Comparative Analysis Of Software Process Model Using Neuro Fuzzy Classifier

Dr. Megha V. Polishwala[1], Dr. Viral Polishwala[2]
[1]*Assistant Professor, Vivekanand College of Advanced Computer and Information Science, Nr. Saroli Bridge, Jahangirpura, Surat*
[2]*Assistant Professor, Sutex Bank College of Comp. App. & Science, Amroli, Surat*
doi.org/10.64643/IJIRT211655

***Abstract*—In the area of software development, software is significant part for a long time. To build the accurate and reliable software, software process models play an important role. The main goal of this paper is to represent comparison study of different models and analysis of waterfall model and RAD model using neuro fuzzy classifier that shows which model gives better performance to build the reliable software product. This analysis is done on the basis on parameters of both the model. The software development has always been characterized by parameters that contain level of fuzziness. Uncertainty issues are tackled using fuzzy logic techniques. Fuzzy Logics could produce better estimation of accuracy provided that various parameters and factors pertaining to fuzzy logic are carefully set. A neuro-fuzzy classification model is proposed. This model has classification rules as well as membership function for each features' value. The primary purpose of the paper is to compare the features of both models and provides which model give best result to develop a new software product.**

***Index Terms*—Waterfall model, RAD model, Increment model, Comparison Analysis, Concept of Fuzzy Logic and Artificial Intelligence (AI), Result and Discussion, Conclusion.**

I. INTRODUCTION

From the past few decades, software development industry has been growth very fastly. Software development become standard practice of software industry. Software development follows the software process models to build a software product. To produce a good quality and reliable software, developers follow the software process models. There are various kind of software models available in the market. But as the IT industry growth rapidly, the new approach of the software development is required for the IT industry. The latent benefits from reducing cost to improving quality and challenges faced in software development are highlighted by many researchers [6]. One of the challenges is that which model is best suitable according to the requirement of software product. Selection of software process model for the software product is effect on the cost, time, reliability of the software product. There are various software models available in the IT industry. Each model describing variety of activities that take place during the process. Proper selection of model gives very high impact on the testing. The basic activities those are broken down into steps requirement analysis, design, coding, and testing. The next step is "maintenance" that is considered after the system deployment. Software doesn't wear out [4] unlike hardware. During testing phase, some errors or bugs are found by the developers. But some errors or bugs are not found by developers or software engineers, those are found by the customer when the product is delivered to the customer. These errors or bugs are required to solve immediately.

Various models are available for the software development:

- Waterfall model
- Iterative model
- RAD model
- Agile model
- Spiral model
- Incremental model

In this paper, Waterfall model and RAD model are discussed in brief. Which kind of performance are given by these models, is discussed here. This

performance is found from the features of the models using neuro fuzzy classifier.

II. WATERFALL MODEL

The oldest and most classical model is waterfall model in era of software engineering. It is widely used model in government projects and in many major companies [5]. This model is proposed by Winston Royce in 1970 [3]. Waterfall model contain 7 non overlapping stages. Model begins with requirements and end with maintenance. Other stages are in between architectural design, detail design, coding, testing.

Following are the advantages of waterfall model:

- It is easy to implement and understand.
- Define before design and design before code.
- When requirements are very well understood, and limited, then it works well.

III. RAD MODEL

Rapid Application Development model is type of incremental model [7]. In this model, entire team meets at the beginning stage to decide requirement and fundamental design of the software product. Then RAD teams break out of this cycle. After defining the project requirements, the developers model the structure, interaction of object needed to implements the requirements. After completion of one iteration, product is delivered to the customer and used by the customer then customer gives any necessary changes, those changes are accepted by the developers. These changes are implemented in the next iteration. This cycle will continue until system is developed. The main advantage of the RAD is its insistence on design phase. Because before implementation begins, developers and customers agree on a design. Because of RAD is not designed to accommodate substantial change, it is difficult to define up front, problems are likely for project that is very dynamic [5].

IV. INCREMENTAL MODEL

It combines waterfall models elements and applied in an iterative fashion. This process model is effective in the situations where requirements are defined precisely and there is no confusion about the functionality of the final product. The whole requirement is divided into various builds [8]. By converting in multiple process cycles, it makes "multi-waterfall" cycle [8]. Each release delivers an operations product [10]. This model is flexible compare to waterfall model.

V. SPIRAL MODEL

Spiral model is similar to Increment model in point of risk analysis [6]. Software project is passes through different phases in iteration which is call spiral [14]. This model is developed by Boehm in 1988 [8]. A risk is initially started with planning, requirements gathering. Uncertainly which is inherited to software project, does not handle by this model. Important software projects have failed because project risks were neglected.

