

# IJERT

ISSN : 2278-0181

## International Journal of Engineering Research & Technology

**Call for  
Papers**

**Publish & Find Papers @**



**www.ijert.org**



**BROWSE**

OPEN



ACCESS

## An Overview on Web-Based in Software Engineering

Pradeep Kumar<sup>1</sup>, Debabrata Samanta<sup>2</sup>, Mousumi Paul<sup>3</sup>

<sup>1</sup>Student, MCA Dept, Acharya Institute of Technology, Bangalore

<sup>2</sup>Asst. Prof., MCA Dept, Acharya Institute of Technology, Bangalore

<sup>3</sup>PhD Scholar, National Institute of Technology, Durgapur, West Bengal

Email: research.k04@gmail.com

### Abstract

*It is well known that a software process is the origin for a successful software project. This paper talks about the development of a web-based software process management and monitoring tool proposed to assist in producing superior software projects.*

*The process management tool, developed in the programming language Java, is accessible through standard World Wide Web browsers. This provides entrance to project management information in a transparent platform-independent manner to geographically distributed programming teams and project managers. Java was chosen as it can be used with standard World Wide Web browsers; it is architecturally neutral and provides good support for graphics.*

### 1. Introduction

Software projects have the potential to suffer from numerous dilemmas, with: missed deadlines, inaccurate budgets, unmet specifications, product defects, unforeseen project risks, changing requirements, poor resource planning, and poor management. These risks have the potential to turn any software product into a disaster, and to ruin a software organisation. It is possible to minimise these risks by improving project planning and management.

Projects that have deficiencies in their plans have the potential to expose people to unnecessary risks. Developing software by following a defined software engineering process is one widely adopted system of risk management. A software engineering process covers activities including the development life-cycle model, collection of project metrics, planning based on past performance, the applicable standards that should be followed when building the software, as well as managing the process.

Managing the software process is vital for the success of a project. Suitable management aims to produce better quality software products while simultaneously improving the organisation's software capability as specified by the Capability

Maturity Model (CMM). Most organisations should be able to achieve levels 2 or 3 of the CMM. This denotes that any process used in a software organisation needs to be utilised in a manner that enables the organisation to grow and prosper. Failing to manage the process in a responsible manner causes it to degenerate into an inefficient bureaucratic exercise as staff become complacent therefore reducing the effectiveness of the software process. If management see poor results, they are likely to ignore the process which, if managed correctly, could have provided significant benefits. Management of the software process has several benefits. Project plans become more realistic and attainable because they are based on past history rather than guesses and the organisation is in a better position to plan future projects. More realistic project plans mean that resource scheduling can be achieved realistically and deadlines are more likely to be met.

The problem of receiving timely information can be addressed through the use of the World Wide Web (WWW). This provides a mechanism that can be quickly accessed by people all over the world resulting in project teams in Australia and in the United States, for example, having access to the same information without the time delays caused by shipping documents between the two locations using conventional mechanisms. However, the question of security is even more important with the use of the WWW than using the standard electronic mail system..

### 2. Context of Web-Based in Software Engineering

Many different software engineering tools exist, all of which are invented to assist with different aspects of software production. Some of the existing tools include:

- configuration management tools which provide a means of controlling access to documents and software artefacts helping to make sure that consistent systems exist for different platforms etc.,
- Coding tools which assist in writing and compiling source code,

- Modelling and simulation tools which allow a proposed system to be evaluated,
- Structure editors which are designed to allow documents to be produced, and
- Prototyping tools which assist with building a prototype system.

Amongst this arsenal of technology there is a lack of tools explicitly designed to deal with the software process associated with managing and defining a process model

### 2.1 PROCESS WEAVER

PROCESS WEAVER is a work-flow process management system developed by Cap Gemini Innovation, France, and is designed to support the work of a group of people. PROCESS WEAVER brings unique benefits to an organisation including:

- Allowing business procedures to be modelled, helping to ensure that any work performed conforms to the recommended business process,
- Allowing pending actions and task completion to be tracked,
- Providing a mechanism to measure a processes effectiveness through the information that is gathered on project completion,
- Allowing multiple processes to exist and permitting the user to choose which process is right for the job,
- Assisting with process improvement and maintenance, through the use of graphical editors.

These benefits are achieved by providing a suite of tools designed to manage the tasks performed by individuals as well as a sequence of tasks performed by a team.

### 2.2 CONVERSATIONBUILDER

ConversationBuilder is an open environment which facilitates the flow of information among a group of users.

