Using Learning Styles to Improve Cost Effectiveness of Software Inspection Teams

Anurag Goswami, Gursimran Walia
North Dakota State University, Fargo, USA
anurag.goswami@ndsu.edu, gursimran.walia@ndsu.edu

Abstract—Inspection is a widely known technique to detect faults in requirements document. It helps manager by early detection and removal of faults. Much of the cost can be saved by detecting faults during the inspections of early software artifacts as compared to testing in later stages of development. Our research utilizes a Learning Style – LS concept (coined by psychologists that refer to individuals’ information perceiving and processing traits) in the context of an inspection of software requirements document. This paper reports the results from an empirical study that, inspectors of diverse LS’s in a team are more cost effective (i.e., found larger number of faults faster with larger testing cost savings) as compared to the inspection team made up of random assigned inspectors or inspectors with similar LS’s.

Keywords—software inspection; learning style; Kusumoto Metric

I. INTRODUCTION

A successful software organization ensures software quality by minimizing the impact of faults committed during the early stages (when they are cheapest to find and fix) in order to reduce its impact in later stages. Amongst early stage artifacts, Software Requirements Specification (SRS) document records requirements in Natural Language (NL) and is especially fault prone due to inherent characteristics of ambiguity, imprecision and vagueness of NL. Among various methods used for early detection and removal of faults, software inspections have been empirically validated [1-3].

Software development works within budget and time constraints. Hence, it is imperative that software inspection teams are cost effective. Evidence shows that inspection’s success rely heavily on the extent to which individual inspector uncovers faults (i.e., their individual ability to locate defects) [2]. Generally, inspection teams are formed randomly (or based on their domain experience). However, research efforts on correlating the background knowledge and domain experience to the inspection effectiveness have not yielded any success [4]. For an inspection to be cost-effective, inspectors need to be selected in an objective manner that would minimize the fault overlap and increase the overall inspection effectiveness. We hypothesize that; an individual’s innate abilities (as opposed to the domain knowledge or technical background) are more relevant to their ability to find requirement faults.

On that note, Cognitive Psychologists provide empirical evidence that individuals have different Learning Style (LS) preferences and strengths and they perceive and process information better if information is presented in their preferred LS [5]. Software engineers also vary in the way they “perceive” and “process” the information documented in software artifacts. This is especially true in context of NL requirements document which involves collaboration between technical and non-technical stakeholders. Our earlier results validated that an inspectors with certain LS’s are more effective at finding defects in NL requirement documents [6].

This paper presents the result from an experiment that investigated the cost-effectiveness of inspection teams formed by inspectors with dissimilar LS preferences vs. similar LS preferences vs. no LS preference (by randomly selection).

II. BACKGROUND

This section provides the background information regarding inspection cost model which is used to calculate inspection cost (Section II.A). Section II.B describes the Kusumoto metric for evaluating the cost effectiveness of inspection. Section II.3 explains the LS model and LS categories used in our research.

A. Inspection Cost Model

The inspection cost model [7] has these components:
1) \( C_t \) – cost spent on an inspection, is the sum of total time taken to inspect a document by each inspector.
2) \( c_i \) – Average cost to detect a fault in testing, is not available during inspection. Therefore, it is measured as a factor of an average cost to detect a fault during an inspection. If a defect introduced at the earlier stage passes to the later stages, it requires rework which involves huge cost. Hence, it is always cost-effective to detect a fault as early as possible.
3) \( D_{total} \) – total number of faults present in the software product, can be determined if a document seeded with faults or the number of faults found by multiple inspectors.
4) \( D_r \) – number of faults detected: unique faults found during the inspection by all inspectors.
5) \( C_r \) – testing cost: cost to detect remaining faults in testing, (i.e. \( D_{total} - D_r \)) after inspection. If we consider \( c_i \) as the average cost to detect a fault in testing, then the cost can be measured as the product of total number of faults remaining after inspection and the average cost to detect a fault during testing. This is, \( C_r = (D_{total} - D_r) \times c_i \).
6) \( \Delta C_t \) – testing cost saved by inspection: by spending cost \( C_t \) during inspection, the cost \( \Delta C_t \) is saved during the testing. It is calculated as the product of the total number of unique faults found during the inspection \( D_r \) and the average cost to detect a fault in testing \( (c_i) \). That is, \( \Delta C_t = D_r \times c_i \).
7) \( C_{vt} \) – virtual testing cost, (i.e. testing cost if no inspections are performed) is the total of the cost required to...
detect faults left after inspection \( (C_r) \) and the testing cost saved by inspection \( (\Delta C_t) \). That is, \( C_{st} = (C_t + \Delta C_t) \).

