorized into (1) experience, (2) productivity, (3) review rate, and (4) efficiency.

6. Results

This section summarizes survey results. First, it establishes that experienced inspectors participated in the survey. Second, quantitative survey responses are summarized. Last, factor analysis is used to identify underlying dimensions of the two primary constructs.

6.1. Survey Participation

Thirty-one experienced software developers were surveyed. Two participants were interviewed concerning survey contents and specific issues suggested by the tabulated results. Survey participants represent eleven organizations ranging from small software development teams to large international firms.

- Thirteen represent an international hardware and software development firm with a reputation for maturity (CMM level 4).
- Six represent small United State software development firms with international markets.
- Four represent European software development firms most of which have multinational development efforts.
- Eight are graduate students who also work for software development firms. A large international hardware and software development firm employs five of the eight students.

On average, those surveyed have 12.6 years of software development experience and 12.2 years of employment with their current employer. Individuals were asked to participate based on having software inspection experience. On average within the last twelve months, they participated in 19.8 code reviews, 21.6 design reviews and 4.5 other formal technical reviews. Out of sixty individuals asked to participate thirty-one responded (with a 52% return rate). The rate was significantly higher for the twenty-three individuals that were directly asked to participate (with 18 responses and a 78% return rate). This compares to thirty-seven individuals who were asked to participate by a quality/process leader within a single large international firm (with 13 responses and a 35% return rate). In summary, the individuals surveyed have significant relevant experience.

6.2. Survey Responses

Detailed survey responses are available upon request. Major results are presented.

Significant process variations exist within mature development organizations. The thirteen individuals from the same international firm stated that inspection practices for their last inspection process ranged the entire spectrum of process maturity (2 ad hoc, 1 repeatable, 4 defined, 3 managed, and 3 optimized). The volume reviewed was bipolar with six reporting that they last inspected less than 50 lines of code or 5 pages and the other seven reporting that they inspected over 4,000 lines of code or 25 pages. Only three had expectations about the number of major issues to be found and only five reported the actual number of issues uncovered. Only one reported a significant number of issues uncovered (141 total issues of which 54 were major and 25 majors were anticipated based on inspection of 5,000 lines of code).

Following are quantitative results for all survey participants. The majority surveyed expect to find general defects (29 of 31) and specific types of defects (19 of 31). Additional expectations include requirement assurance, product usability, and...
verification of known defect fixes. Similar to the findings reported for the single firm, last inspection process ranged the entire spectrum of process maturity (6 ad hoc, 9 repeatable, 9 defined, 3 managed, and 4 optimized). For most, inspectors came from the same development team (22 of 31) and the same project (20 of 31). Methods for assigning inspector roles varied widely (8 informally, 14 to the same tasks, 8 to different main responsibilities, 2 involved only one inspector, and 1 used a checklist). For most, written standards and guidelines exist (17 of 31); however the formality of the process definition was subject to wide variations (10 informally defined, 11 written processes, 8 defect codes and type classification, 14 checklists, 1 script, and 2 scenarios). Most worked with individual inspection members on similar reviews numerous times previously (28 of 31 with 7 reporting an average of 2.4 times during the last twelve months).

The preceding table summarizes perceptions about whether inspector proficiency can and should be assessed formally, informally, or not at all. Surveyors were asked to use a seven-point Likert scale (with 1 = no importance, 3 = some importance, 5 = important, and 7 = extremely important and expected). The responses are ranked from most important down to least important.

6.3 Factor Analysis

Quantitative answers were used to assess underlying dimensions of the two primary constructs. Five different types of quantitative answers are included or derived from the survey. Multiple choice answers are treated as qualitative variables and coded with a nominal value (true="1" or false="0"). Likert scale answers are coded with the corresponding ordinal value (from 1 to 7). Numeric counts (such as the number of inspections or defects expected) are coded with ratio scales; however these counts have little value within the following analysis other than to indicate whether the quantity is known or equal to zero. Instead, aggregated ordinal scales (unknown = "(1)", zero value = "0", and positive value = "1") are used to characterize numeric count questions. Finally, aggregated counts of nominal values are used for several questions with similar answer choices and for which a count of similar choices is presumed to infer a stronger relationship (such as within question #11 to distinguish whether the reviewer comes from the same project, company, division, or development team).

