Software change management – Technological dimension

S. M. Ghosh[1], H. R. Sharma[1], V. Mohabay[2]
[1]*Chhatrapati Shivaji Institute of Technology, Durg (CG) – INDIA*
[2]*Department of Electronics and IT, Kalyan Mahavidalaya Bhilai (CG) - INDIA*
samghosh06@rediff.com
hrsharma44@gmail.com
mohabay@gmail.com

Abstract

Effective change management requires continual assessment of the potential change impact and the organization's willingness and ability to adopt the next wave of transformation. Naturally, no change management program goes completely according to the plan: people tend to react in unexpected ways; areas of anticipated resistance may fall away; and the external environment may shift. However, supported by solid decision-making processes and using the up-to-date information from the field, the change leaders usually make the adjustments necessary to maintain momentum and drive overall business results.

Keywords: *Software Engineering, Change Management, Process Model.*

1. Introduction

The requirement or even the thought of bringing change or redesign processes at an organization, no matter how minor it is often creates 'frowning faces' and 'raised eyebrows'. However, equally true is the fact that changes keep occurring every now and then in various departments and divisions of firms. This is to say that amendments can neither be escaped nor denied. The only passage that remains in such cases is to willingly accept the bitter reality of fluctuations and embrace those with all affection.

To talk of the nature of change, it can either be of 'planned nature' or fall in the 'sudden category'. In some cases, organizations follow this trend of changes just because they wish to be on safer side. In other words, to keep the workforce attentive and prepared for the amendments, they would keep introducing little changes so that when it gets 'compulsive', the resistance from employees is minimal. Also, these can relate to any of the levels from 'operational', 'tactical' and 'strategic'.

2. The context of change management – Technologic Perspective

Change management support has to involve all of the following facets: the groups involved with the change and change activity have to be coordinated and managed, the change must be supported by organisational change management process models, and technical support for change activities has to be provided [1]. Software change management

tools and systems do not aim at reducing the rate of change, but they aim to reduce the overall costs of change management by facilitating the speed with which changes can be processed [2].

The techniques used for supporting change management can be categorised into generic software engineering techniques and change management specific techniques [3] . Generic software engineering techniques have been designed or are used in general for software development, but are vital in soft ware change management, as well. These techniques include, for example, soft ware configuration management environments [4] and basic development tools, such as compilers, debuggers and editors.

However, special techniques for supporting software change management exist. These include, for example:

- Impact analysis techniques for analyzing and modeling the impacts of the modification in the system. These techniques are useful, for example, in project schedule estimation, consistency checking and risk analysis [5][6][7].

- Change request tracking to support management of change requests. The change requests are triggers for change. Their purpose is to (1) express a need for a change, and (2) to document change activities by providing change histories for individual changes and the software entities. The benefits of change request tracking systems usually are in documenting and communicating the changes.

- Reverse engineering techniques for deriving higher level descriptions from lower level presentations to help in understanding the software and improving its quality by the terms of understandability and consistency by updating outdated documentation [8][9][10]. Examples of such techniques are tools generating graphical design descriptions from source code, for example the ReverseNICE tool by Intecs Sistemi for generating HOOD descriptions from Ada source code.

- Regression testing for assuring that the modification has not created undesirable side effects in the system. Regression testing tools usually repeat old test cases and compare the new test results with the old ones in order to find out deviations.

One of the greatest challenges of change management is to keep the system parts and several abstraction levels of the system consistent with each others. This can be supported by an integrated software development and maintenance environment, where the tools used for creating and managing the software systems are able to communicate with each other and can share common parts .

Figure 1 presents an example of an integrated software change management environment, where the tools used for software modifications and development are able to communicate with each other and share knowledge. The environment was developed in the AMES project [11]. The tool environment presented here was constructed to support change activities defined by the V model for software change. The technical dimension of change management is therefore defined by the selected process model. The change process (adapted from [12]) is

presented in the upper part of the diagram and the individual tools used in change management are linked with the process via a process support tool. In this example, change request tracking is performed by the process support tool. The interoperability service provides a link between the tools. All software related items are stored in one archive, which is used by all the tools through software configuration management. Links between semantically related software parts, different abstraction levels, composition structures, etc. are stored in a traceability database.

