

Software Team Skills on Software Product Quality

¹Viji Vinod, ²J. Dhanalakshmi and ³Sunil Sahadev

¹Department of Computer Applications, ²Department of Information Technology,
Dr. M.G.R. University, Tamil Nadu, India

³School of Management, University of Sheffield, United Kingdom

Abstract: This study provides, an analysis of the skills of the software development team and its effect on the quality of the final software product. An approach for assessing software development team skill and experience was derived from a workforce management tool currently in use by the national aeronautics and space administration. Using data from 220 small scale software development projects based in India and US, the team skill measures are correlated to five software product quality metrics from the ISO/IEC 9126 software engineering product quality standard. An attempt is made in this research to analyze the composition of the development team in terms of personality and in the light of team dynamics. The results indicate that development team skill is a significant factor in product development and project management of the software development. In addition, the results imply that inexperienced software developers are tasked with responsibilities ill suited to their skill level and thus have a significant adverse effect on the quality of the software product.

Key words: Project management, technical skill, software metrics, soft skills, software team

INTRODUCTION

Upon assignment of a new software development effort, a project manager begins to build the project's infrastructure in order to maximize the chances for project success. Does this project manager aggressively pursue the latest development process and demand its implementation? Does this project manager define thresholds for measures of cyclomatic complexity (McCabe, 2006) or lines of code in order to set boundaries on the complexity of the people management solution? No typically, among the first actions a manager of a software development project will make will be to secure the service highly skilled and experienced team members. To the software project manager, this is strongest approach to project success.

Despite the intuitive relationship between development team skill and software product quality, the body of software engineering research is surprisingly sparse in its coverage of the topic. While, very few software engineering models have attempted to capture development team skill as a driving cause of the success of software projects (Krishnan and Kellner, 1999; Krishnan *et al.*, 2000; Faraj and Sproull, 2007; Beaver and Schiavone, 2005; Judith, 2007), the primary focus of most software quality models (Blackburn, 1996; Borchers, 2003; Boehm, 1981; Richard *et al.*, 2007) is the analysis of people management complexity measures as indicators of

software quality. In practice, the software development industry has focused on well-defined and followed software processes as the key to software quality and project success (Diaz and Sligo, 1997; Haley, 1996; Gopal *et al.*, 2002; Michael *et al.*, 2007). While, it is difficult to argue that complexity and process are factors in the quality (Basili and Briand, 1996) of a developed software product, there is a general failure to address what every software project manager knows all too well: the skill/experience of the software development team is a driving factor in software product quality.

James Bach has authored studies insisting that the quality of the people is the primary driver for software quality and that too much industry focus has been on the process. He argues that personal performance is guided by higher level process models embedded within experience, education and insight. Certainly from an intuitive and experiential perspective, the education and abilities of an individual play an important part in the ultimate quality of their developed software. However, there is little empirical evidence to support this logical assumption.

This research attempts to quantify the relationship between the skill of the software development team and several measures of software product quality. It begins by proposing a framework for measuring the collective skill of a development team. Also, a description of several software product quality measures is provided including

an overview of how those measures were obtained. An analysis of the degree of correlation between the development team skill measures and the software product quality measures is presented and some inferences are made based on the underlying software engineering data.

Measurement of development team skill: The assessment of the capability of the software development team requires insight into the team's collective skills. The assumption is that a greater skilled team will produce higher quality in their software product and artifacts (Krishnan and Kellner, 1999). The software development team skills are examined as product development skills and business development skills to accomplish successfully. The product developments skills are focused on various technical skills on language, programming, application and hardware experience (Carmel and Bird, 1997). The business development skills are focused on different project management skills-motivation, leadership, communication coordination (Bruce, 2007; Michael *et al.*, 2007; Durani, 2004; Galin, 2004; Clampitt and Downs, 1993; Cougar, 1980); Myers, 2004.

Because of the wide range of skills (Ethiraj *et al.*, 2005; Leonardi, 2007) required at each areas of software development, the approach to measurement of development team skill is to evaluate skill and experience of the team in terms of those specific areas. For example, a development team with vast people management skill, but little technical development skill may produce a fantastic product, but may not necessarily meet the needs of the customer.

