Impacting Motivation Of It Professionals Through An Objective Assessment Of Source Code Excellence

Sebastian Antony Ukken
Research Scholar in Management, Karunya Institute of Technology & Sciences
General Manager – Business Excellence
QuEST Global, Trivandrum

Dr S. Anthony Raj
Associate Professor
Karunya School of Management,
Karunya Institute of Technology and Sciences, Coimbatore

Abstract:
The work quality of any professional has a strong direct bearing on the motivation of the professional. This is a general norm in all walks of life. Similarly in the Software projects, code is vital for the quality of the product. The quality of source code has a direct impact on not only the quality of software product, but also on the motivation of IT professionals. Reviews, testing and motivation tops the list associated with ensuring a good quality software deliverable. The rework and testing effort can be significantly reduced if the developed code is of high quality and the developers have adequate skill in writing a robust and high quality code. High quality of source code is a feather in the cap for every IT professional as this is the crux of any Software project. Through this paper, the authors are attempting to propose a structured objective assessment of source code and an effective feedback mechanism for individual developers, which not only set the platform for improving individual coding skills but also the motivation of the individual developers.

Key Words: Work quality, IT Professionals, software project, individual developers

Introduction
A software quality factor is a non-functional requirement for a software program which is not called up by the customer's contract, but nevertheless is a desirable requirement which enhances the quality of the software program. Various analysis proves that, enormous effort is spent for testing activities to achieve the software quality, where the motivation level of the employee is on the lower side. This paper attempts to showcase that poor software quality can be mitigated by methods other than brute testing. The scope of this paper is to develop a measurement system for Source Code Excellence and suggested a metric based continuous improvement framework to improve the coding skill of the developers and also the motivation of the individual developers so that they produce better quality software.

Understanding Software Code
In the context of software engineering, software quality refers to two related concepts mentioned below:

- Software functional quality reflects how well it conforms to a given design, based on functional requirements or specifications.
- Software structural quality reflects on how well it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Structural quality is evaluated through the analysis of the software’s inner structure, its source code, at the unit level, the technology level and the system level, which is in effect how its architecture
adheres to sound principles of software architecture. Functional quality is enforced and measured through software testing.

Consortium for IT Software Quality (CISQ) has defined five major desirable structural characteristics needed for a piece of software to provide business value: Reliability, Efficiency, Security, Maintainability and Size.

- **Reliability:** Reliability measures the level of risk and the likelihood of potential application failures. It also measures the defects injected due to modifications made to the software. The goal for checking and monitoring Reliability is to reduce and prevent application downtime, application outages and errors that directly affect users.
- **Efficiency:** Efficiency is especially important for applications in high execution speed environments such as algorithmic or transactional processing where performance and scalability are vital.
- **Security:** A measure of the likelihood of potential security breaches due to poor coding practices and architecture.
- **Maintainability:** Maintainability includes the notion of adaptability, portability and transferability. It is also essential to keep maintenance costs under control.
- **Size:** Size of source code is a software characteristic that obviously impacts maintainability. Size of the code is measured by Lines of code. Complexity of code also needs to be tracked and re-factored as Software complexity is a well-known software metrics, this would be likely to reduce the time spent and cost estimation in the testing phase of the software development life cycle (SDLC), which can only be used after program coding is done.

Considering these inputs, the workflow for Source Code Excellence Evaluation can be defined for the organization.

**Source Code Excellence Evaluation Workflow**

For an effective Source Code Excellence evaluation system, there has to be a systematic approach which can be aligned to a PDCA cycle. First of all the organization needs to identify and define the critical characteristics of the of the software code as mentioned earlier. For each of the identified characteristics, detailed attributes are identified which are then evaluated. This forms the measurement system for the Source Code Excellence which then is evaluated against the code developed by the programmers of the organization. The weak areas are identified based on the evaluation and appropriate action plans formulated to improve the identified weak areas, thereby improving the skill of the resources. This overall process are categorized into different stages as mentioned below.

![Figure 1: Workflow for Source Code Excellence Evaluation](image-url)
The individual steps for Source Code Excellence evaluation is detailed in the subsequent sections.

**Stage 1: Establish measurement system for Source Code Excellence**

- **Identification of Major Attributes of Code:**
  A team of domain experts, technical architects, senior software engineers etc. are constituted and a brainstorming session is conducted for covering all the aspects of a Code starting from File headers, Function Headers, Indentation to Error checking, Resource and memory management etc.

