Integrated Approach for Software Project Management Methodology Selection

Wondwossen Elssa
Ministry of Communication and IT, Addis Ababa, Ethiopia
wondelsa@gmail.com

Fekade Getahun
Department of Computer Science, Addis Ababa University, Ethiopia
fekadegetahun@gmail.com

Abstract

The success rates of software projects are remaining too minimal over the years due to various technical and managerial issues. One aspect of the managerial issue includes selection and usage of appropriate software development and management models. In this work, an Integrated Framework for Software Project Development and Management Selection is proposed. The approach is comprehensive and inculcates the most common types of software development and project management models.

The framework has a knowledge base and inference engine components. The knowledge base component is organized from facts of completed software projects and extracted from the literature. The inference engine component has Mapper and Inference sub-components, responsible to extract appropriate software project management and software process models.

To demonstrate our approach, a prototype, Software Management Models Selection System (SMMoSS), is developed. The system accepts high level user requirements and project characteristics and returns list of highly relevant project management and software development models that fit to the need. The result of this work is believed to be a basis for software development works towards the realization of successful software projects.

Keywords: Software Development and Management Models; Software Development and Management Models Selection; Knowledge Base; Inference

1. Introduction

Most software development projects end-up far from the expectations of users. This can be supported by results such as in [2] that was obtained after evaluating software projects against the three traditional standard measurements of software success, namely time, cost, and quality.

Factors contributing for software failures could be both technical and managerial. Tesch *et al.* in [5] emphasize that the management aspects of a project are typically more critical to its success than the technical aspects. The software management aspect by itself may include various factors and identifying and selecting the right software development process and software management process models are few among them.

Software projects development and management models selection support system is characterized with diverse managerial and technical challenges. It is clear that providing a comprehensive and full-fledged solution for such an area is too vast and unimaginable. The coordinated effort of diversified professionals in the software engineering fields is vital. In this work, an attempt has been made to address one of the key aspects regarding the problem.

In this paper, an integrated software inference framework that could infer the appropriate model given system characteristics and overall user requirements (functional and non functional) of the new system is proposed.

2. Background

Unlike the rapid increment of software development projects that are globally conducted by different organizations [1], the success rates remain too minimal over the years [2]. The report in [2] shows incredible decrease in project success rates from 2006 to 2009; only 32% of all projects are considered successful (delivered on time, on budget, and with required features & functions); 44% were challenged (which were late, over budget, and/or with less than the required features and functions), and 24% failed (which are cancelled prior to completion or delivered and never used). These numbers represent very low success rates of software projects. Factors contributing to the failures could be both technical and managerial. Accordingly, the solutions to overcome these failures should involve both; as only the technical approaches would not help to improve the high failure rates of software development projects.

It is reported that most of the blames for software failures are associated with human errors, such as management and lack of skilled manpower [3]. Moreover it is showed in [4] that effective project management (PM) is vital to the success of any software development project. The authors in [5] emphasized that the management aspects of a project are typically more critical to its success than the technical aspects. According to Baccarini in [7], project success is mainly influenced by project management success that focuses on the project management process, in particular on the successful accomplishment of the project with regards to cost, time, and quality. Also types of development models have relationship with cost, time, and quality and impact project success. The RAD development model, for instance, is characterized by short duration, low cost, and low quality. On the other hand, the Waterfall model is characterized by long duration, high cost, and high quality.

Understanding this and aiming to reduce the impact of software failures caused by the management issue, so far, a number of studies have been conducted.

The studies propose many new models, frameworks, or methodologies as a solution. Lessons are learnt from them and are incorporated in this paper. On the other hand, the limitations in those works are tried to be addressed.

3. Related Work

Aiming to reduce the impact of software failures caused by management, so far, a number of studies have been conducted. For Example, Rehman and Hussain [3] suggested a generic project management approach after comparing five important project management methodologies/frameworks with the PMBOK. Decisions in which a methodology/framework is chosen or used based on a single criterion can have serious impacts. In this work, features of the most common software management models are described thoroughly. However, it does not consider software development models.