In this research, the collective skill of the development team is based on the skill ratings of individual team members. The individual skill ratings will be consolidated into a set of measures that reflects the distribution of skills for the entire team across the three life cycle phases. The National Aeronautics and Space Administration's (NASA) Competency Management System (CMS) (NASACOM, 2005) has been adopted as the approach to individual skill assessment. The CMS is a measurement framework, applied enterprise-wide, to quantify the knowledge, skills and abilities of the NASA workforce and to intelligently manage the allocation of that workforce those projects, where appropriate needs exist. For each technical discipline (e.g., software engineering, chemistry, etc.), the CMS establishes an ordinal, 4-tier scale for assessing an individual's knowledge, skill and experience. For each of the 4-tiers associated with a discipline, a set of criteria has been developed, which an employee must meet in order to be assessed at that level. In this research, the CMS approach

has been adapted to apply in terms of the software engineering life cycle. A 4-tier ordinal scale is used to rate an individual's skill/experience within each of the two skills of the software development: product development and project management Associated with each tier and for each skill is a set of criteria which, if met, would allow an individual to be assessed the appropriate numerical rating for that field.

Although, NASA's CMS provides the basis for an approach to individual skill assessment, it is necessary to broaden the individual ratings in order to capture the combined skill and experience of an entire software development team. Assessing skill for a development team involves consolidating the individual skill levels into a set of measurements that would accurately reflect the team's skill distribution in each of the life cycle phases, as first proposed by Beaver and Schiavone (2005). Table 1 shows the 12 m selected to represent a development team's skill and experience. For each phase and skill level, the proportion of the development team members that were assessed at that phase/skill level is the metric used. Normalizing the distribution of team skill by team size allows for the comparison of projects regardless of the number of team members. In addition, incorporating measures that independently address each life cycle phase allows for different team sizes and skill mixes in each phase to be accurately represented.

Measurement of software quality: Software product quality in this research is captured using as a guide for the expectations of quality within the delivered software product. The ISO/IEC 9126 is an international standard for software product quality that represents the quality of a delivered software product in terms of six major characteristics: Functionality, efficiency, reliability, usability, maintainability and portability.

In the standard, each of the six quality characteristics is further partitioned into sub-characteristics and associated indicator metrics that allow for consistent measurement and assessment of quality. This research has focused on capturing and modeling product quality in terms of the Suitability of the product. Suitability is an ISO/IEC 9126 sub-characteristic of Functionality that quantifies the adequacy of the software product in terms of both its coverage of user needs and its implementation correctness. Table 2 lists the metrics used to capture software product quality for the Suitability portion of the ISO/IEC 9126 standard. These measures attempt to capture the level of quality of the software from the perspective of functional operation of the system. The only discrepancy between the metrics presented here and the ISO/IEC 9126 is that the measure of functional adequacy has been expanded to address adequacy of both people management modules and source code units.

Table 1: Collective development team skill measures

Skill	Skill level	Measure description
Technical skill	Language experience level 1	Proportion of development team assessed at language skill
	Programming experience level 2	Proportion of development team assessed at programming skill
	Application experience level 3	Proportion of development team assessed at application skill
	Hardware experience level 4	Proportion of development team assessed at hardware skill
Project management skill	Motivation level 1	Proportion of development team assessed at motivational skill
	Leadership level 2	Proportion of development team assessed at leadership skill
	Communication level 3	Proportion of development team assessed at communicational skill
	Coordination level 4	Proportion of development team assessed at coordinating skill

Table 2: Software product quality metrics

ISO/IEC 9126 m	Metric purpose	Metric description
Design Adequacy (DA)	How adequate are the checked design modules?	The ratio of the number of design modules evaluated to function adequately to the number of design
Implementation Adequacy (IA)	How adequate are the checked source code units?	The ratio of the number of source code units evaluated to function adequately to the number of source code units
Functional Implementation Completeness (FICMP)	How complete is the functional implementation?	One minus the ratio of the number of functions detected as missing during evaluation to the number of functions described in the requirements
Functional Implementation Coverage (FICOV)	How correct is the functional implementation?	One Minus the ratio of the number of functions detected as missing or incorrectly implemented during evaluation to the number of functions described in the requirements
Functional Specification Volatility (FSV)	How volatile is the functional specification after baseline?	The ration of number of requirements changed after baseline to the number of requirements

MATERIALS AND METHODS

The approach to establishing the relationship between development team skill and measures of software quality was to record skill and quality measures from actual software development projects and analyze the results. A correlation coefficient was calculated to quantify the relationship between each of the skill variables and each of the software product quality metrics. The equation used for the correlation coefficient is shown in Eq. 1.

