Ensuring Success and Quality through the Use of Standards in Team Projects: Students' Perceptions

Elsje Scott[1], Robert Brown[2], Jeffrey Pearce[3] and Peter Weimann[4]

[1] Senior Lecturer at the University of Cape Town, RSA, elsje.scott@uct.ac.za
[2] Business and Software Solutions Group, RSA, robobrown@gmail.com
[3] Business Analyst at White Wall Web, RSA, pearce.jp@gmail.com
[4] Professor at Beuth University of Applied Science Berlin; Visiting Academic at the University of Cape Town, Germany/RSA, weimann@beuth-hochschule.de

Abstract. This paper reports on a study analyzing the factors that contribute to the success and quality of software development projects in an educational environment. Software development standards were reviewed to identify key project success factors, as well as measures of success and quality. Interviews with students then investigated the degree to which students implemented software development standards in their projects, and the perceived impact of these standards on project success and quality. Students generally viewed standards as the "proven" ways of doing something, and felt that the use of standards supported project success. The factors that were perceived to contribute most to the success and quality of their team projects were team composition, skills within the development team, and communication within the team.

Keywords: Project Success, Software Development, Quality Standards

1. Introduction

Information Systems (IS) professionals are often involved in tailoring application technologies and developing systems or parts of systems to suit the requirements of an organization [1]. In preparation for this, students should gain experience in the application of information and technology-enabled business processes in such a way that the organizations will benefit and obtain a competitive advantage [2]. However, despite endeavors to effectively educate prospective IS professionals, one out of three Information Technology (IT) projects still fails more than 10 years after the famous 1995 Chaos Report of the Standish Group [3]. Projects that fail either miss the targets or they do not deliver the required business functionality [4]. The main aim of this study was therefore to determine the factors which contributed to the success and quality of software development projects in an educational environment.

The paper reports on an empirical research study conducted in 2006 at the University of Cape Town (UCT), South Africa, to investigate students'

perceptions of the factors that contributed to the success and quality of the IS undergraduate system development projects. The paper briefly outlines success and quality as defined in the literature and the relevant factors contributing to success and quality. It comments on standards as determinants and measures of success and quality. The students' perceptions are then compared to those revealed in the literature, various themes are identified, and their relevance to the objectives of the study is discussed.

2. Success and Critical Success Factors

Many different and sometimes very diverse factors can contribute to the success or failure of software projects, making it difficult to define and measure their success [5]. The factors "on time", "within budget" and "to specification" have often been used to measure the success of projects in terms of their outcome [6, 7]. Even though Turner [8] argues that these three factors primarily represent the view of the contractor of a project, they remain the highest ranked factors in a number of different surveys [9, 10]. Other important factors that have been used to assess project success are:

- The fit between between project and organisation [10]
- The consequences of the project for the performance of the business [10]
- The quality levels met [9]
- The satisfaction of users and other stakeholders [11]

Baker, Murphy and Fisher [12] point out that a project can still be perceived as successful even if it has not met the timescale and budget requirements. This implies that the success of a project depends partly on the viewpoints of the different stakeholders [9]; and so project managers in particular cannot ignore what developers consider being important in terms of project success [13]. Furthermore, the success of a project can also be described in terms of how the project affected the team and its individual members with respect to the level of stress, overtime, conflict, satisfaction and the level of motivation [14].

The list of factors *influencing* the success of a project is as diverse as the set of factors *measuring* the success of a project. In a survey by White & Fortune [10] the most frequently mentioned criteria are: clear goals; support from senior management; adequate funds and resources; and realistic schedules. Other project success factors frequently mentioned are: end user commitment; clear communication channels; effective leadership/ conflict resolution; effective monitoring and feedback; flexible approach to change; taking into account past experiences; recognizing complexity; taking account of external influences; effective team building/motivation; and effective management of risk. These critical project success factors are in line with Belassi and Tukel, [15]; Magal, Carr and Watson [16] and Pinto and Slevin [11].

3. Quality and Factors Contributing to Quality

Software quality is an important determinant in the success of software projects [9]. It is therefore not surprising that it is recognized as one of the 10 knowledge areas of the Software Engineering discipline and pervasive in the Guide to the Software Engineering Body of Knowledge (SWEBOK) [17]. Software quality has traditionally being defined as "fitness for use" [18]. According to the International Organization for Standardization (ISO) 9001 standard, software quality is "the degree to which a set of inherent characteristics fulfills requirements" [19].