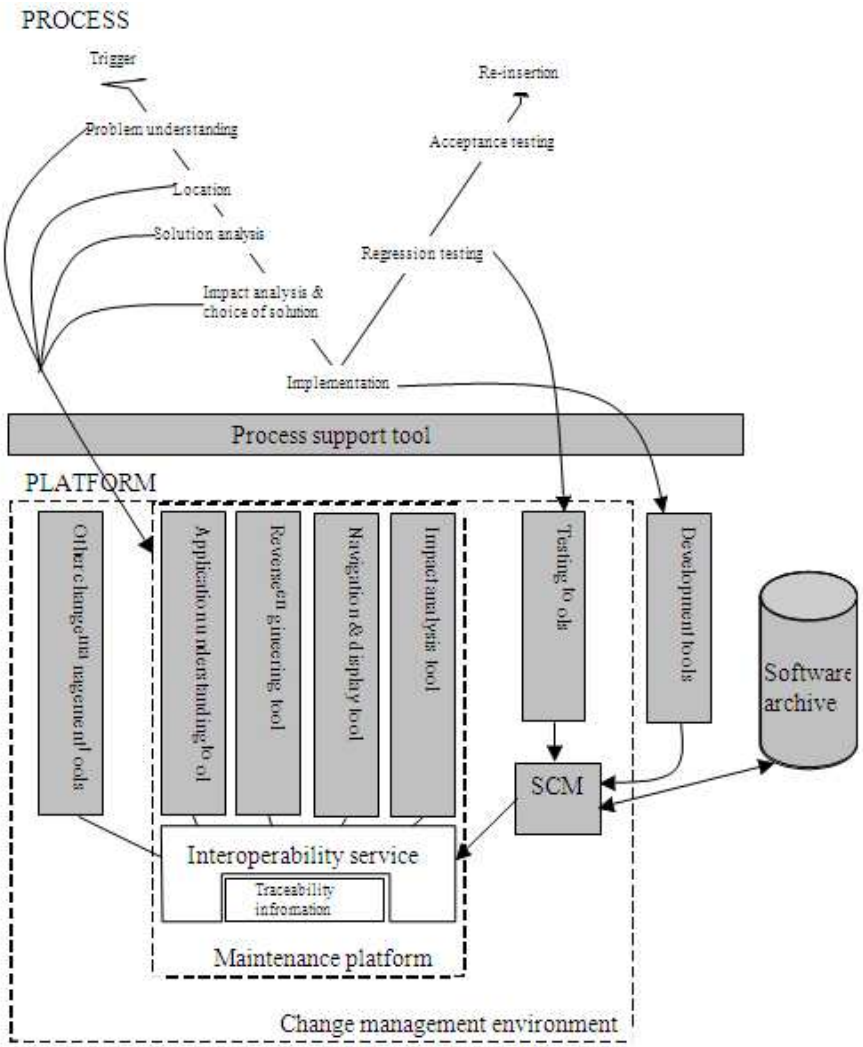

Figure 1. An example of an integrated change management environment (Mäkäräinen 1996).

An integrated change management environment was developed in the AMES project using the requirements elicited from the case studies. The project delivered a set of methods and tools for managing software changes.

3. The proposed integrated Change Management Environment [14]

Inspired by spiral model of B.Boehm [13] and the studies carried out with analyzing the various merits and demerits of change management models, the origin of Spiral model came in existence. The proposed model was presented in Figure 2.

The spiral model divides the change management process into four cycles, in which the same main tasks are performed by each cycle, but the viewpoint is different in each cycle. The execution of the process starts from the innermost cycle. The first cycle is performed by the founder, or "owner", of the problem. The problem at this point can be either:

- A request for something new, e.g. a new feature or service
- A problem in an existing product, e.g. an error situation in a product

As a result of the first cycle, the owner of the problem decides if the problem needs to be taken care of, and how it should be taken care of. The second round of the spiral is optional. It is executed if the problem needs to be examined from a non-technical viewpoint. If the technical solution is known and clear after the first cycle, the second round can be skipped and the execution can be continued in the third round.

The third round of the process examines the modification from the system point of view, and makes an implementation plan for the last round. At this point, the affected parts of the system and the requirements for the modification task are forwarded to the fourth and final round of the process. The final round generates, implements and verifies the technical solution planned during the third cycle. It also closes the modification action by delivering the result and documenting the actions and observations made.

The process is generic from the organisational point of view, i.e. it has to be instantiated for the organisation using it. It is not life cycle dependent, but models the changes in both the development and maintenance phases. Different types of changes are not considered by the model.

Comparing the model of integrated change management environment given by Mäkäräinen and the proposed spiral based model is that the spiral model does not treat software change as a post-delivery activity. Ince's change management model describes only the outer cycle of the spiral model and the last quarter of the system engineering cycle, i.e. the actual technical implementation of the change. The V-model was used in defining the outermost cycle of the spiral model. Olsen's change model treats all software development activities as changes. The spiral model only addresses the actions performed for changing existing pieces of work, not creating new artifacts but modifying it based on the changes done.

