Software Engineering Challenges: A Case Study

Proceedings of the 4th National Conference; INDIACom-2010
Computing For Nation Development, February 25 – 26, 2010
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

"Software Engineering Challenges: A Case Study"

Prof P.L.Ramteke
Asstt.Prof in IT
HVPM's COET ., Amravati
Maharashtra (INDIA)
Mob. +919421818808
pl_ramteke@rediffmail.com

Dr.Mohammad Atique
Associate Professor
University Department, Amravati
Maharashtra (INDIA)
Mob. +919823724560
atique_shaikh@rediffmail.com

EXTENDED ABSTRACT

Software developments are the challenges today by changing nature of requirements. There are various software development processes, based on software engineering's disciplinary approach. This paper focuses on the issues that occur during the development of software specially quality software and how efficiently handled these issues using effective management principles and techniques. There are various challenges to produce a quality product, within time frame. The challenges may be: changing the customer requirements, estimation of efforts & resources, risk impact, miscommunication among staffs, unrealistic deadlines, a lack of action to correct problem, software myths, quality controls, Software configuration management issues, technical difficulties, human resources, right estimation, analysis-design-coding-implementation and testing issues. It is very difficult task to handle these issues but not impossible. Therefore in this philosophy oriented research paper some management principles and techniques are suggested through the case study. To handle these issues efficiently and effectively used principle and sound project development plan. Also restrict software product by process framework & metrics in context with software objectives and goals. To achieve quality product in accordance with requirements, budget and time used with some effective techniques. The success of quality software development is very much associated with the ability to recruit responsible & skilled persons and it is equally important to provide an environment in which these persons can work smoothly, with full efficiencies and without any unnecessary burden or restrictions from management peoples of business organization. There has been an increasing need to find effective ways to impart Software Engineering practice. This case study discusses the problems of different software engineering practices. A practical focus in a scalable approach is to developing world-class software. The proposed case studies approach that addresses these issues. This research aims to understand the best practices for software development in this software engineering approach. To create effective software process the focus is to emphasize on creating a context to learn Software Engineering through case studies that imbibe the best practices from real world experiences. Improvement of software process to achieve high quality in a software development organization is the key factor to success.

KEYWORDS

Software Engineering practices, quality software, process activity, software myths, risk management, software configuration management, quality assurance.

1.0 INTRODUCTION

The Software development is one of most important technologies on the world stage. Software has evolved a specialized problem solving and information analysis tool to the industry. Software becomes very necessary technology for business, science, engineering, transportation, medical, telecommunication, bank & so on. There are demand of changing nature of software requirements like system software, application software, engineering /scientific software, embedded software, AI software, web application. With this changing environment it is difficult to develop high quality software within budget and in time.

There are so many software development process models used for this purpose. But due to complexity of problems itself or by involvement of various issues like *changing requirement, under estimate efforts, increased project tracking overhead*, it is difficult to develop quality product. So it is needed to use agile or modern software development approaches that are appropriate for project team and product too One of most traditional model approach is system development life cycle (SDLC), and waterfall model, incremental model, rapid application development model, spiral model are also be used for this purpose. They are not 100% accurate or perfect, but they provide a useful roadmap for software engineering tasks. To develop software product is no matter at all but to produce quality software within budget and in time boundary is really a matter. This is a complex job, and will not be feasible without the management of various software constraints and possible issues in software development process. Therefore it is really a challenge to handles these issues effectively.

1.1 LITERATURE REVIEW

Software Engineering is a disciplinary approach that deeply linked to practical aspects of developing software within budget, schedule and quality. For this purpose the Software Engineer should have technical as well as managerial expertise including problem-solving skills, while being aware of various issues. Many development models and pedagogies are in use & many new ones are being developed for Software Engineering product in professional environment. But the current Software Engineering Education is found mostly based on the class room learning model. So at what extent this modeling approach is suitable to produce capable software Engineers. Can they possess the required skills through this learning approach to fulfill the industrial requirements? It is difficult to find the realistic way to handle various issues in software development. This case study support in this regards as well as it focuses various issues in s/w engineering education through class room teaching-learning approach. Hence the different traditional and agile approaches are needed to identify, for Software Engineering practices by using Case Study approach. The approach is backed by its successful implementation in management, medicine and various other disciplines. The focus is on using case studies as the teaching instruments to effectively address the learning disabilities [3]. This approach is being implemented at International Institute of information Technology (IIIT), Hyderabad, India. The initial observations and results are quite encouraging.

