Modern Education
and Computer Science
PRESS

A Simple Software Engineering Environment for Coming Decades

Md. Yasin Ali Khan [a] and Md. Abu Sayed [b]

[a] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong-4349, Bangladesh
[b] Department of Computer Science and Engineering, American International University-Bangladesh, Banani, Dhaka-1213, Bangladesh

Abstract

Effective Software development is a very complex task. When this task span over large geographical regions with different cultures, it becomes very critical maintaining every task and making proper co-ordination among them. For coming decades, considering different web technologies and other emerging technologies, developers will need a robust and secure development environment for proper combination with collaboration tools having existing social and technical elements. Again this software engineering environment should ensure participation of individuals such a way that their potential get full force workable mode. This environment should also face time critical project management issues having high risk at different stages. Fast and cost effective development also requires use of off the shelf resources. In this paper, a software engineering environment is proposed for coming days considering all these issues hoping that, it will serve as a primary standard of development environment while meeting all demands of developers.

Index Terms: Software engineering environment, software failure, agile software engineering, distributed software processes, communication pattern.

1. Introduction

Software development is an art and is a layered process in which everything from software management to development is deeply studied, analyzed, implemented, evaluated and tested before the use of the product in the real world. Developing and maintaining a software project needs substantial effort, preplanning and management. Software failure [17] rate on different projects indicates that we need robust disciplined software engineering environment. Impact of better software engineering environment tremendously helps to develop and deliver better quality product in a much better way than traditional process or environment. Developing software in an agile way can ensure successful integration with legacy infrastructure. Application of

* Corresponding author:
E-mail address: shihabyasin@gmail.com[a], abusayed93.cse@gmail.com[b]

Information Technology will help people solving their different problems in real life situation if an easy interface can be provided. Designing large effective software is not a task that can be done sitting in a desk and working one night. Software development currently in many forms where distributive development [1][2] is suitable for fast, module and coherent development. For large enough development we need strategy of co-ordination within the structure of development environment for social and technical collaboration [3].

Distributed software development with better socio-technical co-ordination [3] for coming decades will be extremely sophisticated as new technologies such as web 2.0, Social Networking [4] etc are emerging. Here a simple multi-view software engineering environment [5] will be recommended for morphogenic software[1] development using collaboration tools [1][6] that support "distributed cognitive framework [7][8]". This paper will intuitively focus on "Collaborative SEE architecture" [5][9][6][4], "Architecture of Participation" [2][10], GDSS [11], "Collaboration Maturity Model(CMM)" [7], "Social Network Analysis" [12][13][4] "Emerging People's Role in Software development" [14][12] and "pair programming" [15].

2. Related Works

Different people and organization from different countries and cultures contribute in this fascinating field; Ricardo de Almeida Falbo says software engineering systems are systems to develop and maintain software development and maintenance, they provide means to integrate developers with the software process and the supporting technology. These models provide a platform to provide all the domain and case tools for the software environment [23]. Emily Navarro et.al says that most of the time software engineering knowledge is made limited to only books and classroom, it is not practiced in the real world like other engineering disciplines. Most of the work is made limited only to books and written work of requirements gathering and test cases written only [24]. We can also mention tremendous contribution from Software Engineering Institute [26].

3. Problems On Software Development

Some problems to be mentioned on developing software especially in large distributed fashion are like: task scheduling, project management, dynamic nature of software requirements, resource management, risk management, testing strategy, distributed nature of development [27][28], planning, effective tracking, quality assurance [29], realistic schedules, coding conventions, co-ordination problem [32] etc [30][31][32].

4. Our Proposed See Architecture

Software architecture is the core element in software development. This paper is based on the basics of software development. Here, a proposed architecture be given that is designed for simple software development suitable for meeting need of coming decades.

It has few modules which are interlinked to each other. First layer is User's layer, second is Client side coordination layer and third one is Server side.

All these layers are interlinked and interdependent on each other where these web repositories are connected to the server and Client layer through web cloud.

A customer side is based on many modules like design, human resource management system where this system is linked to a server which keeps all record of the events and actions done on the client layer.