**B. Kusumoto Metric \( M_k \)**

Kusumoto et al. [7] proposed a metric for evaluating the cost effectiveness of the inspection in terms of reduction of cost to detect and remove all defects from software product. It is a ratio of the reduction of the total costs to detect and remove all faults from the software product using inspections to the virtual testing cost (testing cost if no inspection is executed). It can be compared across different inspections and projects, and is deemed most appropriate for our research.

\[ M_k = \frac{\Delta C_t - C_r}{C_t + \Delta C_t} \]

\( M_k \) is intuitive as it can be interpreted as the percentage of fault rework savings due to inspections. Using \( M_k \), cost-effectiveness can also be compared across inspections on different projects. This research uses Kusumoto to evaluate the cost effectiveness of inspection teams formed using the LS’s of the inspectors (i.e. dissimilar, similar and no preference).

**C. Learning Styles Models**

Kolb [8] introduced the concept of LS’s and is recognized with the development of first LS measurement instrument. Over the years, psychologists have developed variety of LS models [9] and validated the use of LS’s in engineering education [5]. Previous researches revealed that the Felder and Silverman’s Learning Style Model (FSLSM) is the most advanced and widely used to measure the LS’s through an instrument called the Index of Learning Styles (ILS) [10]. The following section explains the FSLSM model and LS instrument used for measuring an individual’s LS.

1) Felder Silverman Learning Style Model: This study applies the FSLSM to capture most important LS preferences among individuals [5]. The FSLSM model classify characteristic strength and preferences across four LS dimensions. These dimensions related to the way individuals “perceive” and “process” information. The two dimensions which relates to perceiving information includes: a) Sensing/Intuitive; and b) Visual/Verbal. The remaining two dimensions (i.e. Active/Reflective and Sequential/Global) relate to information processing. Brief description of four dimensions of LS model is listed below:

- **Sensing** Learners (oriented towards facts, concrete content, data, careful with details, follow existing ways) or **Intuitive** learners (abstract, conceptual, oriented towards theories and meaning, discovering possibilities);
- **Visual** learners (prefer visual representations– pictures, diagrams, flow charts, video, demonstration) or **Verbal** learners (prefer written and spoken explanations);

**Fig. 1:** Example result of the questionnaire on the ILS

- **Active** learners (learn by trying things out, working in groups, discussing, explaining, brainstorming) or **Reflective** learners (learn by thinking things through, working alone, writing summaries);
- **Sequential** learners (linear, orderly, learn in small logical steps) or **Global** learners (holistic, context and relevance of the subject, learn in large jumps).

**LS of an individual is measured with the help of an instrument called Index of Learning Styles (ILS). The ILS has been empirically validated for its reliability [10]. The ILS is a score sheet of an online questionnaire with 44 questions. Each dimension has 11 questions. For example, in Visual/Verbal dimension, if a person selects 10 answers that favors visual category and 1 towards verbal category then the score will be 9 (i.e. 10-1) on the visual category represented by a symbol ‘X’ on the top of the score, shown in Fig. 1. The number of answers favored for a category is termed as Actual Scores (actual score of Visual category is 10 and for verbal category is 1 in the example above) in our research. Score ranging from 1-3 on ILS represents that a person is balanced and has preference towards both the categories in a LS dimension. A score of 5-7 and 9-11 states that the person has a moderate and strong preference towards a category in a LS dimension.

**III. RESEARCH APPROACH**

This section describes the techniques that we have used for generating inspection teams with dissimilar vs. similar LS’s for varying number of inspectors. A tool was created that takes in LS’s as input and outputs inspection teams (teams which never met) of a particular size. Section III.A explains principal component analysis which is used to convert correlated variables (i.e. LS scores) into independent ones. Sections III.B and III.C discusses cluster and discriminant analysis which are used to maximize the LS variations across different clusters.

**A. Principal Component Analysis (PCA)**

PCA is a multivariate technique that is used to convert a set of observations of possibly correlated variables into set of values of PCA is used in this research to gain a better understanding of the interrelationships between two categories (e.g., visual/verbal) of each of the four LS dimensions and between all the four LS dimensions for each individual. PCA transforms the original FSLSM output (Fig. 1) into a new set of uncorrelated variables called principal components (PCs) [12].

For each individual, the numbers of possible PC’s are always equal to or less than the number of original variables (i.e., 8 categories across 4 LS dimensions) [12]. Each PC accounts for
certain variance between the categories in each dimension; and between the dimensions. The PCA will try to account maximum possible variance with the first PC that exists in LS preferences of individual subjects. The second PC will try to account maximum possible variance that could not be explained by first PC and so on. However, it takes all possible number of PC’s to explain 100% variance of original data [11].