The field of software quality is broken down into three subtopics, namely: Software Quality Fundamentals, Software Quality Management Processes and Practical Considerations [17]. Software Quality Fundamentals refer to the culture and ethics of software engineering, the value, costs and characteristics of quality as well as quality improvement processes. Software Quality Management (SQM) encompasses the different perspectives of software processes, products and resources whereas Practical Considerations refer to requirements, defect characterization, management techniques and measurements.

Software quality factors such as understandability, completeness, conciseness, portability, consistency, maintainability, testability, usability, reliability, structure, efficiency and security constitute non-functional requirements for a software program. In many cases, related attributes of these factors can be used as metrics, which allows subsequent measurement of how well the project goals have been achieved [17]. In addition to the technical qualities of software, the end user's perspective on the usability of a software product must also be considered.

4. Standards as a Requirement for Success and Quality

Moore [20] states that the sound engineering approaches provided by The Software Engineering Standards of the Institute of Electrical and Electronics Engineers (IEEE) Computer Society and their SWEBOK can be applied to increase the probability of success. According to Moore [20] a standard is a measure of comparison, a characterization to establish allowable tolerances or constraints for categories of items and a level of required excellence. All standards have limitations, and it is often not possible to comply only with one single standard when developing software [21]; because of this, the adoption of standards should not necessarily be viewed as mandatory.

The ISO 9000 family of standards represents an international consensus on good quality management practices that assist organizations to consistently deliver quality products and services across all industry sectors. The ISO/IEC 9126 standard of reference helps to stabilize the software process by providing a framework for the evaluation of software quality [22]. In part 1 of this standard a generic quality model is defined in terms of six quality characteristic, each with its own set of sub-characteristics. Part 2 describes external metrics that are applied to an executable software product in the later stages of developments or during the

testing process. Internal metrics (part 3) are applied to a non executable software product in design or the early stages of coding.

The use of standards is also encouraged by the SWEBOK, which consists of ten knowledge areas that establish the appropriate set(s) of criteria and norms for software engineering practice upon which industrial decisions, professional certification and education can be based [17]. Other standards bodies such as the Software Engineering Institute (SEI), ISO, IEEE (SWEBOK), Object Management Group (OMG) and PMI (PMBOK), assist in providing a foundation for project success. These are incorporated in the CxOne quality standard document [23], which contains a number of sub-areas including project planning, process planning, testing and verification and validation.

At UCT a systems development group project is one of the major deliverables of the one year capstone course of the IS undergraduate curriculum. The course includes topics like requirements planning, software design, software construction and testing. Templates, checklist, patterns and guides are custom designed and support the creation of the relevant artifacts [24]. A comprehensive assessment strategy implements various instruments to accomplish formal summative assessment, formal continuous assessment and an informal formative assessment [25]. The course content adheres to international curriculum standards as specified in the IS Model Curriculum [2] and the Computing Curricula 2005 [1].

5. Research Methodology

The main aim of the study was to determine those factors contributing to the success and quality of the UCT 2005 IS systems development projects. A case study approach was used to unearth the factors and explore the question: "What are the students' perceptions on success and quality in software development projects in an educational environment?" Several authors agree that a case study approach provides an effective way to examine specific phenomena or to explore a question or an issue of concern in its own context e.g. [26]. According to Flyvbjerg [27] it fosters an in-depth understanding of the issue of concern, and Yin [28] concurs that it helps to link causes and effects. It thus seemed viable to investigate the students' perceived ideas of success and quality, before exploring which factors they thought contributed to the success and quality of their systems development project.