According to Schwalbe [6], only all the nine project management knowledge areas (i.e., project integration, scope, time, cost, quality, human resource, communications, risk, and procurement management) and all the five process groups (i.e., initiating, planning, executing, monitoring and controlling, and closing) of PMBOK Management Model are described with regard to software projects and other management models did not get consideration.

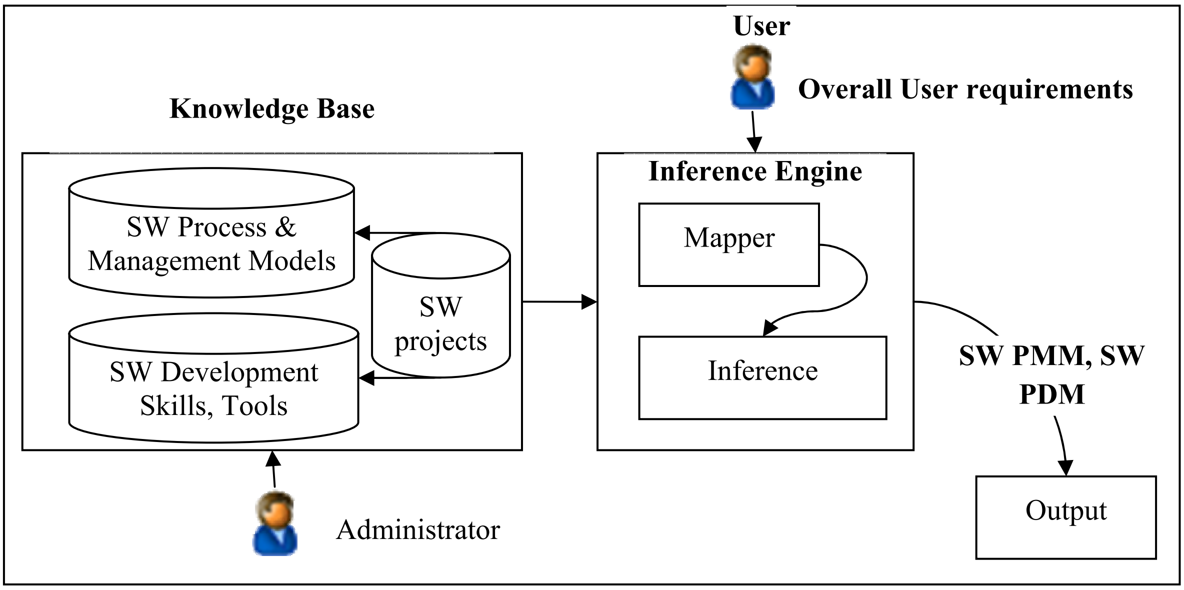
Tesch *et al.* [5] presented IT project risk factors that pose threats to successful project implementation and describe assessment of these risk factors. Finally they conclude with suggested strategies for avoiding and or mitigating these risks. The limitation of this work is software development models do not get attention at all.

Hui in [4] outlined types, advantages and disadvantages, and applicable range of software IT projects on the basis of analyzing the relationship between the whole lifecycle. But it does not consider software development models one by one.

Recently, Hewagamage and Hewagamage in [1] proposed an integration of software project management and software development practices as redesigning framework for IT project management. However, their approach is towards only the waterfall development model and PMBOK management model. Thus, their approach is not complete enough to be used in a general context and hence doesn't support the need of MCIT-Ethiopia. Also they did not provide a tool to select the best appropriate software project management and development models.

4. The Proposed Approach

The proposed approach is an integrated software inference framework that could infer the appropriate model given system characteristics and overall user requirements (functional and non functional) of the new system. These characteristics and overall user requirements would then be aligned with existing



knowledge.