These software repositories provide sessions to the clients through web.

Abbreviation for describing figure 1.

WSD = WORKFLOW SUPPORT DATABASE

The WSD is an extending database technology that will support process-centralized environments. It can be used in other environment and database platforms and can be modified independently.

PMT = PROJECT MANAGEMENT TOOL

The PMT will help to manage and incorporate the project conveniently. It will help to organize key files, keep track of what the team is working on, never ever miss any information, lay out milestones and deadlines, and get regular feedback to make sure everything all right to deliver a better product.

PDT = PROCESS DEFINITION TOOL

With each passing day of today's business world, the need of improvement in workflow is felt frequently, which indicates high requirements to the PDT. The PDT is used to create workflow process definitions. PDT will create it by defining the transitions between activities in a workflow. PDT also allows to view a version of the workflow process definition.
It will check the process definition and report any error for transitions, for activities or for workflow process.

WH = WORKFLOW HANDLER

WH are fairly simple classes that interact with the service layer.

WE = WORKFLOW ENGINE

WE is an application that organizes business processes. Being a key element in workflow technology, it makes use of a database server efficiently. It automate the flow of activities, reducing the time took to complete and errors caused by human interaction. Workflows help business users and processes to reflect changes in the business and to work together to modify systems. WE also makes processes more efficient by ensuring that every process step is monitored in time and optimized for maximum output. Using WE, workers can make smarter decisions.

EE = EVENT ENGINE

The EE is a collection of objects that manage access to a list of extended events and enable processing event data.

WMS = WORKFLOW MANAGEMENT SYSTEM

WMS is a piece of software which provides an environment to monitor and execute workflows. In other words, the Workflow Management System ensures an infrastructure where higher precision and better quality of experimental data can be executed. A workflow management system ensures an environment for the set-up and monitoring of works, arranged as a workflow. Workflow management systems also automate redundant tasks and allow the user to define different workflows for different types of processes. The workflow system ensures that the employees assigned for the next work are notified by the proper data they need to execute.

CS = COMMUNICATION SYSTEM

Today's communication systems have become very complex. Communication systems in software engineering is the design, development and maintenance of technology for communications in a wide range. To be successful in the field, first we need to discern the needs of the organization for which systems are being developed. Then we need to ensure that all elements of the system are properly connected and installed. The process will then involve monitoring, repairing and upgrading the system to ensure it as efficient as possible.

WPMS = WORK PRODUCT MANAGEMENT SYSTEM

WPMS system is a key tool in enabling a company to service the global clients. The integration of the employees within a work product management system, in addition to having the capacity to improve one document and populate another document, would be of great value to a professional. To manage work product efficiently and effectively, an improved work product management system would be very helpful to the professionals.

HRMS = HUMAN RESOURCE MANAGEMENT SYSTEM

A HRMS refers to the systems and processes that combines many human resources functions including payroll, administration, recruiting, training and performance analysis into one package. HRMS connects human resource management and information technology through HR software. When an organization wants to reduce its manual workload and processes, they choose to replace certain processes with various levels of HRMS systems. The function of the Human Resources Management System involves tracking employee abilities, skills, salary, histories and their accomplishments. The linkage of its human resource and financial through one database is the most important distinction to the developed predecessors, which makes this software application both rigid and flexible.

GDSS = GROUP DECISION SUPPPORT SYSTEM

GDSS is commonly defined as a collection of procedures, software and hardware designed for the automated support of group work. Using a GDSS, group members may rank solution alternatives, edit group documents, exchange comments or perform some other type of group work. GDSS are used to support customized projects requiring group work, input to a group and various types of meeting protocols. GDSS proponents claim that these sorts of technologies can support project collaboration through the enhancement of digital communication with various tools. GDSS is specifically engineered to support a team or other group. GDSS improved the organization of the decision-making process. GDSS helps a team of decision-makers solve problems and make choices. GDSS are targeted to supporting groups in analyzing problem situations and in performing group decision-making tasks.

SN = SOCIAL NETWORK

Social network analysis tools to adapt this architecture with development methods and to take right organizational decision upon collaboration.