Traditional Approach toward Software Engineering Education: The traditional and the most prevalent approach to Software Engineering Education are to impart theories and concepts in a class room environment. Research done by learning scientists has proved over and over that this kind of learning is no learning at all. [3, 4]. The traditional model of class room learning enforces a reactive rather than proactive approach. i.e., the participants are expected to react to a solution presented instead of proactively thinking about the problem on hand. The participants settle for short term objectives like making it through the course instead of leveraging on long term goals of learning. Many times, academic projects accompany the regular course, but the observations are that the participant's focus is more on Computer Science issues especially on a working program and not on the development process and the relevant issues [5]. Rarely do the participants experience in full the intricacies related with a project in the professional environment. In reality though, it is often the much neglected application of the Software Engineering concepts that largely determine the success or failure of a project. Owing to this mismatch between the education and its application, the present teaching methods prove to be inadequate for the purpose. Many new models, often criticizing the conventional methods have emerged. Some teaching methods are based on principles of goal driven learning [6], learning by doing, story centric learning] and few others [7, 8] recommend self-direction, cognitive learning, team based or collaborative learning. Each technique has its own pros and cons. But none of these techniques has been found a best fit for Software Engineering Education in both the academic and the corporate environment and most of them don't satisfy the three paradigms of learning, namely 'learning by doing', learning by mistakes' and 'learning by stories' [4].

2.0 CASE STUDY APPROACH

A case study is defined as a "systematic inquiry into an event or a set of related events which aims to describe and explain the phenomenon of interest" [9, pp.302]. Looking from Software Engineering perspective, a ‘case’ or ‘case study’ is an account of software engineering activities, events or problem containing background and complexities actually encountered by a software engineer. Case studies as teaching instruments facilitate “thinking forward from first principles”. They emphasize on the application aspects of the tools, techniques and most necessarily, the concepts learnt so far. This approach is suitable for those subjects where the subject itself is not just a bundle of *concepts, techniques or models; instead the subject has a broader perspective* and uses concepts from many domains to solve a problem. Case studies are more relevant to the kind of challenges professionals, both young and experienced, face and bring in the elements of practical way of solving problems and resolving complex situations. Software Engineering is just this kind of subject, and Software Engineering Education needs just this kind of learning model that help the participants to understand and manage the true nature of the present software industry, Complex, evolving and challenging.

This case study approach may be beneficial over the conventional software development methodology. To built up the strong and relevant questionnaires, investigate and evaluate the feedback towards Software Engineering practices that helps to create an effective learning and development environment. It must be useful to conduct research to build sound software process and to identify the best approach for software engineering education.

2.1 PROPOSED PHILOSOPHY

Philosophy is the study of general and fundamental problems concerning matters such as existence, knowledge, truth, beauty, law, justice, validity. Or it is a discipline involves creating concepts that are always new. Because the concept must be created, it refers back to the philosopher as the one who has it potentially, or who has its power and competence.

the given solution/action), *Firm/ Product Oriented Case* can also be used, which promote a product or a firm and can exhibit industry and technological trends.

Concepts must be invented, fabricated or created and would be nothing without their creator's signature.Philosophy helps us to develop our ability to:

Think and express ourselves clearly

- see both (or more) sides of a case
- get to the heart of the matter
- see the implications of a line of thought
- detect bad reasoning, and be confident that our own is good
- detect ambiguity, vagueness, inconsistency, and other weaknesses in the expressions of ideas
- distinguish different types of question, claim or argument, and respond to them appropriately
- distinguish what is relevant to a given issue from what is not
- see ways in which an argument or explanation could be improved

2.2 CONSTITUENTS

The Case study, Faculties related to software engineering courses, professional developer team and students participants are the basic constraints. Case studies are the main instruments in this approach. A rightly carved case study can increase the effectiveness of the learning environment in many folds. The types of Software Engineering cases that can be used include requirement analysis case (users, technical, managerial or s/software, hardware etc) d*esign solution cases* (usually requirement based designs-the challenge is to come up with a suitable system.),code for functional based cases. *Select Solution Case* (given a problem and its various solutions, evaluate all and suggest one with focus on suitability and effectiveness for the problem), *Read-between-lines case* (given a problem/ decision dilemma and the solution or action that followed, critically analyze the given solution and judge the applicability of

Case Studies have many goals. They can’t focus on just one issue, as that would not reflect the real world situation This variety of cases and different scenarios makes the case study approach very flexible. The case study can be a hypothetical one, a real life project experience, a hybrid of both, or an adapted version of the real project.

It is important to have a good understanding of the approach by the facilitator and the participants. All participants should be willing to put sincere efforts for analyzing and presenting the analysis.

2.3 EVALUATION

# Activity Weight Factors Detail

(1) Student Participation: 50% weightage will be given to active learning and analysis skills, based upon questions asked and issues raised during various discussions

(2) Software Engineer's and SE course teachers' feedback evaluation: 50% weightage will be given to this feedback based on questionnaires and their comments.