Figure 1: Integrated software inference framework

Figure 1 shows the two main components of the proposed framework and the interaction between them. The knowledge base is organized from knowledge of completed software projects and facts extracted from the literature. The knowledge base contains software characteristics and user requirements. The Inference Engine component has two sub-components (Mapper and Inference). The Mapper sub-component maps or aligns the overall user requirement and software characteristics to information stored in the knowledge base. The inference sub component is responsible to extract and propose appropriate software project management model and software process model.

The proposed solution has three basic steps and the third step has five intermediary steps:

Step 1: Software Project Knowledgebase (KB) representation

Step 2: Model Project Characteristics and User Requirements

Step 3: Inference

3.1: Mapping

3.2: Building Intermediate data structure

3.3: Counting 'Y' values

3.4: Sorting

3.5: Computing Confidence Level

Step 1: KB Representation

Software Project Knowledge Base (KB) can be expressed in terms of:

- Software project Characteristics (KBC),
- Overall User Requirements (KBR),
- Software Development Model (SDM), and
- Software Management Model (SMM).

Assuming the overall requirements of a user and the characteristics of a given software project are equally important criteria and have equal weight in selecting the right software development and software management methodologies, if the total number of characteristics is 'm_1' and the total number of requirements is 'm_2', then the characteristics and requirements of the knowledgebase KB would be:

$KBC = f(P_i, KBC_1, KBC_2, \ldots, KBC_{m1})$ and $KBR = f(P_i, KBR_1, KBR_2, \ldots, KBR_{m2})$

$SDM = f(P_i, KBSDM_1, KBSDM_2, \ldots, KBSDM_q)$ and $SMM = f(P_j, KBSMM_1, KBSMM_2, \ldots, KBSMM_k)$;

where f is a function and P_i and P_i are codified projects:

$[KB = KBC]_{\downarrow}([P]_{\downarrow}i = [P]_{\downarrow}i)^{\uparrow}(full\ outer\ Join)\ KBR_{\downarrow}([P]_{\downarrow}i = [P]_{\downarrow}i)^{\uparrow}(full\ outer\ Join)\ SDM_{\downarrow}([P]_{\downarrow}i$:

Step 2: Model Project Characteristics and User Requirements

A given project (P) is defined in terms of project characteristics (PC) and overall user requirements (PR). Let the total number of project characteristics be 'x' and the total number of overall user requirements is 'y'. Accordingly the project characteristics and overall user requirements of the project P would be:

$PC = f(P, PC_1, PC_2, \ldots, PC_x)$ and $PR = f(P, PR_1, PR_2, \ldots, PR_y)$

Step 3: Inference

A structure to store the information of the project P will be created as shown in Table 1:

Table 1: Project descriptor of a given project P

PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y

Step 3.1: Mapping

Match or map each project characteristics and overall user requirements of the given project P against project characteristics and overall user requirements in the Knowledge Base KB.

For each descriptor P_y of P, if a match descriptor KB_x is found in KB, write KB_x as a match and for others mark its match as NULL. In Table 2, the second row shows KB descriptors that correspond to the given project descriptors. Thus, Table 2 represents list of all projects from the KB that have at least one match descriptor of the project P.

Table 2: Sample mapping of project P and KB descriptors

PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y
KBC_3	KBC_1			KBC_2		KBR_1	KBR_3		KBR_2

Step 3.2: Building Intermediate data structure

Modify the structure created in Step 3.1 so that it will be suitable to hold the necessary information of each project in the KB as shown in Table 3.

Table 3: Projects with matching KB desciptors

P#	PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y	
P	KBC_d	KBC_b	KBC_k				KBR_1				No of 'Y'
1											
2											
…											
n											

Step 3.3 Counting 'Y' values

For each project in KB, count the total number of 'Y' values of the KB descriptors having matching project descriptor of P (i.e., result of Step 3.2) as shown in Table 4.