Given thirty-one participants and seventy-one quantitative answers, it is difficult to make statistical inferences or generalize findings. A larger sample size would provide greater statistical power; however, given selective participation in the survey, large sample sizes might not enhance the ability to generalize findings. For statistical analysis, the number of factors to be analyzed is limited by the sample size of thirty-one participants. In addition, using a number of factors approaching the sample size prevents obtaining some standardized statistical metrics such as Cronbach's alpha. Nevertheless, given the exploratory nature of the survey some insight can be gained by exploring latent factors.

Process maturity construct measurement is more complex than participant perceptions. Attempts to use regression to predict the process maturity level (from question #10) resulted in models with very low statistical significance (adjusted R2 < .15) and the only factor having some significance is the number of defects expected (probability value = .032). As previously stated, participants within relatively mature organizations perceive wide variations in inspection process maturity. During a follow-up interview, it was suggested that the classification wording might imply "goodness" of personal practices and thus is difficult to assess. Inspector proficiency is even more difficult to measure and subject to measurement dysfunction [1] as illustrated in the following participant quotes. "Yes, although this is difficult to quantify." "People will be less open about the inspections and their results." "People will not log if they think management will base performance evaluation of author on the results." "Counting is ridiculous especially if you have to justify what you find is over or under an expected amount."

Using a heuristics based on inspection practices might be more stable than using perceptions of maturity level. A heuristic was adopted based on the CMM assessment method. The heuristic requires demonstration of competence in all key process areas at the current and lower levels before being designated at that current level. For analysis purposes, qualitative answers were aggregated by how relevant they were to each process maturity level. Process maturity level was determined based on the presence of at least one indicator within the level and the lower levels (with the exception of level one - ad hoc for which no key process area is assumed). The only apparent difficulty with the heuristic occurred with an indication of a level five (optimized) process and none for level four (managed). In this case, the process was classified as a level five. For the thirteen participants from the single firm, the distribution changed significantly. Ten of the thirteen participants indicate either level four or five (which is more similar to their organization's level 4 CMM designation). Three participants indicated a
level two (repeatable). Perhaps, process maturity should be assessed based on inspection practices within key processing areas in a manner similar to CMM.
In order to be a proficient inspector, how important are the following?

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>programming language proficiency</td>
<td>6.1</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>(for code inspections)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cognitive ability to find defects</td>
<td>5.9</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>experience as an inspector</td>
<td>5.3</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>development environment proficiency</td>
<td>5.2</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>experience as an author being reviewed</td>
<td>4.1</td>
<td>1.7</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Assuming the following information can be recorded or derived about prior individual performance, how acceptable is it to use the following information to select and assign inspectors to an inspection team?

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>experience (number of prior inspections)</td>
<td>5.0</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>efficiency (percent of defects found per defects suggested)</td>
<td>4.5</td>
<td>2.2</td>
<td>2.2</td>
</tr>
<tr>
<td>experience (pages or lines of code inspected)</td>
<td>3.9</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>review rate (pages/lines of code reviewed per hour)</td>
<td>3.7</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td>productivity (number of defects per page/lines of code)</td>
<td>3.7</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>productivity (defects found per preparation hour)</td>
<td>3.6</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td>productivity (defects found per meeting hour)</td>
<td>3.5</td>
<td>1.9</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Assuming the inspector proficiency information is not formally recorded or available, what factors should be considered when inspectors are selected and assigned to inspections?

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>experience with programming language</td>
<td>5.5</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>experience as an inspector</td>
<td>5.1</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>productivity as a person who can find defects</td>
<td>5.1</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td>efficiency as a finder of defects</td>
<td>5.0</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>experience with the development environment</td>
<td>4.9</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>review rate as a through and detailed inspector</td>
<td>4.3</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>experience as an author of reviewed materials</td>
<td>4.0</td>
<td>1.7</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Table 1 - Ranking of Inspector Proficiency Perceptions

Exploratory factor analysis reveals significant underlying dimensions. Factor analysis was used for both major constructs. For process maturity, analysis was done at summary and detailed levels. For inspector proficiency, analysis was done at the detailed level. The factors identified might not generalize; however, they give insight into perceptions of those surveyed.