The data was collected in semi-structured interviews. The sample comprised a selection of 13 teams out of a total of 25 teams who completed the course in 2005. The selected teams were representative of teams in the upper, middle and lower mark brackets. In most cases the team leaders were approached for the interviews. In some cases other team members were interviewed as well, accounting for the 18 interviews conducted. The interview comprised three sections. The first was used to gain an understanding of the student's awareness of the role standards play in software development arena. It also provided some insight into the importance the students placed on the usage of standards in their projects. The second section

of the interview attempted to identify the student's perception of the impact standards have on the success and quality of the systems development project. The third section was designed to establish the interviewee's awareness of the key areas of software development as revealed in the literature. The interviews were treated as confidential and conducted individually to guarantee anonymity of the interviewees at all stages. Notes were taken during the interview to assist in identifying themes. Additional questions were asked to aid in the understanding of the respondent's viewpoints. The interviews were recorded with the consent of the respondent and transcribed in a textual format to enable the researchers to identify themes guided by Ryan and Bernard [29]. Mind maps as a visual aid assisted to group the various themes into categories as described in the following section.

6. Findings

We grouped the themes that emerged in the data collection process into six categories namely: factors contributing to the success of projects; factors contributing to the quality of projects; tools and techniques; standards; focus areas; and the adherence to UCT course guidelines.

6.1 Success Factors

The most important factors contributing to the success of software development projects as mentioned in the interviews were:

Team roles: Healthy team dynamics and open communication in software development teams was a major theme raised by most respondents. It was also important that team members exhibited commitment and contributed to a good "mix" of skills and as one of the team members verbally confirmed: "Need to choose the right people, if you don't have the right people, there is no way you can pass the project well."

Good testing: Although only three teams claimed to have used formal testing methods, most respondents felt that proper testing is of major importance. This is consistent with the literature which claims that if software is tested properly and functions correctly it contributes to success and quality [18, 30]. Many teams only performed informal testing and did not execute formal testing methods due to poor time management.

Project sponsors: The respondents stated that the support given and relevant information obtained from the sponsor enabled them to build up a better system and therefore contributed to the success of the project. Project success was also dependant on meeting the sponsor's requirements.

Good time management: "We wasted a lot of time, and were really crammed towards the end. I feel that if we had planned our time better we would have done better". Most respondents agreed that should they have managed their times better, more tasks would have been completed successfully. It is clear that time

management was another major theme that emerged from the interviews. Although compulsory, only ten out of the 13 teams interviewed constructed and used their Gantt charts effectively to assist them in task allocation.

Proper analysis: The majority of respondents admitted to using tools and techniques such as Use Case diagrams to assist in requirements planning, but only two teams felt that conducting thorough analysis was essential to the success of their projects.

Coding standards: Several respondents stated that better adherence to coding standards could have improved their final result.

From the above factors it is clear that the respondents felt that people as stakeholders of a project played an important role in the success of their projects. This is in line with the study of Procaccino et al. [13]. Although the respondents focused more on personal achievements such as their marks and the learning experiences as indicators of a successful project, they agreed that one of main factors for success was to effectively meet the sponsor's requirements.

6.2 Quality Factors

During the interviews the researchers tried to group the responses on quality in software projects under the six different criteria for quality as depicted in the ISO 9126 standards [22]. Five of the six areas, listed below, were either directly or indirectly mentioned in the interviews.

Usability: Phrases like "easy to use", "user friendly", "user interface finish" and "value to end user" were mentioned as determinants of quality.

Functionality: Ten out of the 13 teams interviewed, mentioned that "meeting the sponsor requirements" is an important determinant of quality. Other responses reflecting the functionality aspect of quality were "fit for purpose", "innovation" and "value to end users".

Efficiency: The themes that emerged from the responses under this area were "speed of use" and "speed and efficiency".

Maintainability and reliability: A few respondents mentioned that they view "reliability" and "robustness" as valuable to enhance the quality of a system. One respondent focused strongly on the use of coding standard to make the system "maintainable".

The general perception of students towards quality was primarily that of usability, functionality and efficiency. They considered maintainability and reliability as secondary factors and no mention was made of portability. This could perhaps be contributed to the fact that students gain little experience in the implementation of the software products they develop. Other than this, the data collected regarding students' perceptions of quality correlated with the ISO 9126 standard on quality software [22].