SF = SESSION FACILITATOR

SF clarify the purpose of the session and the value participants will gain from attending. Session facilitator create an environment for open exploration of ideas in which all participants are comfortable expressing their opinions. Session facilitator also keep the discussion focused on the topic so that the learning objectives can be achieved within the allotted time.

SP = SESSION PARTICIPANTS

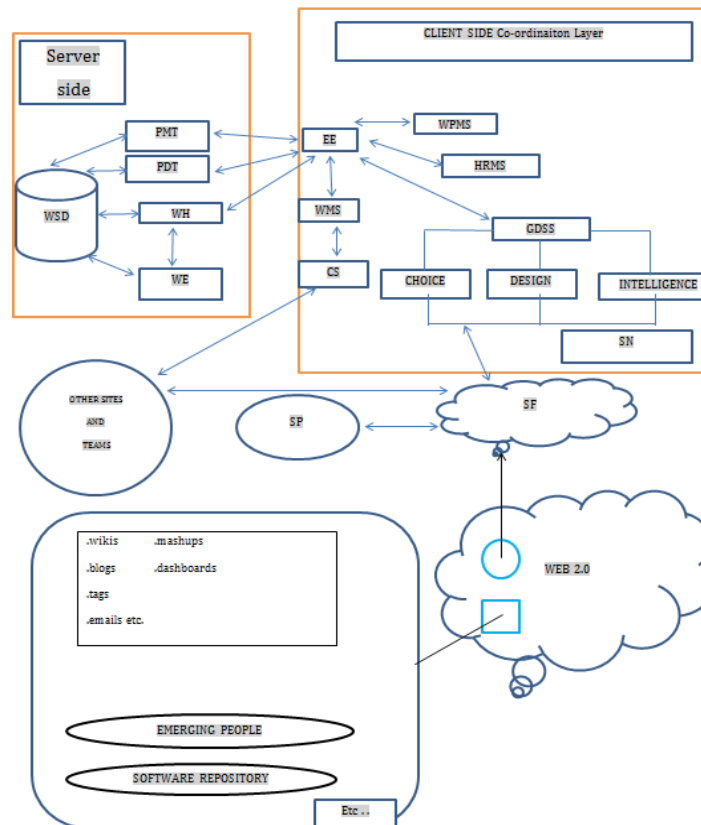SP mobilizes presenters around a particular topic.

Fig.1. Proposed SEE Architecture (BASIC FORM)

5. Short Description

This Web Based Cloud Framework [11] for "agile software engineering practice" [33] will help us to think of a development component that is at the same time "IDE" and "BROWSER" [11]. Careful use of 'off the shelf' component of Software Repository, while balancing individual potential [12] on productivity within "the architecture of participation" [34], that will lead developing "Softer Software [35]" for industry standard development. Through Social Network [4] Analysis, being aware of collaboration and co-ordination type(centralized , Centro-Multiple , Multiple , Multiple Distributed, Distributed) [35],communication pattern (expertise communication , co-ordination communication) [36][37] , it will be possible to make distributed software processes [38] as middleware and ensure effective risk management [39]. "GDSS" [40] will be used for team meeting (e.g. with stakeholders to do electronic brainstorming), requirements management [1], multidimensional separation of concerns [7] and for various decision related purposes [41].

6. Explanation

On server side, workflow database is kept for different level of access. Project management tool, process definition tool, workflow handler, workflow engine interact for data and information storing or retrieval purposes online and in updated form. On client side co-ordination layer, event engine interact with project management tool, process definition tool and workflow handler while built in communication system

communicate with workflow engine communication tool and also with other sites and teams may be geographically dispersed. For co-ordination, workflow management system in client side co-ordination layer interact with communication system and event engine. Event engine synchronize with work product management system, human resource management system and with group decision support system. Group decision support system is a parallel combination of intelligence, design and choice. This capabilities enables group decision support system to properly interact in exchanging information and relevant data with session facilitator from client side co-ordination layer. One another element in client side co-ordination layer is social network analysis tools that interact with different web technologies (suitable for web2.0 [4]). Session facilitator keep track of session participants. In this way, social network analysis tools will be able to get relevant information on emergent people, new suitable technologies, software repository and its usability etc. At last, after a rough analysis give periodic report to session facilitator.