Using summary level analysis, process maturity contains two major dimensions (with Cronback’s alpha = .63). The first relates to process formation based on defined roles, process formality, and team interactions. The second relates to anticipated and actual results based on known volumes inspected and number of issues found. However, aggregated coding might cause distortion. Therefore, detailed level analysis was also conducted.

Using detailed level analysis, process maturity contains nine significant dimensions. Each dimension is briefly described along with significantly positive and negatively correlated variables with larger than a ± .40 correlation. (1) Immature process relates to an expectation of establishing standards and unknown numbers of issues expected or actually found. (2) Informal process relates to informally defined role, a perception of ad hoc level one maturity level, and working together previously. Informal processes are negatively related to written processes, standards, guidelines, working on similar reviews, working on the same discovery task. (3) Specific or targeted issues relate to an expectation of finding specific issues, reviewing an unknown volume, finding an unknown number of issues, and working within the same development team. (4) Corporate practices relate to working within the same company or development
team, and written standards, defect codes and types. Corporate practices are negatively related to an expectation of establishing standards. (5) **Recurring issues** relate to an expectation of finding general defects, use of checklists, and written standards and guidelines. Recurring issues are negatively related to looking for unknown defect types. (6) **Issue recognition** relates to first-time experiences, use of scenarios, written defect codes and types, and perception of managed level-four maturity level. (7) **Learning expectation** relates to an expectation of learning and assignment to different main tasks. Learning expectation is negatively related to working within the team and assignment of the same task. (8) **Project formality** relates to working together, with reviewers from the same project, using a written process, and not being a first-time experience. (9) **Non-recurring issues** relate to looking for pre-designated issue type and a perception of an optimized level-five process maturity. The first factor (immature process) from the detailed analysis is similar in nature to the second factor (anticipated and actual results) from the summary analysis. Also the remaining eight factors from the detailed analysis appear to be dimensions of the first factor (process formation) from the summary analysis.

Using detailed level analysis, inspector proficiency contains five significant dimensions. (1) **Problem solving skills** relate to the perception of being a finder of defects through productivity, review rate, and efficiency, as well as having a high percentage of defects found per defects suggested. (2) **Inspection productivity** relates to the prior individual performance in terms of the number of defects found in total, in preparation, and during meetings. (3) **Inspection experience** relates to prior inspection experience whether perceived or actually counted and as either an author or inspector. (4) **Programming skills** relate to the importance of programming language proficiency, and perceptions of experience with the programming language or development environment. (5) **Environmental awareness** relates to the importance of development environment proficiency and the cognitive ability to find defects.

7. Discussion

This section discusses the stated research objective of exploring whether the two feedback mechanisms significantly affect software inspection results. This paper does not prove the research objectives; rather it explores perceptions of experienced inspectors. Nevertheless, the findings generally support the research objectives and suggest additional avenues of inquiry.

The first objective is to determine whether process maturity significantly affects results of software inspections. Question #18 most directly addresses the stated objective. This open-end question asks what impact, if any, does process maturity have on inspections? More specifically, does the nature of the inspection change as the process matures? Twenty-eight of the thirty-one participants responded to the questions. Only five of the responses suggested minimal or no impact. **Most of the remaining comments supported the assertion that process maturity impacts inspections.** One of the more interesting lines of reasoning was established in a follow-up interview. The participant asserted that the team composition is a given constraint, and “although people are more important, you must manage the process.” Another insightful comment follows, “This is a soft issue; if the organization has experience, culture, and willingness to work along defined routes, inspections are taken seriously and actually carried out and followed-up. In a low maturity organization it is difficult to actually prove the effects of inspections and they tend to get postponed or canceled.”

**Measuring process maturity presents problems.** As previously documented, use of a CMM framework might provide a more consistent assessment, especially given establishment of key practice areas. Can process maturity be assessed based on overlapping phases such as definition, productivity, and proficiency? In this alternate phase-related framework, the category is not as important as understanding the support requirements for each phase and identifying how much associated support is required for the current inspection. One quote supports the notion that process mature in a non-linear manner, specifically “Our process matures chaotically.” As shown in the preceding table, factor analysis and previous qualitative observations can be categorized within such a framework. Such categorization serves as a foundation for further defining and exploring each phase. Whether three phases is sufficient or necessary remains unresolved.