  The aspects of Source Code Excellence are brainstormed in multiple meetings and based on the analysis, the critical characteristics of the software code are classified in major heads like:
  - Readability
  - Sustainability
  - Efficiency
  - Resoluteness

- **Identification of details against the major attributes:** There are many factors which contribute to the above mentioned critical characteristics of the software code. These attributes will help in analysing the code in-depth and enable rating for the code. These attributes are identified and mapped to the major heads which are referred to as the measurement system for evaluation as mentioned below.

  ![Figure 2: Measurement System for Source Code Excellence Evaluation](image)

  These attributes shall be listed in a sheet with ratings & weightage and is named the ‘Source Code Excellence Sheet’ (wherewith referred to as SCeS)

- **Establish Ratings:**
  Each of this attribute are rated at a maximum score of 3 and a weightage is assigned. The ratings are then summed and converted to a percentage value to get the SCeS score.
Stage 2: Assessing the present situation based on the measurement system

A Cross Functional team is formed within the organization/Account to evaluate the source code for various software languages and domains. Samples of the source code of minimum 500 to 1000 LOC size is collected periodically for evaluation. The evaluation is done for the software programmers normally coming under the category of Software Engineers/Trainees/Senior Software engineers. The identified SCeS review team will perform the ratings against the identified attributes in the SCeS sheet to arrive at the SCeS score.

In order to ensure that all the reviewers rate the attributes in a uniform fashion, Measurement System Analysis (MSA) is performed among the reviewers.

![Figure 3: MSA table](image)

The coefficient of concordance was found to be greater than 90% (Overall 92.14%) which indicates that there is consistency among the reviewers. Hence the measurement system for the SCeS process was validated.

![Figure 4: MSA results](image)

Stage 3: Analyze the situation and identify the weak attributes of Source Code Excellence

After the completion of SCeS evaluation, the weak attributes of the code are identified for improvements. Data analysis based on scores associated with each code attribute are analyzed and weak areas identified.
Also the score against the major attributes are analysed to identify the area of improvement. The organization can set a target for the SCeS score and the points below the score are considered for improvements. The resource can use their ratings against the attributes for their self-improvement.

Stage 4: Establish mechanism to provide Objective feedbacks:

The individual SCeS sheet is shared with the respective developer and where ratings are low, one to one sitting with the review and developer is facilitated. Inputs on how to overcome the weak areas are discussed and action plans formulated. The action plans are then implemented and the reassessment of the code is done for verifying the effectiveness of the action proposals.

Stage 5: Corrective actions:

Other than individual feedback, consolidated analysis are done at the organizational/Account level to ensure the improvement of the identified weak areas. Action proposal including trainings,
enhancements of the existing coding standards, introduction of new static analysis tools are generated to improve the overall coding skill of the employees at the organization level. The consolidated charts of the SCeS ratings is one of the best input for the senior management of the organization to identify the weak areas in coding. Also the training team can use this as an input for the training plan for the developers. The effective implementation of the corrective actions are ensured for the individual skill and SCeS score improvement in the next iteration.

Based on the SCeS score obtained within the DU, baseline can be generated to arrive at a target. Proactive action proposals shall be planned to improve upon the SCeS baseline value of the DU.

Stage 6: Monitor progress and Re-evaluation:

A SCeS calendar is generated to ensure that that Source Code Excellence feedback is provided on a regular basis. At the end of each cycle, analysis and comparison of results with the previous iteration is analyzed to understand the impact of the corrective actions taken.

At the organization level, the SCeS score across the Accounts against the target is plotted and the actions planned for the Accounts where the score is not meeting the target.

A dashboard is maintained at the organizational level, which gives the status of the ratings of the critical attributes and the SCeS score. From this the senior management can easily monitor the progress of the SCeS action plans and the effectiveness of the action plans.