It is very important that Change Management is planned before development starts and not just put on as an afterthought in the maintenance phase. This allows the Change Management method the possibility to influence the development discipline in a positive way. More importantly, it also permits us to gather information bout the project right from the start. One of the major benefits of Change Management is that it serves as a central collection of all information. The proposed Spiral model for software change management is basically based

on the study of various existence models and their merits. The primary objective of this work is to provide a model that is leveled and can be used to make sure that a project has the right amount and sort of change management, neither more , nor less because in both the cases it is dangerous.

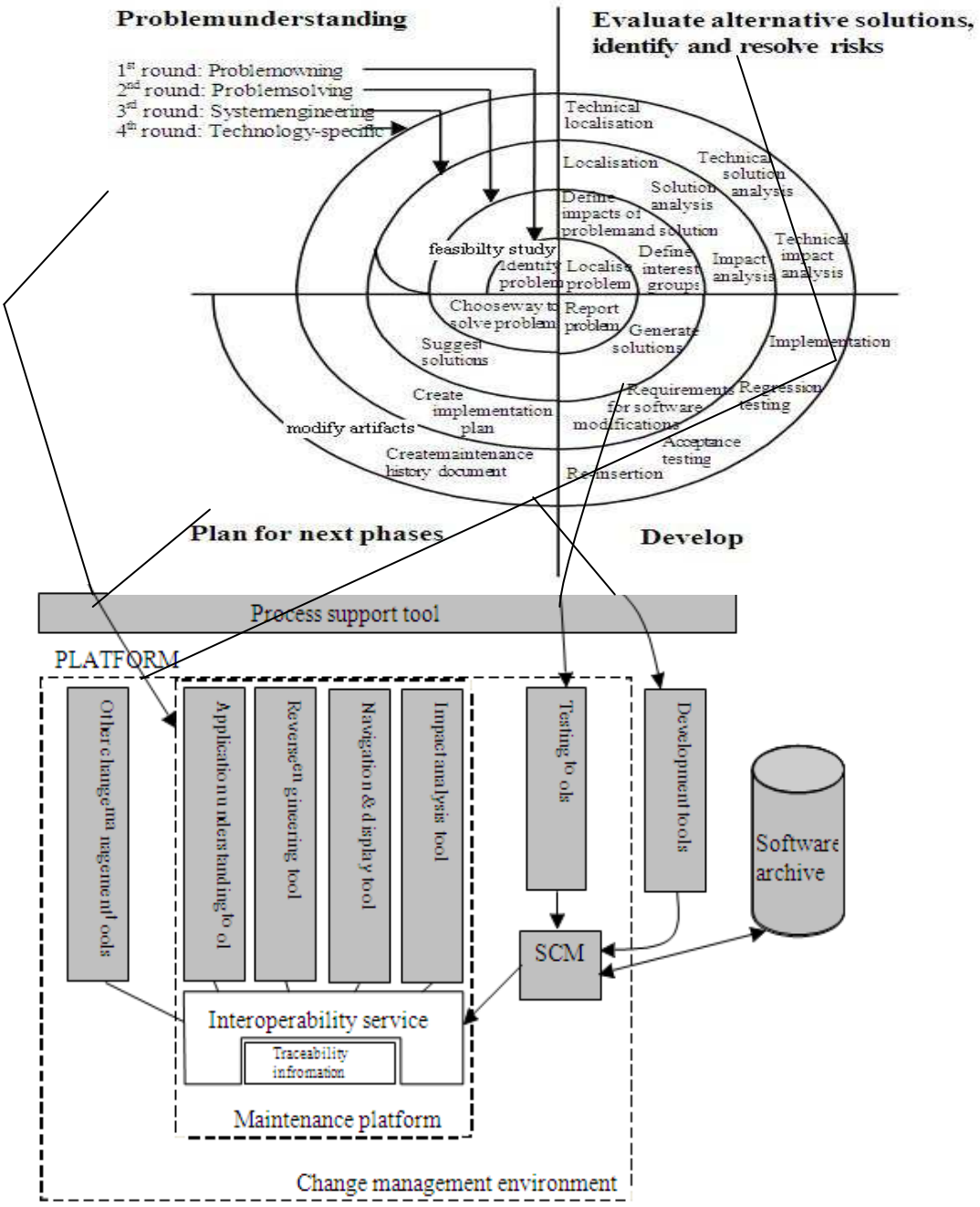

Figure 2. Proposed integrated change management environment using Spiral model

4. Summary

The process and technology dimensions for improving software change management process need more focus on techniques used for supporting change management. Because of the volume of data involved, you often need tool support to manage software change. As with any type of tool, we should get the right tool for our job. Process should drive the tool not the tool to solve the problems alone. Unfortunately, often we don't know what process we want until we have tried using the wrong tool. Identifying the best current process and the problems with it are the first steps to defining a better process. We present a spiral model approach integrated with Change Management environment for better software change process. It benefits over role model, a conventionally used modeling approach, in terms of expressive power, flexibility, and inconsistency control

References

[1] Michelis, G., Dubois, E., Jarke, M. Matthes, F., Mylopoulos, J., Papazoglou, M., Pohl, K., Schmidt, J., Woo, C. & Yu, E. 1997. Cooperative information systems: A manifesto. In: Papazoglou, M. & Schlageter, G. Cooperative infor mation system: Trends and directions.

[2] Jones C. 1996. Strategies for managing requirements creep. Computer. (June 1996 Pp 92 – 94.

[3] Mäkäräinen M. 1996. Application management requirements for embedded

[4] Ince, D. "Introduction to software quality assurance and its implementation", McGraw-Hill, ISBN 0-07-707924-8, 1994

[5] Arnold R. 1993. Software impact analysis. Tutorial Notes, IEEE Conference on Software Maintenance. Montreal, Canada.

[6] Arnold R. & Shawn, A. 1996. Software change impact analysis. IEEE Computer Society Press. 392 p. ISBN 0-8186-7384-2.