A potentially important aspect of process maturity is the inspection review rate. Most experts consider review rates as very important. Inspection expert, Tom Gilb states, “check at your organization’s optimum rates to find major defects” (Gilb, 1998). Michiel van Genuchten finds that review rate must be controlled in order for electronic meeting system inspections to be more effective than traditional Fagan inspections. He also finds that very few defects are found for either method when the review rate is too...
fast. Survey participants appear to place relatively little emphasis on review rates. Use of review rate is given a relatively low 3.9 ranking with a relatively large 1.9 standard deviation (see Table 1). The observation is similar for the related issue of perceived thoroughness that is given a 4.3 ranking and a 2.0 standard deviation. The only quote related to thoroughness and indirectly to review rates follows. "You want people who spend time and find problems."
<table>
<thead>
<tr>
<th>Factor Analysis</th>
<th>Definition Phase</th>
<th>Productivity Phase</th>
<th>Proficiency Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process formation</td>
<td>Anticipated and actual results</td>
<td>Specific or targeted issues</td>
</tr>
<tr>
<td></td>
<td>Immature process</td>
<td>Corporate practices</td>
<td>Issue recognition</td>
</tr>
<tr>
<td></td>
<td>Informal process</td>
<td>Recurring issues</td>
<td>Non-recurring issues</td>
</tr>
<tr>
<td>Qualitative Responses</td>
<td>Team ownership</td>
<td>Inspection rates</td>
<td>Problem-solving skills</td>
</tr>
<tr>
<td></td>
<td>Training</td>
<td>Workload</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project commitment</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 - Categorization of Factors and Responses by Process Maturity Phases

Apparently, this is an issue that many inspectors either do not remember or have forgotten the importance.

Further, process maturity should focus on how-to support teams. Teams own inspection processes as supported in the following quotes. "Our goal is to find defects ourselves as a development team, not to create professional super-inspectors. Every member should inspect once in a while." "The team typically knows who are the good inspectors. The data about it should be owned by the individual or the team." Follow-up interviews supported this assertion. One participant observed that there are at most ten people in the world who have sufficient expertise to inspect their code. This observation appears to apply for both large and small organizations. In large organizations, inspection responsibility falls on development teams that usually consist of ten or less members. In small organizations, there might only be a few developers in the entire organization.

Process maturity is directly affected by workload and project commitments. This assertion is supported by the following quotes. "We barely have time for reviews, much less all the formalization." "Our environment is changing given shortened time-to-market constraints. We need to tailor processes to fit the life cycle." "Don’t include a person that is not motivated to participate.” Do not "increase overhead." "Inspection is not a popular pastime. Many of us find it drudgery even though it is of critical importance to quality. People may decide to find fewer defects if they know the more they find, the more review time they must spend."

As processes mature, opportunity must be given for training of new team members and transferring institutional knowledge. This assertion is supported in the following quotes. "Others ought to be included to improve their skills.” “Other members can be in training.” “Everybody needs to be in the game.” For an organization with responsibility for maintaining large amounts of legacy code, the observation was made that inspections provide a significant source for training and knowledge transfer even when few defects are found. Individuals who maintain the code will eventually retire and inspections provide a structured opportunity to transfer knowledge from senior to junior level team members.

The inspector proficiency research objective has two parts. The first is to establish significance and the later addresses whether proficiency should be assessed formally or informally.

There appears to be support for the assertion that inspector proficiency affects results of software inspections. Question #23 most directly addresses the stated objective. This open-end question asks whether a person’s past performance should be considered when an inspection review team is formed? If so, how? If not, why not? Only three of the twenty-eight responses suggest that inspector proficiency does not affect the results and one of these merely state concerns about privacy and formal recording of performance information. However, the tenor of the discussion is
supportive. Two supportive quotes follow. “The team typically knows who are the good reviewers.” “Only certain people can be considered good reviewers, or can learn to be good reviewers.”