The environment can be adapted to different development models and new tools can be incorporated into the environment. Dynamic changes, which occur during execution of a development model, can occur and the environment has some understanding of the actions performed by users and can use this information to provide context sensitive help.

Each activity is called a conversation or protocol.

The protocol makes available a set of rules which govern the behaviour of conversations instantiated from the protocol.

### 3. Proposed Tool

A web-based process management tool is developed in the programming language Java. Java is a programming language that has been designed with the WWW and security in mind. Java allows

secure, platform independent programming that can be accessed through standard WWW browsers. This means that an organisation can control access to the web page, which ultimately restricts access to TOOL.

### 3.1 System architecture

The Tool follows platform independent client-server architecture is accessed via the WWW and communicates with the server through sockets. The server is used to control access to a central data repository.

Multiple users in geographically dispersed locations can access TOOL simultaneously. Each user has their own instance of TOOL which is connected to the server via a socket over the WWW. Security of information is achieved through user authentication via a username and password mechanism and, by encrypting messages sent via sockets.

The server runs as a Java application on any platform for which a Java interpreter is available. One restriction that is imposed is that the server must be run on the system where the database is stored. This is necessary because the server accesses the database. It can receive messages from each instance of TOOL and can send messages back in response. Failure of the server causes the link to be broken and the database to become inaccessible to TOOL. Error messages are supplied to the user informing them of this.

### 4. Conclusion

Following a software process when building a software product has been confirmed to improve the quality of the software but it is important that this process is managed in a responsible manner. Any process identified using TOOL should not be blindly followed, but rather it should be used in a responsible manner that includes making sensible decisions about the process and where it should be improved.

### 5. References

- [1] Gibbs, N. E. The SEI education program: The challenge of teaching future software engineers, Communications of the ACM, 1989, 5.
- [2] Skordalakis, E. Software engineering teaching at NTUA, in Proceedings of the First International Conference on Software Engineering in Higher Education, ed. G. King, C. Brebbia, M. Ross, G. Staples, Computational Mechanics Publications, 1994.
- [3] Koutoumanos, A., Papaspyrou, Retalis, S., Maurer, H. &Skordalakis, E. Towards a Novel Networked Learning Environment, in Proceedings of World Conference of Web Society (WebNet'96), San Francisco, USA, November 1996.
- [4] Hiltz, R. The virtual classroom learning without limits via computer networks, Ablex Publishing Corporation, USA, 1995.

- [5] R.S.Pressman, Software Engineering – A Practitioner’s Approach, 3rd Ed., McGraw-Hill International Editions, 1992, pp. 22 – 33.
- [6] S.R.Schach, Classical and Object-Oriented Software Engineering, 3rd Ed., Irwin, 1996, pp. 52 – 70.
- [7] W.W.Royce, Managing the Development of Large Software Systems: Concepts and Techniques, 1970 WESCON Technical Papers, Western Electronic Show and Convention, Los Angeles, August 1970, pp. A/1-1 – A/1-9.
- [8] B.W.Boehm, A Spiral Model of Software Development and Enhancement, IEEE Computer, Vol. 21, No. 5, May 1988, pp. 61 – 72.
- [9] H.Wohlwend, and S.Rosenbaum, Schlumberger’s Software Improvement Program, IEEE Transactions on Software Engineering, November 1994, Vol. 20, No. 11, pp. 833 – 839.
- [10] I.Bhandari, M.Halliday, E.Tarver, D.Brown, J.Chaar and R.Chillarege, A Case Study of Software Process Improvement During Development, IEEE Transactions on Software Engineering, December 1993, Vol. 19, No. 12, pp. 1157 – 1170.
- [11] P.Armenise, S.Bandinelli, C.Ghezzi and A.Morzenti, Software Process Representation Languages: Survey and Assessment, Proceedings of the 4th Conference on Software Engineering and Knowledge Engineering, 1992, pp. 455 – 462.
- [12] S.M.Sutton Jr, P.L.Tarr, and L.J.Osterweil, An Analysis of Process Languages, Computer Science Technical Report 95-78, University of Massachusetts, August 1995.
- [13] C.Fernstrom, PROCESS WEAVER: Adding Process Support to UNIX, Proceedings of the Second International Conference on the Software Process, 1993.
- [14] S.M.Kaplan, W.J.Yolone, A.M.Carroll, D.P.Bogia and C.Bignoli, Supporting Collaborative Software Development with ConversationBuilder, Symposium on Software Development Environments, December 1992.
- [15] J.Gosling and H.McGilton, The Java Language Environment: A White Paper, Sun Microsystems, [http://java.sun.com/doc/language\\_environment/](http://java.sun.com/doc/language_environment/), May 1996.