Technically, the participants must have taken a first course on domain understanding & in programming and should have some experience of software projects, though academic. The participants can belong to any level in the industry, profession or academic.

(3) Other relevant methods.

2.4 OBSERVATIONS

The author being the senior faculty and the mentors of SE practices and courses will be observed the questionnaires and other relevant methodology based results of use-cases models, with concluding remark.

Table 1. Table captions should be placed above the table

Constituent s	Feedback Topic	Feedback Quantity	Response from Constituents
1. S/w Course Students	-S/w process is backbone of quality -*Unrealistic deadlines, changing requirements , resource difficulties, risk identify missing.* -Estimation of efforts and resources are not big challenge for quality s/w	36 No.s *36 No.s* 36 No.s	60 to 80 % agreed And 15% very agreed *More than 90% agreed.* 100% not agreed.
2. S/w development professionals	Success of Quality s/w is not associated with responsible & skilled persons. There is impact of s/w myths on s/w plan and quality. S/w metrics is not needed to manage s/w project.	10 No.s 10 No.s 10 No.s	95% faculties are not agreed. More than 90% faculties are agreed. 80 % faculties are disagreed
3. Related Teachers	Industries experience is beneficial to improve quality of s/w	15 Nos.	80 %& above are agreed.

2.5 RESULTS & INFERENCES

Based on observations and data interpretation there are definitely the challenges in s/w development. Unrealistic deadlines, changing requirements, resource difficulties, risk identify missing are real challenges. Software process must be sound and strong. Estimations of Resources is needed.

2.6 CONCLUSION

It is concluded that based on experimental result of questionnaires there are various big challenges even today too. It will compel the various software development issues. and identify relevant management technique in several conventional and non-conventional learning models can be effectively addressed by the Case Study approach questionnaire based. Further research and evaluation of the learning process may be indicated if any to extend the application of this approach to create effective learning environments.

2.7 FUTURE SCOPE

Find the methodology or phenomenon to study s/w engineering courses in engineering education, based on case study approach.

REFERENCES

[1]Jorgensen, M.; Gruschke, T.M.,” The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty assessments**”,** Software Engineering, IEEE Transactions on Volume 35, Issue 3, May-June 2009 Page(s):368 – 383

[2]Napier, N.P.; Mathiassen, L, Johnson, R.D,”Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study Software Engineering”, IEEE Transactions on Volume 35, Issue 5, Sept.-Oct. 2009 Page(s):593 – 606

[3] Rogger S .Pressman, “Software Engineering: A practitioner’s approach”, , 6th edition, Mc-Graw International edition, 2005.

[4] Kendall & Kendall, “Systems Analysis and Design”, , 5th edition, Pearson Education (LPE), first Indian print, 2003.

[5] Senge PM, “The fifth discipline: the art and practice of the learning organization” New
York: Currency Doubleday, 1990.

[6] Kemi Jona, “Rethinking the Design of Online Courses”, Keynote paper, ASCILITE 2000
Education and Practice (SE:EP '96), January 24 - 27, 1996

[7] David Carrington et.al, “Teaching PSP SM in the Large Class”, 14th Conference on
Software Engineering Education and Training, 2001, Charlotte, North Carolina

[8] “Guide to the Software Engineering Body of Knowledge-SWEBOK”, 2004. A project of
the IEEE Computer Society, Professional Practices Committee

[9] Sims-Knight, J. E., & Upchurch, R. L. (1992). “Teaching object-oriented design to
nonprogrammers: A progress report”. Proceedings of OOPSLA-92 Educators'
Symposium. Vancouver, British Columbia, Canada.

[10] V.E. Veraart, S.L. Wright, “Experience with a Process-driven Approach to Software
Engineering Education”, 1996 International Conference on Software Engineering:

[11] Bromley, D. B., “Academic contributions to psychological counseling: I. A philosophy of Science for the study of individual cases” Counseling Psychology Quarterly, 3(3), 299-
307, 1990

WEB RESOURSES

[1]http://www.banglajol.info/index.php/JBAS/article/viewArticle/2432
[2].http://www.philosophy.unimelb.edu.au/about/why-philosophy.html
[3] http://en.wikipedia.org/wiki/Philosophy
[4]http://www2.warwick.ac.uk/fac/soc/philosophy/research/phillit/research/wip/research/philosophy/
[5]http://whitepapers.techrepublic.com.com/abstract.aspx?docid=889215
[6]http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1167820
[7] http://portal.acm.org/citation.cfm?id=1083258.1083264
[8] http://books.google.co.in/books
[9]http://www2.computer.org/portal/web/csdl/doi/10.1109/CSEET.2009.23
[10]http://www.iiit.ac.in/techreports/2008_56.pdf