Table 4: Projects with matching KB descriptors and values

P#	PC_1	PC_2	PC_3	…	PC_X	PR_1	PR_2	PR_3	…	PR_Y	
P	KBC_d	KBC_b	KBC_k				KBR_1				No of 'Y'
1		Y					Y			Y	3
2	Y	Y			Y		Y	Y		Y	6
…											
n	Y	Y			Y		Y			Y	5

Step 3.4: Sorting

Sort the result of Step 3.3 in descending order on number of yes (last column) and take the project no, and number of match of the top K results as candidates. Notice that 'no of Y' is considered here as Z value.

Step 3.5: Computing Confidence Level

For each project identified in Step 3.4, identify the corresponding SDM and SMM from the KB. In addition, compute its confidence level using equation 1.

$$\left(\frac{Z}{(X+Y)} * 100\right)\% \qquad (1)$$

where X is the number of Project characteristics and Y is the number of user requirements.

5. Prototype

SMMoSS implements *integrated approach for software project management methodology selection*. The system serves as a prototype to demonstrate the feasibility of the proposed solution. SMMoSS is a window based application developed using Visual Basic .NET 2010 Development Environment and using SQL Server 2008 R2 as back end.

In its development process, first, data such as software development models, software management models, common project characteristics, user overall requirements, and list of actively participated software development companies are identified. All these were identified or used by local software private companies. Then the data on various project characteristics with respect to software development and management models are collected and organized in a way to assist for decision. Finally the system is experimented with proper data to evaluate the feasibility of the selected models.

The functionalities are provided through Windows Explorer type interface that groups basic functionalities as data entry and knowledge inferring. Also the data entry group consists of setup sub group and developed software projects sub group. The setup sub group gives functionality to enter basic information on software development models and their characteristics, software management models and their characteristics, project characteristics, user overall requirements, and software project developing companies. The developed software projects sub group gives functionality to enter different information on completed software projects. Finally the knowledge inferring group provides functionality to accept information on user overall requirements and on characteristics of a new project and then analyzes these with the knowledge base and the science and finally yields the recommended software development and management models that best suit. Figures 2 and 3 show sample screenshots for SSMOSS Software Management & Development Models Recommendation Form.

In order to show the workability of the proposal made, two case studies are selected and experimented.

Case Study 1

An "Integrated HR and Payroll Handling Management System for Company X" from MCIT-Ethiopia is arbitrarily selected to verify the prototype. The project characteristics are: the application type is transaction-oriented, it is a medium sized project, it is a long duration system, the requirements are well clear, the requirements are stable, the project has clearly stated objectives, the assigned project manager is highly skilled and experienced (certified), and a strict requirement exists for formal approvals at designated milestones. The overall user requirements have facilities such as data storage, retrieving information, easy to use, and provision of role based security system.