[7] Queille, J-P , Voidrot, J-F., Wilde, N. & Munro, M. 1994. The impact analysis. Task in software maintenance: A model and a case study. Proceedings of International Conference on Software Maintenance. IEEE. Pp. 234 – 242.

[8] Chikofsky,E. & Cross, J. 1990. Reverse engineering and design recovery: A Taxonomy. IEEE Software. (Vol. 7, No 1). Pp. 13 – 17.

[9] Sneed,H. 1995. Planning the reengineering of legacy systems. IEEE Software. (Vol 12, No 1.)

[10] McClure,C. 1989. The three Rs of software automation. Prentice Hall. 304 p.ISBN 0-13-915240-7

[11] AMES Consortium. 1993. AMES technical annex, Version 1.0. ESPRIT Project 8156

[12] Harjani, D.-R. & Queille, J.-P. "A process model for the maintenance of large space systems software", Proceedings of Conference on Software Maintenance. Los Alamitos: IEEE Computer press. Pp. 127 – 136. ISBN 0-8186-2980-0, 1992.

[13] B.Boehm, "A spiral model of software development and enhancement", Computer. (Vol. 21.) pp. 61 – 72. ISSN 0018-9162, 1988.

[14] S.M.Ghosh , H.R.Sharma, V.Mohabay, "Analysis and Modeling of Change Management Process Model" proposed for the SERSC –International Journal of Software Engineering and application 2011.

Authors

S.M.Ghosh
Director
Prism School of Business and Entrepreneurship
S.M.Ghosh received his Master Degree in Computer Application and authored 3 books i.e. Software Engineering , Introduction to Computing and Management Information System. Published research papers in Software Change management related topics in National and International Journals. Designed Enterprise Resource Planning Software for Job based fabrication and Casting Steel Industries. Teaching QMS , Software Engineering and MIS to the PG Students from last 12 years.

Dr. H.R. Sharma
Director, Department of Computer Science
Chhatrapati Shivaji Institute of Technology
H.R. Sharma received his M.Tech degree from Delhi University in Computer Science and has completed his Ph.D from IIT, Delhi. He has also published more than 380 research papers at National and International level. He has supervised 9 research projects sponsored by UGC & AICTE. He has more than 45 years of teaching experience at Graduate and Post Graduate level. He is presently working as the Director of Chhatrapati Shivaji Institute of Technology, Durg.

Dr. V.K.Mohabey
Head of Department, Depatment of Electronics and Information Technology
Kalyan Post Graduate College
V.K.Mohabey has published more than 356 research papers in which 68 are published at National level and 12 at International level. He has supervised 7 research projects sponsored by UGC, DST, MAPCOST, etc. He is presently working as a Professor and Head of Department of Electronics and Information Technology in Kalyan PG College, Bhilai. He has over 35 years of teaching experience at Graduate and Post Graduate level.