A significant aspect of inspector proficiency is problem-solving skill. Both quantitative and qualitative supports exist for this assertion. As previously stated and based on factor analysis, problem solving is the most significant proficiency factor. The factor relates to the perception of being a finder of defects through productivity, review rate, and efficiency, as well as having a high percentage of defects found per defects suggested. The qualitative support comes from the following quotes. "Are they good engineers?" Inspectors need a "knack" and ability to "see to the heart of the problem." "At least one person who can find defects is a must to make the review worthwhile." "Only certain people can be considered good reviewers, or can learn to be good reviewers.”

The issue of formally recording inspector proficiency information appears to depend on the organizational culture. Question #24 asks what concerns, if any, do you have about formally recording inspector proficiency information. The thirty responses, out a possible total of thirty-one, are divided into ten strongly supportive comments and thirteen strongly opposing comments. Many expressed concerns like “it is valuable information for performance evaluation. But I don’t think it needs to be a formal process to record. A close working relationship among the team members would yield better information.” The assertion that the response depends on the organizational culture is best stated in the following quote. “As with all data concerning a person, it can be used constructively or destructively. Whether formal recording should be used is a function of the person’s organization personality.” This finding is also supported by Robert Austin’s research [1].

Perceptions about individual performance might be preferable to formal recordings of actual performance. Maintaining formal records adds overhead, forces formally defined metrics and raises privacy concerns. Can perceptions sufficiently measure individual proficiency? Most individuals are painfully aware of their inadequacies. To formally record painful information might not be advantageous. Also, most team members have a sense of who to rely upon within their team. The assertion is supported by the following quotes. "If you can record the original error maker, you can record who can find the error.” But recording "can hurt a person’s privacy” "Sure, if I know that they are not checking details, I probably won’t use them." "It is valuable information for performance evaluation. But I don’t think it needs to be a formal process to record. A close working relationship among the team members would yield better information." "Don’t waste time doing recording.” Although not statistically significant, rankings for the second question (see Table 1 on page 12) related to formal recordings are lower than the other two questions related to perceptions.

Perhaps, the focus of inspector proficiency should shift to how-to-best support recognized expertise within an inspection process. Inspectors know when they are beginners or even productive team members who are expected to contribute. What might not be known is how to recognize expertise, and constructively use the expertise without burning it out. Thus there might be a need to manage a list of corporate or outside consulting inspection expertise. One of the follow-up interviews bought out the need for a non-team member with expertise in the Windows/NT operating system. What if the inspection was tailored to maximize knowledge transfer for specific questions asked of recognized experts?

8. Significance

This paper explores perceptions of two feedback mechanisms, process maturity and inspector proficiency. Some feedback mechanisms might be intuitive and used by teams to enhance their performance. For example, the team might sense who the good inspectors are and assign important inspection task accordingly. As Robert Austin suggests, performance measures are best left unmeasured if the team collectively moves along the best performance path [1]. The challenge is to identify the impact of feedback mechanisms or calibrate their effects.

Feedback mechanisms might impact the development of tools. Providing access to easily quantifiable factors such as number of major issues expected and actually found should have a major impact on group performance. For more difficult to quantify factors like process maturity and inspector proficiency, the question is will workgroups recognize and support their integration within a collaborative tool.

Process maturity might impact how inspection tasks are defined and shared among inspectors.
Task templates might be used to match process maturity expectations. More specifically, the tool might prompt for inspection type, objectives and anticipated inspectors. The system would then provide a base template of inspection tasks assigned to inspectors. The moderator can modify inspectors and task assignments. The tool would assess the optimum review rates and identify any potential mismatches of task and proficiency.

Inspection processes might be tailored to support expert inspectors. Perhaps the inspection tool should interface with corporate personnel skill databases or to an external cottage industry of expert inspection consultants. If a particular expertise is needed, enable the inspection team with the ability to identify and obtain appropriate expertise.

9. Conclusion

This paper explored perceptions of experienced software engineers concerning the impact of process maturity and inspector proficiency on software inspections. The findings generally support assertions that process maturity and inspector proficiency affect results of software inspections. In some cases, teams may prefer using perceptions of performance instead of formally recording actual prior performance.

Feedback mechanisms might provide a means to measure and manage performance at the workgroup level. Doing so in an unobtrusive and effective manner is a challenge. The goal is to enable teams to self-define tasks and assign of responsibilities in a manner that encourages productivity and decreases time to market. At a minimum, developing a better understanding provides insight into asynchronous work processes.

References