B. Cluster Analysis (CA)

The objective of CA in our research is to form clusters of individuals based on their LS data. The resulting clusters explain high similarity of LS’s within each cluster and high dissimilarity of LS’s between different clusters [14]. In our study, CA was used to group the participants into clusters of similar LS. These clusters helped us order members on the scale ranging from dissimilar to similar LS preferences. A team with different cluster members led to dissimilar LS group and a team from same cluster members leads to a similar LS group.

C. Discriminant Analysis (DA)

The DA is used to maximize the LS variations across different clusters, and minimize the LS variations within each cluster [16]. DA provides Group Membership (GM) to determine the dissimilarities between individual LS within the same cluster and with respect to the individuals in other clusters. Maximum GM value indicates that the individual has most similar LS when compared to the particular cluster. The dissimilarities between each individual LS’s within the same cluster could be evaluated by using the difference between the GM values of individuals. Therefore, DA delivers GM values for each individual with respect to each cluster.

DA classifies all the individuals into known clusters (that were generated by CA). GM was used to sort the teams ranging from most dissimilar LS to teams with most similar LS preferences and strengths. This process of extracting software inspection teams with varying levels of LS preferences was automated using the software tool developed for our research.

IV. EXPERIMENT DESIGN

The primary motivation for this study was to evaluate the cost effectiveness (\(M_k\)) of inspection teams formed with inspectors with dissimilar LS’s vs. similar LS’s vs. randomly selected. Table I defines the independent and dependent variables in this study. We hypothesize that, ‘Inspection teams with dissimilar LS’s are significantly more cost-effective as compared to teams of inspectors with random or similar LS’s’.

A. Artifact and Inspectors

Artifact: Participants in our study were given a requirements document (developed externally) that described the requirements for Restaurant Interactive Menu (RIM). The RIM document was 52 pages long and contained 98 faults.

Inspectors: The eleven inspectors in our study were students enrolled in the System Analysis and Design course at North Dakota State University.

B. Experiment Procedure

1) Step 1: First, at the beginning of the experiment, all the participants were given LS questionnaire. During this step, the participants answered 44 multiple choice questions (at http://www.engr.ncsu.edu/learningstyles/ilsweb.html) and, the LS results were generated on the ILS scale (see Fig. 1). The LS categories (Active, Reflective, Sensing, Intuitive, Visual, Verbal, Sequential, and Global) were measured by calculating the LS score in each dimension of ILS. Each participant has a score towards one category within a LS dimension.

These LS scores of each inspector are then converted into actual scores which has scores in both the categories. Table II represents the subset of actual scores (the number of answers supported by the participants for each LS category). As an example, Sequential/Global category has 11 questions that have to be answered by the participants. Each question has an answer that supports a category.

2) Step 2: Training and Inspecting SRS for Faults: During the training, the participants received the SRS document for the RIM system, and the fault-checklist. They were instructed on how to use the fault-checklist to record faults using a set of example requirements. Next, the subjects individually inspected the RIM requirement document using the fault-checklist and log faults that they found during the inspection.

The researchers validated that the fault reported by each participant were true positives. The researcher, who had knowledge of the system for which the requirements were developed, read through the faults reported by each participant to remove any false-positives before analyzing the data.

C. Evaluation Procedure

After gathering the LS and inspection data of 11 inspectors, virtual inspection teams were created from our tool as follows:

1) Step 1 - Virtual inspection Teams generation: We created virtual inspection teams (i.e. teams that did not actually meet, we just combined their fault and timing data) to determine the effect of LS on inspection performance. We created inspection team size ranging from 2 to 9 and each team size has all the possible combinations of virtual teams. For e.g. to create inspection team size 2, 55 inspection teams (i.e. 11C2) were created from a pool of 11 inspectors.

2) Step 2 – Grouping of similar inspectors in clusters: The correlated LS of inspectors in each LS dimension were

<table>
<thead>
<tr>
<th>Cluster No</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspector ID</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>#</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE I. STUDY VARIABLES AND HYPOTHESIS