7. Benefit

Well defined and module task decomposition with proper time and resource management will increase productivity, decrease project time delay. In functional level, different implementation technology may be employed but overall architecture will remain same for serving better industry standard software quality.

8. Discussion

Hope this architecture will help in commercial product development in medium to large scale software industry. Future research may be directed to more light and integrated module development. Practical aspects of this proposed architecture can be evaluated after deployment. But theoretical analysis may indicate practical level gain and quality improvement.

References

[1] Rafael Prikladnicki, Jorge Audy, Roberto Evaristo, "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context".

[2] Jeff Kramer, Distributed Software Engineering, Department of Computing, Imperial College, London.

[3] James D. Herbsleb, Global Software Engineering: The Future of Socio-technical Coordination Future of Software Engineering (FOSE'07), IEEE.

[4] Taisa Alves dos Santos, Renata Mendes de Araujo, Andrea Magalh ẽs Magdaleno, "Identifying Collaboration Patterns in Software Development Social Networks."

[5] William Harrison, Harold Ossher and Peri Tarr, "Software Engineering Tools and Environments: A Roadmap." IBM T.J. Watson Research Center.

[6] Filippo Lanubile, Christof Ebert, Rafael Prikladnicki, and Aurora Vizcaíno, "Collaboration Tools for Global Software Engineering", IEEE SOFTWARE.

[7] Andrew Walenstein, "COGNITIVE SUPPORT IN SOFTWARE ENGINEERING TOOLS: A DISTRIBUTED COGNITION FRAMEWORK", THESIS.

[8] Hany H Ammar, Walid Abdelmoez, Mohamed Salah Hamdi, "Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems", ICCIT 2012.

[9] All Inge Wang. Reidar Conradi, Chunnian Liu, "A Multi-Agent Architecture for Cooperative Software Engineering".

[10] BERNHARD RIEDER AND MIRKO TOBIAS SCHÄFER, BEYOND ENGINEERING, 2008 *Kluwer Academic Publishers. Printed in the Netherlands.*

[11] LAKSHMI M. GADHIKAR,LAVANYA MOHANV, MEGHA CHAUDHARI,PRATIK SAWANT, YOGESH BHUSARA, Fr. C.R.I.T, Vashi, Navi Mumbai, BROWSER BASED IDE TO CODE IN

THE CLOUD ,International Journal of Internet Computing ISSN No: 2231 – 6965, VOL- 1, ISS- 4 2012.

[12]   Vikash Lalsing, Somveer Kishnah, Sameerchand Pudaruth, PEOPLE FACTORS IN AGILE SOFTWARE DEVELOPMENT AND PROJECT MANAGEMENT , International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012

[13]   Douglas E. Lowe and Guy M. Cox, Implementing the Capability Maturity Model for Software Development, August 1996 Hewlett-Packard Journal.

[14]   Irwin Kwan, Daniela Damian, The Hidden Experts in Software-Engineering Communication (NIER Track), ICSE'11, May 21–28, 2011, Waikiki, Honolulu, HI, USA.

[15]   Laurie Williams, Pair Programming North Carolina State University.

[16]   Morisio, Maurizio et.al. "Success and failure factors in software reuse", Software Engineering, IEEE Transactions on (Volume: 28, Issue: 4).

[17]   K.T. Yeo, "Critical failure factors in information system projects", International Journal of Project Management 20 (2002) 241–246.

[18]   Anthony Finkelstein, Jeff Kramer, "Software Engineering: A Roadmap".

[19]   Jianli Dong, "Research Survey on Integrated Software Engineering Environment Based on Product Line", Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT '09).

[20]   Binghui Helen Wu, "On Software Engineering and Software Methodologies a Software Developer's Perspective", ICSE 2008.

[21]   Cybulski, J.L., La Trobe Univ., Reed, K., "A hypertext based software-engineering environment", Software, IEEE (Volume: 9, Issue: 2).

