

Evolvea Framework for Selecting Prime Software Development Process

Abhishek Choudhary¹, Deepak Kasgar², Lokesh Kashyap³

¹(Department of Computer Science & Engineering, Bhagwant University, India)

²(Department of Computer Science & Engineering, Bhagwant University, India)

³(Department of Computer Science & Engineering, Bhagwant University, India)

ABSTRACT-Today in era of software industry there is no perfect software framework available for analysis and software development. Currently there are enormous number of software development process exists which can be implemented to stabilize the process of developing a software system. But no perfect system is recognized till yet which can help software developers for opting of best software development process. This paper present the framework of skillful system combined with Likert scale. With the help of Likert scale we define a rule based model and delegate some mass score to every process and develop one tool name as MuxSet which will help the software developers to select an appropriate development process that may enhance the probability of system success.

Keywords: Likert Scale, Software Development Process, Skillful System

I. Introduction

The issue of selecting an acceptable software development process for the projects has been addressed in multiple ways by many industry experts but there is no appropriate framework developed till yet. This paper extends prior work by considering the expert system with the Likert scale for selecting leading software development process.

The aim of producing a rule based skillful system is to come up with the result that would assist to establish which software development process is distinctly suitable for a particular project. In other words exploring one clear answer was not the aim, but rather getting guidance in choosing from a known set of processes. The model selected should match the characteristics of a given project should correspond to the criteria put out in the selection of process, the criteria should be stored in knowledge base of the system, then obtaining information about the suitable methodology. The user of such a program would have to discover the characteristics of their project by answering a set of questions asked by the system.

1.1 Software Development Processes

A software development process describes a scenario that is used to organize, plan, develop, and maintain a software system. There are many software development processes and these processes contain some basic stages of software development life cycle. These stages are planning, analysis, design, implementation and maintenance. In this paper we compare three software development methodologies. These are waterfall, spiral and prototype methodology.

Waterfall Development Methodology

The waterfall model process is the linear approach in which development is seen as flowing continuously descending with the phases namely requirement analysis, design, implementation, testing, integration and maintenance. Each phase of waterfall model has well defined begin and conclusion criteria.

Waterfall development process is most exact when-

- Requirements are well documented, clear and fixed from the starting phase.
- Finale product is stable.
- Technology is static and recognized.
- The project is small and short.
- Not good process for complex and object oriented model project.
- Strain in adapting modifications after project development.

Spiral Development Process – Boehm defines spiral model as process generator model.

Spiral methodology is most appropriate when-

- Risk assessments are important.
- Consumers are not sure about their requirements.

- Requirements are complicated and needs comprehensibility.
- It can act well with the changing user requirements.
- This process is mainly used for large projects.
- Forecast is high.
- Small documentation is needed as compared to waterfall development process.

Prototype Development Methodology

In Prototype development process initially the working prototype is developed instead of developing the actual software. In this process to understand clear requirements a prototype is built before design and coding.

Prototype process is most exact when-

- Prototyping is an irresistible idea for complex and large systems for which there is no manual process or existing system to help control the requirements.
- This software methodology is used when it is hard to gather all the requirements of customer.

1.2 Expert system

Expert system can be used in various research fields [8]. In this paper we extend the foregoing work by considering the expert system in the field of software development model. The expert system is computer program that emulate the decision making ability of the human experts. There are three elements of expert system i.e. knowledge base, user interface and inference engine. The knowledge base contains all the knowledge and directive about a particular problem and provides it to the inference engine when they require. The inference engine is the program that inspects the issue and obtain conclusion by implementing logical processing. The user interface works as an interface between a system user and the expert systems.

II. Literature Study

Data study exhibit that investigation has been taken on multiple facets of software development process but there is no exact framework and tool being developed for the selection of best software development methodologies among multiple. In 2005 M. AYMAM AL AHMAR [1] present a model of expert system supported with object oriented modeling that assists in software development process selection process. Abdur Rashid Khan, Zia Ur Rehman and Hafeez Ullah Amin [2] concentrated on some approaches like fuzzy logics, certainty factors and analytical hierarchy factors to evolve expert system and they developed expert system named as ESPMS (expert system process model selection). This system was developed using expert system for text animation as a development tool. More work has been done in the area of software development process but nearly of these works are the comparative study between these methodologies [3][10]. In 2010 Nabil Mohammed Ali Munauar and A Govardhan [4] made a contrast between five different processes and show the features and defects of each methodology. When the software developer selects a development model, risk is a mandatory consideration which affects the selection process of software development process. Haneen Hijazi and Thair Khmour [5] explore the risk that exist in the development processes and define risk management. In this paper we use rule based expert system as an important component to discriminate between processes. Relevant study reveals that expert system can be implemented in multiple fields and knowledge base is a component of expert system which stores knowledge about a particular problem. M. Darbari, N. Dhanda [7] presents the model driven knowledge representation framework. Sunita Bansal and M. Darbari [6, 14] define a knowledge base expert system for managing business dynamics. In 2012 Sudhakar Tripathi, Arvind Kumar Tiwari and R.B. Mishra [11] create a rule base model for clustering gene expression data. In 2009 M.S. Josephine, Dr. K. Sankara Subramanian [13] used an expert system for software error detection and correction. In this paper a rule based expert system is created to combine with Likert scale measurement that guides in the selection process of the software development [9] process.