Equation 1 correlation coefficient calculation:

$$\text{Correl (X,Y)} = \frac{\sum (x-x)(y-y)}{\sum (x-x) \sum (y-y)}$$

The sets of data used for this analysis are from 36 software development projects. The projects were small in scale and were generally completed within a 3-4 month time frame. Project teams varied in size from 1-4 developers and included both graduate students and software engineering professionals. The projects selected had sequentially addressed each phase of the development life cycle in a classic waterfall fashion. Of the original 50 projects selected to participate in this research, 36 were actually used because of their willingness to track life cycle measures completely and correctly.

Using the measurement frameworks outlined above, it is appropriate to hypothesize the results of the analysis in terms of correlating development team skill to software product quality. Table 3 captures the expected

Table 3: Expectations for correlating skill to software quality

Skill	ISO/IEC 9126 software product quality metric				
	DA	IA	FICMP	FICOV	FSV
Technical experience	Incr	Incr	Incr	Incr	Decr
Project management skill	Incr	Incr	Incr	Incr	Decr

relationship between the Development Team skill levels and the software product quality metrics. The expectation of a direct relationship between the increasing skill levels and the increasing correlation is annotated in Table 3 as Incr. For example, as the project management skill increases, it would be expected that the correlation between that skill value and the Design Adequacy metric would increase. This indicates a positive relationship and the influence of the higher level of skill and experience on the people management of the system. The Decr. annotation captures the expectation that an inverse relationship, or decrease in correlation, exists for the increasing skill levels. Technical skill's Volatility provides a good example of this case. As technical skill increases, it is intuitively expected that the volatility of the requirements will decrease.

RESULTS AND DISCUSSION

Table 4 shows the correlation coefficients that relate the measures of development team skill to the selected measures of software quality. An examination of the results reveals some expected relationships and also some interesting deviations from the expected correlation values.

Table 4: Correlation coefficients of skill to software quality

Skill level	ISO/IEC 9126 software product quality metric				
	DA	IA	FICMP	FICOV	FSV
Language skill	-0.415	-0.755	-0.022	-0.146	-0.062
Programming skill	-0.419	-0.619	-0.042	0.038	0.190
Application skill	0.434	0.752	-0.013	-0.030	-0.094
Hardware	0.515	0.645	0.266	0.411	-0.212
Motivation	0.045	-0.314	0.246	0.166	0.052
Leadership	-0.513	-0.592	-0.302	-0.404	0.158
Communication	0.433	0.693	0.059	0.219	-0.152
Coordination	0.413	0.358	0.288	0.375	-0.235

Effects of skill on Adequacy: As expected, the increased presence of more highly skilled software developers on the development team was positively correlated with the quality measured in the final software product. This was most evident in the correlation results of the various skill levels to the measures of Design Adequacy. There is a sharp contrast between the correlation of the highly skilled (level 3 and 4) developers to adequacy and the correlation of the marginally skilled (level 1 and 2) developers to adequacy. The increasing presence of the less skilled in each phase of the development life cycle has a significant negative correlation to the adequacy of the people management and implementation. The practical inference made from these relationships is that increasing numbers of less skilled developers decrease the quality of the software product in terms of its adequacy. Conversely, the increasing presence of more skilled developers has a significant positive correlation to the adequacy of the software product, implying that increasing numbers improve the quality of the final product. These correlations provide empirical evidence for the intuitive relationship between development team skill/experience and soft-ware quality.

Effects of skill on specification volatility: The correlation coefficients associating development team skill to functional specification volatility were as expected in terms of their inverse relationship. That is as skill levels increased, there was a general decrease in the volatility of the functional specification. Despite the expected behavior, the magnitude of the correlation coefficients is surprisingly low. Intuitively, one would expect volatility to decrease dramatically as the collective skill of the requirements team increased. The low correlation coefficients indicate that technical skill has very little impact on the volatility of the software specification. Further research is required, possibly exploring the effects of software development process or product complexity, to identify a more significant factor that affects functional specification volatility.

Adverse impact of skill level 2: The most prominent unexpected result is the divergence of skill level 2 from the

expected increasing and decreasing correlation patterns. For each quality attribute and particularly with respect to people management skill, the increased presence of personnel with skill level 2 has an adverse impact on that quality attribute. For example, people management skill level 2 is expected to be more positively correlated than people management level 1 to the adequacy, completeness and coverage quality metrics. However, for each of those metrics, People management level 2 is discovered to be significantly more negatively correlated than people management level 1. In 3 of the 4 cases the difference is on the order of 0.05.