<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Style</td>
<td>Characteristic strengths and preferences in the ways that humans process information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique Faults</td>
<td>Total # of faults found by the participant.</td>
</tr>
</tbody>
</table>

| Cost Effectiveness | Kusumoto cost metric (\(M_k\)) for each team. |

TABLE II. EXAMPLE OF ACTUAL SCORE OF PARTICIPANTS

<table>
<thead>
<tr>
<th>ID</th>
<th>Act</th>
<th>Ref</th>
<th>Sen</th>
<th>Int</th>
<th>Vis</th>
<th>Ver</th>
<th>Seq</th>
<th>Glo</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>02</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>03</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>04</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>
converted into uncorrelated variables by the tool using PCA (Section III.A). Next, inspectors of similar LS’s were grouped together in the same cluster (number of cluster is same as the team size being analyzed) using CA (Section III.B). Table III shows the cluster output of team size of 2 inspectors that would be formed by the tool. In Table III, first row represents the cluster# and the second row shows the inspector# who belongs to a cluster. In this example, for team size 2, tool forms 2 different clusters (equal to team size) with inspectors 3,5,8,10 and 11 grouped into cluster# 1 and similarly, rest of the inspectors belong to cluster 2 respectively.

3) Step 3 – Sorting teams based on the LS of inspectors: In this step, the tool calculates the group membership (GM) of each inspector in a cluster using DA, shown in Table IV. First row for each cluster (i.e. C1 and C2) represents the inspector ID number and second row shows the GM value of inspectors in their respective clusters.

Next, the tool sorts all inspection teams (i.e. 11C2, from step 1) in the order of decreasing level of dissimilarity (i.e. most dissimilar to similar) in the LS’s of the individual inspectors. The output of this step for inspection team size of 2 is shown in Table V; the most dissimilar team consists of inspector# 7 and 11. As we move down the table V, the level of dissimilarity decreases for teams with highest number of clusters involved (shown by decreasing value of total GM) until we go to team# 30. Similarly, as we move down the table for teams with lowest number of clusters involved (i.e. 1), similarity decreases. Team# 31 is the most similar team since highest probability in a single cluster represents the most similarity among the team members. This procedure was executed for all teams of 2 to 9 inspectors.

4) Step 4 – Calculating inspection performance of teams: During this step, the tool combines the individual inspection data and outputs the total unique faults and average time taken by each virtual inspection team of all sizes.

5) Step 5 – Now, from each team size, we randomly picked 10 teams from dissimilar cluster, 10 teams from similar cluster and 10 teams from the whole team set (e.g., for team size 2, its 11C2). Inspection cost model was applied on each team and the Kusumoto cost metric (i.e., their cost ineffectiveness) was calculated. The following steps were performed for each inspection team:

a) Average cost to detect a fault in inspection (c_r): adding all the faults found by inspectors, average number of faults found by an inspector is calculated. From the available values of time and average fault detected by an inspector, c_r is calculated from the following equation:

\[ c_r = \frac{C_r}{D_r} \]

(cost of inspection / total faults found during inspection)

- “C_r - Cost of inspection”, is calculated by total time spent by all inspectors during inspection.
- “D_r”, is the total number of unique faults found by all the inspectors during inspection.

b) Virtual testing cost (C_v): is the product of average cost to detect a fault in testing (i.e., c_t) and the total number of faults present in the product (i.e., D_total).

- “c_t – Average cost to detect a fault in testing”, is calculated as 6 times the average cost to detect a fault during the inspection (c_r) [17]. The average cost to detect a fault in testing (c_t) is kept constant for the evaluation regardless of the inspection team size.

- “D_total – Total fault count”, is the total number of faults present in the document. The artifact used has 98 natural faults in it.

c) Cost saved from inspection(∆C_r): testing cost saved by inspections is the product of unique faults found during inspection (D_t) and average cost to detect a fault in testing (c_t).

The difference between the testing cost saved by inspection and cost spent on inspection provides the total reduction in costs. Kusumoto metric (M_k) is then obtained as follows:

\[ M_k = \frac{\text{Reduction is cost to detect all faults (i.e., } \Delta C_r - C_r \text{)}}{\text{Virtual testing cost (i.e., } C_v \text{)}} \]

The M_k value can range from -1 to +1. The M_k value of 1 means the most cost-effective inspection. A positive M_k value indicates that cost saved from inspection outweighs the costs spent on inspection. A negative M_k value indicates a cost ineffective process, and M_k value of 0 is when the inspection cost equals the testing savings.

We computed the M_k values for 10 virtual inspections for all team sizes (ranging 2-9 inspectors) and using all three means of team formation (dissimilar vs similar vs no LS preferences). As an example, 10 M_k values (representing 10 virtual inspections) for a team size of 6 inspectors (using the dissimilar LS’s) as shown in table VI. Similar process was used to derive the M_k values for all the 10 virtual inspection teams for all three types of LS variations.