III. Proposed Framework

The proposed framework is shown below:

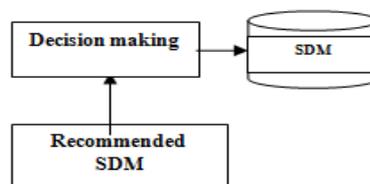


Figure 1. Proposed Framework

IV. Likert Scale

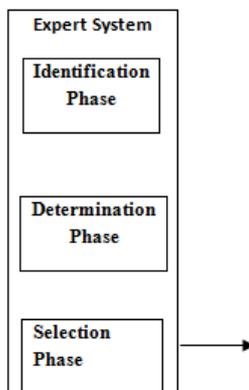
Likert scale is a kind of rating scales which is used to calculate the performance directly. Likert Scale is a five (or seven) point rating scale which permits the user to convey their thoughts means how much they agree or disagree with a problem declaration. User gives their acknowledgement by choosing a Likert item. A Likert item is a word or statement which the user is asked to evaluate according to the given criteria. Likert items are used to measure the level of agreement or disagreement. We use 5 Likert items i.e. poor, fair, average, good and excellent.

Features	Poor	Fair	Average	Good	excellent
Req. specification	1	2	3	4	5
Complexity of system	1	2	3	4	5
Time schedule	1	2	3	4	5
Cost	1	2	3	4	5
Documentation	1	2	3	4	5
Project size	1	2	3	4	5
Change incorporated	1	2	3	4	5

Table1: Likert scale measurement scale

4.1 Selection Parameters for Selecting Best Development Methodology

No one process is ideal so we develop a framework for picking a process which depends on multiple components and project characteristic and selection boundary. These selection boundaries [11] are: requirement specification, complexity of system, time, implementation cost, core documentation, module size and change incorporated.



4.2 Comparison among Waterfall, Spiral and Prototype Model

Features	Waterfall	Spiral	Prototype
Req. specification	Beginning	Frequently changed	Beginning
Complexity of system	Simple	Moderate	Complex
Time	Less	Long	Long
Cost	Low or almost as estimated	High or above budget	Expensive
Core Documentation	Necessary	Yes but not much	Yes
Module size	Large scale	Low to medium scale	Large scale and complex
Change incorporate	Difficult	Easy	Easy

V. Rule Based Model For Selecting Software Development Methodology

Rule based model have been generated with the help of Likert scale measurement.

Rule1:

IF Requirement specification ≤ 1 and
 Complexity of system ≤ 3 and
 Time ≤ 4 and
 Cost ≤ 2 and
 Documentation ≤ 5 and
 Project size ≤ 4 and
 Change incorporated ≤ 2 and
 THEN Waterfall Model (Score ≤ 21)

Rule2:

IF Requirement specification ≤ 3 and
 Complexity of system ≤ 4 and
 Time schedule ≤ 4 and
 Cost ≤ 4 and
 Documentation ≤ 3 and
 Project size ≤ 3 and
 Change incorporated ≤ 3 and
 THEN Spiral Model (Score ≤ 24)

Rule3:

IF Requirement specification ≤ 4 and
 Complexity of system ≤ 5 and
 Time schedule ≤ 5 and
 Cost ≤ 5 and
 Documentation ≤ 4 and
 Project size ≤ 5 and
 Change incorporated ≤ 5 and
 THEN Prototype Model (Score ≤ 33)

5.1 Decision Making Tool

The resolution for adopting particular software development process depends on the selection parameters and final score assigning to the methodologies. The prioritization and strategy for selecting the development methodology might be refined; the important consideration is that the decision is made explicitly as shown below:

Limitation priority

C1: If documentation is mandatory and requirements are stable or unchanging, then it should automatically opt the waterfall process.

C2: If risk in project is high then it should automatically select the spiral process.

C3: If requirements are unspecified then it should automatically select the prototype methodology.

Constraints	Priority
C1: If documentation is necessary and requirements are stable or unchanging then it should automatically select the waterfall methodology.	5
C2: If project risk is high then it should automatically select the spiral methodology.	7
C3: If the requirements are undefined then it should automatically select the prototype methodology	8

Table3: Constraints and priorities for SDM

5.2 Methodology Selection Tool: Muxset

In order to encourage the process selection rule we have implemented a tool provide automated support for decision of development process selection. We have implemented a tool called “MuxSet” selector shown in figure. A tool consists of 5 basic selections with check boxes and conditions.