The unexpected adverse impact of skill level 2 may be the result of a flaw in the assignment of skill levels to personnel. Projects participating in this research were required to have their developers complete a questionnaire from which, each developer was assigned a skill level based on their responses. It is possible that the set of criteria that assesses personnel at skill level 2 is flawed in that developers are incorrectly rated. The existence of deviations from the expected for skill level 2 in all life cycle phases is further evidence that the set of criteria may need refinement. However, a flaw in assignment criteria should also cause unexpected results in the correlation values for the other skill levels. Since, this was not the case, we propose a different reason for the results.

Another explanation is that the deviation of skill level 2 is simply an indication that the presence of less skilled developers has a negative influence on a development effort. This premise must be explored further, however, as the same dramatic deviation was not noticed for the correlation values associated with skill level 1. In a software development effort, the minimally skilled (e.g., skill level 1) are easily recognized and often assigned more trivial aspects of the people management and implementation in order to improve their individual skill/experience. However, those with a marginal skill set or with a small amount of experience (e.g., skill level 2) are not as appropriately allocated to easier tasks. A developer with basic competence in people management skills is often assigned an equal share of the development responsibilities with his/her more highly skilled colleagues. It is proposed that those developers that have a small amount of software development ability (skill level 2) are given software development tasks that are inappropriately matched to their skill/experience level. For that reason, the increased presence of skill level 2 has an adverse negative impact on the quality of the software development product.

Effects of technical skill: The effect of technical skill on software product quality was most pronounced in it's

effect on people management Adequacy. The presence of higher technical skill levels had a strong correlation to the adequacy of the people management, while the presence of lower technical skill levels had an opposite and negative effect on adequacy. Increasing numbers of engineers that are unskilled in technical development appear to impair a project's ability to provide adequate people management modules and source code units. This result supports the notion that the quality of the technical skill has a dramatic effect on its downstream products in the people management area of development.

In analyzing, the effect of technical skill on the functional implementation coverage and completeness, it is interesting to note the lack of significance of skill levels except for skill level 4. It appears that only the involvement of the most highly skilled requirements engineers has a significant effect on the correctness and completeness of the final software product. This may be an indication of leadership on the technical development team. Leadership in developing technical skill was a criterion for assessing individuals at technical skill level 4. It is reasonable to infer that experienced leadership in technical development effort is the driving factor in the effect of technical skill on the functional implementation completeness and coverage.

Effects of people management skill: The people management skill measures correlated most significantly with the adequacy of the people management and implementation. This aligns with the intuitive assumption that those with a higher level of people management skill and experience will produce a more adequate people management and therefore, a more adequate implementation. Similarly, those with less people management skill and experience will produce a less adequate people management and implementation.

In addition, people management skill level 4 was moderately positively correlated with the functional implementation completeness and coverage quality variables. Similar to the inferences made in the analysis of the effects of technical skill, it is postulated that people management leadership is the skill element that distinguishes people management level 4 as a significant factor for these quality attributes.

Threats to validity: All of the software engineering data used in this study is from projects that are relatively small, object-oriented software efforts. While, the results provide evidence that development team skill is correlated to software product quality, it is premature to infer that the results are applicable outside of the scope of the underlying software engineering data. Future research is

required to validate these results for both large-scale projects and projects that do not use the object oriented paradigm.

CONCLUSION

This study provided empirical evidence that development team skill has a significant effect on the quality of a software product. A measurement framework for assessing software development skill and experience was used to quantify the capabilities of the software development team. These measures were correlated to measures of software product quality taken from the ISO/IEC 9126 standard. As expected, higher proportions of skilled and experienced staff in the technical, people management and implementation life cycle phases increased the adequacy of the people management modules and source code units and had a similar albeit more muted effect on the functional implementation completeness and correctness and on the volatility of the software specification.

In general, higher proportions of skilled engineers had the most dramatic effect in terms of adequacy of the people management and implementation. Conversely, higher proportions of less skilled engineers negatively affected the end product quality of the software. The adverse impact of skill level 2 on the software quality metrics implies that marginally skilled developers are assigned research that is inappropriate for their skill levels. For the relationship between development team skill and functional implementation coverage and completeness, the increased presence of experienced leadership (skill level 4) in each of the life cycle phases had the most significant impact.