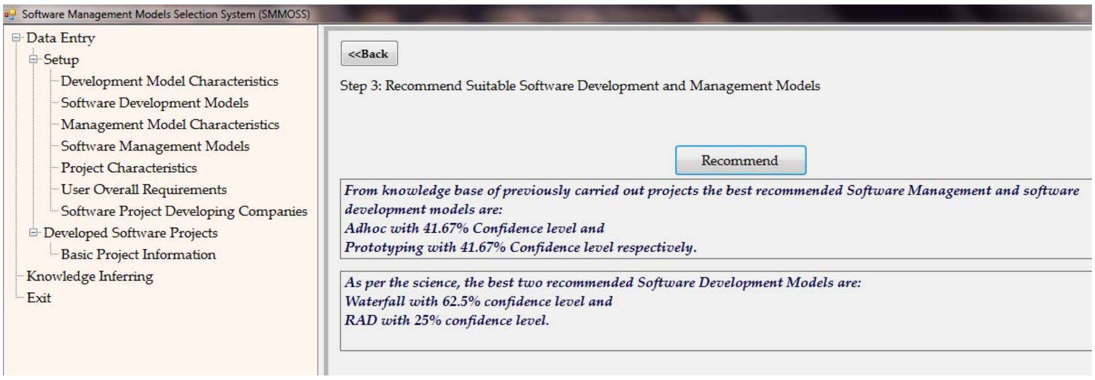


Figure 2: SSMOSS Software Management and Development Models Recommendation Form

Case Study 2

A "Static Web Page Development for Company Y" from MCIT-Ethiopia is arbitrarily selected to verify the prototype. The project characteristics are: the application type is Web informational, it is a small sized project, it is a short duration system, the requirements are not clear, the requirements are not stable, the project has no clearly stated objectives, the assigned project manager is not highly skilled and experienced (certified), and strict requirement exists for formal approvals at designated milestones. The overall user requirements are facilities such as retrieving information, local language support, local language calendar support, easy to use, and flexible to use.

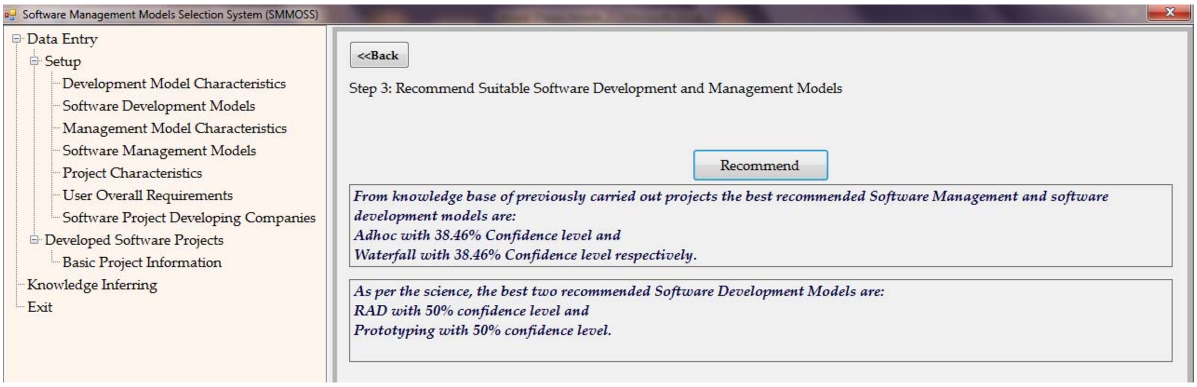


Figure 3: SSMOSS Software Management & Development Models Recommendation Form

6. Conclusion

An integrated software inference framework that could infer the appropriate model given system characteristics and overall user requirements (functional and non functional) of the new system and a Software Management Models Selection System (SMMoSS) that implements the integrated software inference framework are proposed.

References

[1] C. Hewagamage and K. P. Hewagamage, "Redesigned Framework and Approach for IT Project Management", International Journal of Software Engineering and Its Applications, Vol. 5, No.3, July 2011.

[2] Project Smart, "The Curious Case of the CHAOS Report 2009", http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html, last accessed on October 24, 2011.

[3] A. U. Rehman and R. Hussain, "Software Project Management Methodologies/ Frameworks Dynamics - A Comparative Approach", Proceedings of Information and Emerging Technologies (ICIET) International Conference, 2007.

[4] F. Hui, "Research on the Developmental Management Model of Software Type IT Project", in proceedings of Computational Intelligence and Software Engineering (CISE) International Conference, Nanjing, China, 2010.

[5] D. Tesch, T. J. Kloppenborg, and M. N. Erolick, “IT Project Risk Factors: The Project Management Professionals Perspective”, Journal of Computer Information Systems, Summer 2007.

[6] Schwalbe, K., “Information Technology Project Management”, Boston, Massachusetts: Course Technology, 2002.

[7] Baccarini, D, “The Logical Framework Method for Defining Project Success”, Project Management Journal, Vol. 30, No. 4, pp. 25-32, 1999.