Model Selection(Mod Sel)		
<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;"> Software Development Methodologies </div> <div style="padding: 5px;"> Waterfall Model <input type="checkbox"/> Prototype Model <input type="checkbox"/> Spiral Model <input type="checkbox"/> </div>	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;"> Selection Parameters </div> <div style="padding: 5px;"> Requirement Specification <input type="checkbox"/> Complexity of System <input type="checkbox"/> Time schedule cost <input type="checkbox"/> Documentation <input type="checkbox"/> Project Size <input type="checkbox"/> Change Incorporated <input type="checkbox"/> </div>	
Final Score	Priorities(0-10)	Result
$5 \geq 10$ <input type="checkbox"/> $10 \leq 5 \leq 19$ <input type="checkbox"/> $20 \leq 5 \leq 25$ <input type="checkbox"/> $26 \leq 5 \leq 30$ <input type="checkbox"/>	C1 <input type="checkbox"/> C2 <input type="checkbox"/> C3 <input type="checkbox"/>	<div style="border: 1px solid gray; padding: 5px; width: 50px; margin: 0 auto;">Model</div>
<div style="display: flex; justify-content: space-around; gap: 20px;"> Decide Cancel View Heuristic Report </div>		

Figure2. Model Selection Tool

The first block represents the software development process. The next block represents selection criteria, final score, dynamic priority allocation and results. Expect the result block all the blocks have check boxes and the priorities can be assigned between 0 and 10. After making the entire selections “Decide” button is pressed and the result is displayed in the result text box giving the model selected. “View Heuristic Report” button gives the entire report with situational analysis.

VI. Conclusion And Future Work

This study proposed a rule base expert system combined with Likert scale measurement that will become a base for the software engineers in the selection of best software development methodology for the project. In this paper, we provide a tool that will help in the opting process of development process. The future work will include more software development process and many selection parameters.

REFERENCES

- [1]. M.Ayman Al Ahmar, “Rule based expert system for selecting software development methodology”, *Journal of theoretical and Applied Science*, 2005-2010.
- [2]. Abdur Rashid khan, Zia Ur Rehman and HafeezUllahAmin, “Knowledge based system’s modeling for software process model selection”, *IJACSA*, Vol.2, No.2, Feb2011.
- [3]. Vishwas Massey, K.J.Satao “Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology”, *IJARCSSE*, Vol.2, Issue4, April2012.
- [4]. Nabil Mohammed Ali Munassar and A. Govardhan, “A Comparison Between Five Models Of Software Engineering”, *IJCSI*, Vol.7, Issue5, September2010.
- [5]. HaneenHijazi, TthairKhdour and abdulalamalarabeyyat, “A Review of Risk Management in Different Software Development Methodologies”*InternationalJournal of Computer applications*, Vol.45-No.7, May2012.
- [6]. Dr. Sunita Bansal and M Darbari, “Designing and Knowledge Based Expert System for Handling Business Dynamics”, *International Journal of Scientific & Engineering Research Volume2* , Issue 11 , March -2011.
- [7]. M Darbari, N Dhanda, “Applying Constraints in Model Driven Knowledge Representation Framework” *International Journal of Hybrid Information Technology* 3 (3),4, 2010.
- [8]. Folorunso, I. O.,Abikoye, O. C.,Jimoh, R. G. and Raji, K.S, “A Rule-Based Expert System for Mineral Identification”, *Journal of Emerging Trends in Computing and Information Sciences*, Vol.3, No.2, February 2012.
- [9]. M Darbari, S Medhavi, AK Srivastava Development of effective Urban Road Traffic Management using workflow techniques for upcoming metro cities like Lucknow (India), (2)4, 2008.
- [10]. Dinesh Ch. Jain and ShikhaMaheshwari, A Comparative Analysis of Different types of Models in Software Development Life Cycle, *IJARCSSE*, Vol.2, Issue5, May 2012.
- [11]. IA Siddiqui, M Darbari, “Application of Use Case for Identification of Root Cause of the Dependencies and Mutual Understanding and Cooperation Difficulties in Software Systems, *International Journal of Applied Software Engineering*, 4, 10-20.
- [12]. SudhakarTripathi, Arvind kumar Tiwari and R.B.Mishra, Rule based mosel for clustering gene expression data, International conference on Artificial Intelligence & Soft Computing, December-2012.
- [13]. M.S.Josephine, Dr. K.Sankara Subramanian, Software Error Detection And Correction Using Layer Base Approach In Expert System, *InternationalJournal Of Reviews In Computing*, Vol 4, 2012.
- [14]. Dr. SunitaBansal, M Darbari, “Application of Multi Objective Optimization in Prioritizing and Machine Scheduling: a Mobile Scheduler Toolkit”, *International Journal of Advance Information Systems*,Vol3, No. 2, 2012.