6.3 Tools and Techniques

Several tools and techniques identified in the data collection process impacted directly or indirectly on the projects' success and quality. Respondents mentioned that adhering to the UCT coding standards provided them with useful patterns to create higher quality software products. Unified Modeling Language (UML) artifacts were used to improve requirements planning. Storyboarding and extended UML artifacts were used in the design phase. The project management section was supported by the use of Gantt charts; timesheets, risk analysis and estimation techniques. Testing emerged as a major theme, where respondents reported that test cases were used to perform more structured testing. "Testing to break" was applied more informally and unstructured, but contributed to the quality and success of the projects. A few respondents reported to have used "XP style testing", "acceptance testing" and "unit testing" which contributed to the success and quality of their projects.

6.4 Standards

The majority of respondents defined a standard as "something you need to achieve", whilst others thought that it was the "same way of doing something" or "something that is consistent and enables a predictable result". Seven of the respondents were of the opinion that standards were important as they provided guidelines, directions or a baseline for comparison. Their general perceptions of standards as an adherence to sound software engineering approaches, were close to the description of the standards in the literature [20]. Although students felt that using standards was a positive thing, they also felt that it might have limited their abilities within their teams. This is in agreement with Bennatan's [21] viewpoint that standards limit the freedom of developers. Moore [20] however, disagrees as he argues that the use of standards is completely voluntary.

6.5 Focus Areas

In the third section of the interview process the researchers attempted to gauge the respondent's awareness of the key areas of software development process as identified in the literature. Although the respondents did not list these areas directly, their responses could be categorized under the respective areas: project management, requirement planning, software design, coding and testing. Amongst others, aspects like time management, the overseeing of the project, controlling and organizing events as well as the managing of people and risks received special attention. The responses varied significantly, but comments relating to testing however showed some consistency. Barnett and Raja [18] state that many authors in the literature view testing as a means of achieving software quality. Comments like: "testing is a very, very crucial part of the systems development, and it should occur on a constant basis" and "include quality assurance and checking

consistently" confirm that students agree with the view of testing being important to maintain quality and improve their final product.

6.6 UCT Guidelines

Throughout the project course clear guidelines were given to students in three major areas:

Project guidelines: The two top teams felt that their success was largely due to the fact that they followed these guidelines meticulously. The six project teams who obtained middle of the range marks also benefited from following these guidelines. Four of the five teams with very low marks confirmed that they did not follow the guidelines well.

Coding guidelines: Almost all the respondents used the UCT coding guidelines as a base for their coding structure and confirmed that it was a critical component of their project success. One team did not benefit from it. They however admitted that they had "no standard conventions" and "little coding experience". In an extreme case the top team used these guidelines as a base to develop their own "self built code generator", reducing the time to produce standardized code.

Assessment guidelines: The majority of the teams used the appropriate mark sheets and rubrics available when they produced deliverables throughout the year and when they planned their final presentations (code and project). These teams generally felt that it helped them to meet expectations and this contributed to their success. Three teams, who did not use the mark sheets in their preparation, did not do well.

In general it seemed that where project teams adhered to the UCT guidelines provided, they reaped the benefits by enhancing the quality of their system and achieving success.

7. Conclusion

The main focus of the IS Curriculum design is to provide students with sufficient experience in the application of information and technology-enabled business processes to be able to contribute to the competitive advantage of organizations [2]. Although the students' general perception of the success of projects focused more personal achievements such as the marks obtained and the learning experience of the project, they emphasized the importance of good communication and the correct mix of skills in a project. Their perception of standards as being "very important as they are best practices" and "providing a base", "gives us guidelines", or "give you direction for the future" adheres to Moore's [20] description of the role of standards [19, 22]. Testing was not always implemented in a formal manner in the student projects, but the general perception that it is a "very crucial part of the systems development" confirms the acute awareness of the importance of testing.

Through the "lived" experience of the project, students gained a better understanding of a number of different factors that contribute to the success and quality of systems development projects. Team composition, skills and communication within the teams were seen as major success factors, followed by the need to meet sponsor requirements. Student teams who adhered to the UCT guidelines consistently performed better than those who did not. In view of this finding, it is important to ensure that guidelines are aligned to industry standards and that an awareness of the benefits in using the standards is cultivated throughout the delivery of the course.

Acknowledgements: The authors would like to express their gratitude to Brian O'Donovan and Jane Nash for editing the paper. Their useful comments were invaluable in enhancing this paper.