[22]   H Gomaa. L. Kerschberg., V. Sugumaran. C. Bosch. I. Tavakoli, L. Ohara, "A Knowledge-Based Software Engineering Environment for Reusable Software Requirements and Architectures", Automated Software Engineering 3. 285—307(1996), 1996 Kluwer Academic Publishers. Manufactured in the Netherlands.

[23]   Ricardo de Almeida Falbo, Fabiano Borges Ruy, Rodrigo Dal Moro , "Using Ontologies to Add Semantics to a Software Engineering Environment".

[24]   Emily Navarro, "SimSE: A Software Engineering Simulation Environment for Software Process Education", DISSERTATION, UNIVERSITY OF CALIFORNIA, IRVINE.

[25]   Dimitris Apostolou, LjiIjana Stojanovic, Tomas Pariente Lobo, and Barbara Thoenssen, "Towards a Semantically-Driven Software Engineering Environment for eGovernment", E-Government: Towards Electronic Democracy Lecture Notes in Computer Science Volume 3416, 2005, pp 157-168.

[26]   *Software Engineering Institute*, http://www.sei.cmu.edu/

[27]   Miguel Jiménez, Mario Piattini , "Problems and Solutions in Distributed Software Development: A Systematic Review".

[28]   Miguel Jiménez, Mario Piattini, "Problems and Solutions in Distributed Software Development: A Systematic Review", Software Engineering Approaches for Offshore and Outsourced Development Lecture Notes in Business Information Processing, Volume 16, 2009, pp 107-125.

[29]   Su-Hua Wang, "Software Development and Quality Problems and Solutions by TRIZ" International Symposium on Frontiers in Ambient and Mobile Systems (FAMS-2011).

[30]   Nayyar Iqbal, M. Rizwan Jameel Qureshi "Improvement of Key Problems of Software Testing in Quality Assurance", Sci.Int. (Lahore), 21(1), 25-28, 2009, ISSN 1013-5316; CODEN: SINTE 8.

[31]   Lloyd G. Williams, Ph.D., Connie U. Smith, Ph.D., "Five Steps to Solving Software Performance Problems".

[32]   Chintan Amrit et.al. , "Detecting Coordination Problems in Collaborative Software Development Environments".

[33]   Everette R. Keith, Agile Software Development Processes A Different Approach to Software Design.

[34]   Joel West et.al. "The Role of Participation Architecture in Growing Sponsored Open Source

Communities".

[35] Ta śa Alves dos Santos, Renata Mendes de Araujo, Andŕa Magalh ães Magdaleno, Bringing Out Collaboration in Software Development Social Networks, *Profes '11, June 20 - June 22 2011, TORRE CANNE (BR), Italy.*

[36] A.Scroter et.al. To Talk or Not to Talk: Factors that Influence, Communication around Changesets, CSCW'12, February 11–15, 2012, Seattle, Washington, USA.

[37] K.Nokajaki ey.al. "Supporting Expertise communication in Development Environments", chp-11, Springer-Verleg, 2010.

[38] VOLKER GRUHN, Process-Centered Software Engineering Environments ,A Brief History and Future Challenges, Annals of Software Engineering 14, 363–382,2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

[39] Rafael Prikladnicki, Marcelo Hideki Yamaguti, Risk Management in Global Software Development: A Position Paper, School of computer Science, Pontificia Universidade Cat ólica do Rio Grande do Su!, Brazil.

[40] Alexander Felfernig, Christoph Zehentner, and Harald Grabner, Group Decision Support for Requirements Negotiation, Institute for Software Technology, Graz University of Technology.

[41] Milam W.Aiken et.al., "Integrating Expert System with Group Decision Support System." ACM Transaction on Information System, Vol-9, No-1, January 1991, pp-75-95.

Authors' Profiles

Md. Yasin Ali Khan completed his B.Sc. in Computer science & Engineering from Chittagong University of Engineering and Technology, Bangladesh. He is very interested in software research.

Md. Abu Sayed completed his B.Sc. in Computer science & Engineering from American International University-Bangladesh, Bangladesh.