### Table IV. Group Membership for 2 Clusters

<table>
<thead>
<tr>
<th>C1</th>
<th>Inspector ID</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>0.97</td>
<td>0.96</td>
<td>0.94</td>
<td>0.96</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C2</th>
<th>Inspector ID</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>0.65</td>
<td>0.88</td>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>0.95</td>
<td></td>
</tr>
</tbody>
</table>

### Table V. Teams Formed From 2 Clusters

| Most Dissimilar Teams Set | 1 | 2 | 3 | 5 | 8 | 11 | 2 |
| Most Similar Teams Set    | 1 | 2 | 3 | 5 | 8 | 11 | 2 |

### Table VI. Calculation of Kusumoto Metric for Team Size 6

<table>
<thead>
<tr>
<th>Virtual Inspection</th>
<th>Total Defect Count (Fault)</th>
<th>Defect Found (Fault)</th>
<th>Cost of Inspection</th>
<th>Avg. Cost to detect a defect in Testing</th>
<th>Testing Cost Saved by Virtual Testing</th>
<th>Kusumoto Metric (M_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>54</td>
<td>80</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>2</td>
<td>52</td>
<td>80</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>80</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>80</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
<td>60</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>6</td>
<td>53</td>
<td>60</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>7</td>
<td>54</td>
<td>62</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>8</td>
<td>54</td>
<td>62</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>9</td>
<td>54</td>
<td>62</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
<tr>
<td>10</td>
<td>53</td>
<td>62</td>
<td>69.70</td>
<td>51.70</td>
<td>2442</td>
<td>0.43307</td>
</tr>
</tbody>
</table>

---

738
V. ANALYSIS AND RESULTS

This section utilizes the inspection data of virtual teams to compare the cost effectiveness results of teams with varying number of inspectors and various LS preferences.

For each team size (ranging from 2 to 9 inspectors), all possible virtual inspection teams were generated. The result was then sorted by teams with most dissimilar LS’s (involved largest # of clusters) to the teams with most similar LS’s (least # of clusters involved) of participating inspectors. Inspection data was calculated for each of these virtual teams. The data consists of the unique # of faults uncovered and total time taken by the team members during the inspection of RIM document. Next, for each team size, from a pool of all possible virtual inspection teams (e.g., 11C2 for team size 2), 10 teams were selected from each set of LS’s (i.e. for each team size, 10 teams of inspectors with dissimilar LS’s, 10 teams with similar LS’s, 10 teams of inspectors randomly selected). For each team, $M_i$ was calculated.

To provide an overview of our results, Fig. 2 shows the \textit{average cost effectiveness} (shown by lines) and \textit{average unique faults} detected (shown by bars) for each inspection team size (ranging from 2–9 inspectors) for each team formation set (i.e. dissimilar, random and similar). Results are organized in the order of increasing team size. Left y-axis shows the average number of unique faults detected and secondary y-axis on the right shows the cost-effectiveness ($M_i$) of each team.

Cost effectiveness results show that inspection teams with dissimilar inspectors always found more # of faults and were more cost-effective as compared to teams of inspectors with similar or randomly selected. The results also show that teams whose members were randomly selected are more cost effective (except for team size 3) when compared to the team of inspectors with similar LS’s. Overall, dissimilar inspection teams were always cost effective across all team sizes as compared to random or similar teams. That is, more diverse inspectors’ uncovered the requirements from different perspectives and find a larger number of faults faster.

To test our hypothesis, we performed a paired t-test to see whether inspection teams with dissimilar inspectors are significantly more cost-effective as compared to inspection teams of random or similar LS’s. Results showed that dissimilar inspectors in an inspection team had a strong and significant impact as compared to inspection teams with inspectors of random (pair 1) and similar LS’s (pair 2). Among random and similar (pair 3), teams with random LS’s inspectors had strong and significant impact as compared to similar inspectors in a team which was due to the chance that some variation in the LS of the inspectors is achieved during the random selection of inspectors. Therefore, based on the results, forming inspection team with dissimilar inspectors significantly increased the cost-effectiveness.

VI. CONCLUSION AND FUTURE WORK

The results from showed that the dissimilarity in the LS of the participating inspectors had a direct and positive relationship with the cost effectiveness of inspection teams. This means, for an inspection team size, higher the dissimilarity in LS of inspectors more number of unique faults are found during the inspection of requirements. Therefore, using LS’s to guide the formation of inspection teams is beneficial. This result motivates us for further investigations. We are currently working on selection of inspection team based on the LS of author of requirement document. Another future work is to replicate the research studies during the inspection of design documents, code and the test plan reviews.

REFERENCES


Fig. 2: Comparison of LS on cost-effectiveness for team size 2 to 9