<table>
<thead>
<tr>
<th>Org. Unit</th>
<th>Readability</th>
<th>Sustainability</th>
<th>Efficiency</th>
<th>Resoluteness</th>
<th>SCeS Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account1</td>
<td>94%</td>
<td>78%</td>
<td>89%</td>
<td>85%</td>
<td>87%</td>
</tr>
<tr>
<td>Account2</td>
<td>46%</td>
<td>69%</td>
<td>43%</td>
<td>78%</td>
<td>59%</td>
</tr>
<tr>
<td>Account3</td>
<td>82%</td>
<td>91%</td>
<td>76%</td>
<td>84%</td>
<td>83%</td>
</tr>
<tr>
<td>Account4</td>
<td>75%</td>
<td>56%</td>
<td>69%</td>
<td>76%</td>
<td>69%</td>
</tr>
<tr>
<td>Account5</td>
<td>45%</td>
<td>90%</td>
<td>75%</td>
<td>82%</td>
<td>73%</td>
</tr>
<tr>
<td>Account6</td>
<td>82%</td>
<td>44%</td>
<td>81%</td>
<td>72%</td>
<td>70%</td>
</tr>
</tbody>
</table>

Figure 7: Dashboard for SCeS

Motivational Index

The motivation level of an employee is quantitatively measured using the factor termed ‘motivational index’. This index basically gives the measure of the satisfaction level of the employee in his coding task. The motivational index is then derived using the rating level of the employee against the evaluation points given below

Motivational Index evaluation points

- Are you happy with the quality of the code you have produced
- Are you convinced with the structured approach to improve Source Code Excellence
- Do you think the structured approach can help to further improve your Source Code Excellence
- Are you satisfied with the improvement of your Source Code Excellence using this structured approach
- Are you happy with the current work being performed

Source Code Excellence and Motivational Index – The connection

In order to explore the connection between Source Code Excellence score and the motivational index of the individuals, two sets of team were analyzed, one with low SCeS score and the other with high SCeS score. The motivational index of these set of people were calculated based on the above mentioned evaluation points. Based on the two sets of motivational values, Moods Median test was performed and p-value was found to be 0.00. This clearly indicates that the motivational values of these two sets were significantly different (p-value – 0.00) and it was high for the employees who were having SCeS score above the threshold value.
The graphical depiction of the motivational index is shown below. It is obvious that there is a relation between the above data. The correlation between these two parameters was found to be positive. The correlation coefficient was above .95 which infers a high correlation between the parameters.

Also a regression analysis was performed to check the statistical relation between Motivational Index and SCeS score. The results indicate a positive correlation as the p value was found to be less than 0.05 (0.040 for the constant and 0.00 for SCeS score). The R-square value was found to be greater than 80% (91%) which indicates the statistical validity of the regression equation. The below figure depicts the regression analysis for Motivational Index and SCeS score.

**Figure 8: Moods Median test results**

The graphical depiction of the SCeS Score of the two sets of employees along with their motivational index is shown below.

**Figure 9: SCeS Score and Motivational Index**

It is obvious that there is a relation between the SCeS score and the motivation index as per the above data. The correlation between these two parameters was checked with the two sets of data and was found to be positive. The correlation coefficient was above .95 which infers a high correlation between the parameters.
Advantages and Disadvantages

This system is an excellent snapshot for showcasing the Source Code Excellence parameters and for providing a systematic feedback for developers thereby improving overall quality of deliverables and employee satisfaction.

On the flip side, the organization should ensure that there is no bias from the reviewer’s side since the rating can have a significant impact on the individual. Ensure that detailed guidelines are provided for the reviewers to aid in a balanced rating which would take care of the variations in scores to a large extent. Also statistical techniques like MSA can be applied to overcome such limitations.

Conclusion

SCeS evaluation is an effective means to objectively evaluate an individual’s skill in writing the code as well as ensuring Source Code Excellence. This has a direct impact on the motivation of the employees. Organizations investing in building the right attribute and a good evaluation methodology can also serve as parameter towards individual performance appraisals. Introducing rewards and recognition schemes based on SCeS evaluations can also act as a motivating factor for individuals and

Figure 7: Regression Analysis

The scatter diagram with the SCeS Score and Motivational Index is plotted to visually depict the relation between the parameters.

Figure 10: Scatter diagram

A database is maintained at organizational level, where each employee can login and access the current level of their performance. The individual improvement opportunities are logged under this and tracked to closure, which will aid the individual to enhance their performance.
projects towards ensuring good quality deliverables each time and enhancing the motivational level of the individual

References


7. **Singh Satyendra Kumar**, “Motivation as a strategy to enhance organizational productivity”, Advances in Management, July 2012