References

1. Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R. et al.: Computing Curricula 2005: The Overview Report. In: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education. ACM, Houston, Texas (2006)
2. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., Longenecker, H.E.: IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Communications of the Association for Information Systems, 11 (1), pp. 1--63 (2003)
3. The Standish Group.: Chaos Report. Standish Group (1995)
4. Nelson, R.: IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices. MIS Quarterly Executive, 6 (2), pp. 67--78 (2007)
5. Chan, Y.E.: IT Value: The Great Divide Between Qualitative and Quantitative and Individual and Organisational Measures. Journal of Management IS, 16 (4), pp. 225--62 (2000)
6. Project Management Institute (PMI).: PMBOK – A Guide to the Project Management Body of Knowledge. (3rd ed.). Sylva, NC.: The Project Management Institute (2004)
7. Schwalbe, K.: An Introduction to Project Management (2nd ed.). Course Technology Cengage Learning, Boston (2008)
8. Turner, J.R.: Handbook of Project-based Management. McGraw-Hill, London (1993)
9. Wateridge, J.: How can IS/IT Projects be Measured for Success? International Journal of Project Management, 16 (1), pp. 59--63 (1998)
10. White, D., Fortune, J.: Current Practise in Project Management – an Empirical Study. International Journal of Project Management, 20 (1), pp. 1--11 (2002)
11. Pinto, J.K., Slevin, D.P.: Project Success: Definition and Measurement Techniques. Project Management Journal, 19 (1), pp. 67--72 (1988)
12. Baker, B.N., Murphy, D.C., Fisher, D.: Factors Affecting Project Success. In Cleland, and King (Eds.), Project Management Handbook. Van Nostrand Reinhold, New York (1983)
13. Procaccino, J.D., Verner, J.M., Lorenzet, S.J.: Defining and Contributing to Software Development Success: Determining the Process-Related components affecting Software Developers' Perception of Project Success. CACM, 49 (8), pp. 79--83 (2006)

14. Freeman, M., and Beale, P.: Measuring Project Success. Project Management Journal, 23 (1), pp. 8--17 (1992)
15. Belassi, W., Tukel, O.: A New Framework for Determining Critical Success/Failure Factors in Projects. International Journal of Project Management, 14 (3), pp. 141--51 (1996)
16. Magal, S.R., Carr, H.H., Watson, H.J.: Critical Success Factors for Information Centre Managers. Management Information Systems Quarterly, 2, pp. 413--26 (1988)
17. Bourque, P., Dupuis, R. (Eds.).: Guide to the Software Engineering Body of Knowledge: 2004 Version. www.swebok.org. IEEE Computer Society (2005)
18. Barnett, W.D., Raja, M.K.: Application of QFD to the Software Development Process. International Journal of Quality and Reliability Management, 12 (6), pp. 24--42 (1995)
19. ISO: 9001: 2000 Quality Management Systems – Requirements (2000)
20. Moore, J.W.: The Road Map to Software Engineering: A Standards Based Guide. John Wiley and Sons Inc, New Jersey (2006)
21. Bennatan, E.M.: On Time, Within Budget (3 ed.). John Wiley and Sons, Inc, New Jersey (2000)
22. ISO/IEC: 9126: Information Technology- Software Product Quality (2001)
23. Construx.: Cxone. (2008), http://www.construx.com/Page.aspx
24. Scott, E.C.: Systems Development Group Project: A Real World Experience. Information Systems Education Journal, 4 (23), pp. 1--10 (2006)
25. Scott, E., van der Merwe, N.: Using Multiple Approaches to Assess Student Group Projects. Electronic Journal of Information Systems Evaluation, 6 (2), pp. 177--86 (2003)
26. Lee, A.: A Scientific Methodology for MIS Case Studies, MIS Quarterly, pp. 33--50 (1989)
27. Flyvbjerg, B.: Five Misunderstandings about Case-Study Research. Qualitative Inquiry, 12 (5), pp. 219--245 (2006)
28. Yin, R.K.: Case Study Research: Design and Methods. Sage Publications, Thousand Oaks, CA (2003)
29. Ryan, G.W., Bernard, H.R.: Techniques to Identify Themes, Field Methods, 15 (1), pp. 85--109 (2003)
30. Carroll, J.: The Application of Total Quality Management to Software Development. Information Technology and People, 8 (4), pp. 35--47 (1995)