The proposed framework for measuring development team skill and experience provided an adequate representation of the distribution of individual skills on a given development team. It is recommended that software project managers assess their software developers using the proposed framework in order to better understand the strengths and weaknesses of their staff. Such an understanding could prevent the assignment of tasks that are not well suited to less skilled and experienced engineers and thus, prevent the negative impact on product quality.

Future study for this research is planned. Data collection for a more diverse set of projects and in terms of size and complexity is expected. This research also will be expanded to provide a basis for various globally located software engineering projects. The focus will be the determination of optimum skill/experience combinations, taking into consideration personnel costs and resource availability, in order to maximize the quality on all software projects.

REFERENCES

- Basili, V., R. and L.C. Briand, 1996. A validation of object oriented design metrics as quality indicators. *IEEE. Trans. Software Eng.*, 22 (10): 751-760.
- Beaver, J.M. and G.A. Schiavone, 2005. Predicting Software Suitability uses a bayesian belief network. In *Proceedings of the 4th Int. Conf. Machine Learning and Applications*, pp: 82-84.
- Blackburn, J.D., 1996. Improving speed and productivity of software development, working paper, INSEAD, Fontainebleau, France.
- Boehm, B.W., 1981. *Software Engineering Economics*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Borchers, G., 2003. *Software Engineering*, Proceedings of 25th International Conference on Software Engineering. ieeexplore.ieee.org.
- Bruce, J.A., 2007. Leadership: Current theories, research and future directions. *Ann. Rev.*, 60: 421-449.
- Carmel, E. and B.J. Bird, 1997. Small is beautiful: A study of packaged software development teams. *J. High Technol. Manag. Res.*, 12 (2): 110-123.
- Clampitt, G. and W. Downs, 1993. Employee perceptions of the relationship between communication and productivity. *J. Businss Commun.*, 30 (1): 5-28.
- Cougar, J R., 1980. *Zawacki Managing and Motivating Computer Personal*, Wiley.
- Durani, 2004. A survey to examine the effect of team communication on job satisfaction. *ACM SIGSOFT, Software Eng. Notes*, 29 (2): 1-10.
- Ethiraj, S., P. Kale and M.S. Krishnan, 2005. Where do capabilities come from and how do they matter? A study in the software services industry. *Strategic Manage. J.*, 26 (1): 25-45.
- Faraj, S. and L. Sproull, 2000. Coordinating expertise in software development teams. *Manage. Sci.*, 46 (12): 1554-1568.
- Galín, D., 2004. Core and periphery in free/libre and open source software team communications. Addison Wesley.
- Gopal, A., T. Mukhopadhyay and M. Krishnan, 2002. The role of software processes and communication in offshore software development. *Commun. ACM*, 4: 193-200.
- Haley, T.J., 1996. Software process improvement at Raytheon. *IEEE. Software*, 13 (6): 33-41.
- Judy, G. and L. Heather, 2002. Flexible work arrangements and organizational communication an Australian retail experience. Working paper, Monash University, Department of Management.
- Judith, S.T., 2007. Information technology work force skills: Does size matter? *Inform. Syst. J.*, 24 (4): 345-359.
- Krishnan, M., C. Kriebel and Sunder, 2000. Inform an empirical analysis of productivity and quality in software products. *Manage. Sci.*, 46 (6): 745-759.
- Krishnan, M. and M.I. Kellner, 1999. Measuring process consistency: Implications for reducing software defects. *IEEE. Trans. Software Eng.*, 25 (6): 800-815.
- Leonardi, P.M., 2007. Activating the informational capabilities of information technology for organizational change. *Organ. Sci.*, 18 (8): 813-831.
- McCabe, 2006. A complexity measure. *IEEE. Trans. Software Eng.*, 2 (4): 308-316.
- Michael, E., D. Ray and E. Janet, 2007. An analysis of software quality management. *Software Quality J.*, 15 (4): 347-365.
- Myers, 2004. The relationship between perceived instructor credibility and college student in class and out of class communication. *Commun. Reports*, 17 (2).
- NASSCOM, 2005. *NASSCOM-McKinsey report*.
- Richard, W., B. Selby and Boehm, 2007. *Software engineering*. Published by Wiley-IEEE.