VI. AGILE METHODOLOGY

Agile software development is basically a conceptual framework for software engineering this promotes development iteration throughout the life cycle of the project say whatever the life cycle that have come across the entire phase will be in consideration from inception elaboration development. Many different types of agile development method exist today, like scrum.

VII. COMPARISON STUDY OF VARIOUS MODEL

Table 1: Comparison analysis of various process model [18] [19]

Parameter / Process model	Waterfall model	Prototype model	Incremental model	Spiral model	RAD model	Agile
Simplicity	Simple	Simple	Intermediate	Intermediate	Very simple	Complex
Cost	Low	High	Low	Expensive	Low	Very High
Requirement Analysis	Beginning	Frequently Changed	Beginning	Beginning	Timebox Released	Frequently Changed

Maintenance	Least Glamorous	Routine maintenance	Promote	Typical	Easily	Promote
User Involvement	Only at beginning	High	High	High	Only at beginning	Initial phase
Risk Analysis	Yes	No	Yes	Yes	Low	Yes
Reusability	Limited	Weak	Yes	Yes	Some Extent	Usecasereuse
Time Required	Long	Short	Very Long	Long	Short	Least possible
Flexibility	Rigid	Highly	Less Flexible	Flexible	High	Highly

VIII. ANALYSIS STUDY OF MODELS USING CONCEPT OF FUZZY LOGIC AND AI

The development of software has always been characterized by the parameters that contain certain level of fuzziness. This requires some degree of uncertainty be introduced in the models, to make the model reliable. Fuzzy Logic techniques are used to tackle the uncertainty issues. Fuzzy Logics could produce better estimates provided that various parameters and factors pertaining to fuzzy logic are carefully set.

The above table 1 shows the comparison analysis of six different process models. From the above table, it is not clear that which model gives the better performance. It is not be identify easily the performance factor, error measurement, from the above textual data. So, it is necessary to identify the performance analysis using statistical way.

One of the ways to identify the performance factor of above model using statistical way is using concept of fuzzy logic and artificial neural network. To identify the solution of real-world problems, fuzzy logic and neural networks are providing the effective way. Fuzzy Logic is a method that resemble human reasoning. It refers to the things which are vague. Fuzzy logics provides very valuable flexibility for reasoning. Fuzzy logic represents intermediate value to presents which is partially true and partially false. Fuzzy logic algorithm helps to solve a problem after considering all available data. Hybrid neuro-fuzzy systems are homogeneous and usually resemble neural networks. Fuzzy sets can be regarded as weights whereas the input and output variables and the rules are modeled as neurons.

In table 1, data are represented in the textual form. It is necessary to convert this crisp data into fuzzy data using fuzzy membership function are available in MATLAB like trapezoidal function, triangular membership function etc. Here I am going to use evalmf function to perform the fuzzification process of crisp data.

To apply the fuzzy membership function, the crisp data must be converted into the dummy value. This process of converting the data, is known as encode to decode. Using this process, textual data is converted into numeric data. Say for ex: Take parameter, requirement analysis, which contain three textual values, 1) beginning 2) frequently changed and 3) timebox released. To convert this textual value into numeric one, assign 1,2,3 respectively. For beginning-1, frequently changed -2, Timebox released -3. Like this way, data set can be prepared for the 14 parameters listed in table1. By applying evalmf() function, all these numeric values are converted into fuzzy member.

This dataset is now used into the artificial nueral network model. To build the model, feature_matrix is created with the prepared data set. By applying the fuzzy rules on the feature_matrix, model is developed that will identify the performance of the models. MSE, RMSE, regression is also identified within this development. By viewing the performance in terms of MSE, it can be easily identifying that which model gives better performance. Model selection process become easier for the development team as well as client.

Table 2: Comparison of MSE of Process Models

Process Model	MSE
Waterfall model	0.04148
Prototype model	0.14683
Incremental model	0.060759
Spiral model	0.18053
Iterative model	0.11557
RAD model	0.034322
Agile model	0.14683

The accuracy result of the model helps to select the better model to develop the software system. Selection of appropriate model to build the software product, reduces the software failure ratio. The below comparison table gives idea about which model have better MSE value:

The above table gives better idea about the performance of the process models by viewing the MSE value of each process model. From this comparison study, developer get clear idea of performances of the process models. When developer team start to develop any software system, they have to decide the which model is suitable according to users' requirements.

IX. CONCLUSION

There are many SDLC models such as waterfall, prototype, spiral, incremental, RAD, Agile etc. used in various organizations. Each model has its own pros and cons. In software industry, all these methodologies are used, so here the comparison of all these models is provided basis of certain features like – requirement specification, cost, simplicity, expertise, risk involvement, flexibility, maintenance, etc. from comparison through these basic features helps developers to select appropriate model for particular project. Selecting the correct life cycle model is most valuable process while developer has to complete within a given time deadline and estimated cost. This study makes SDLC selection process easy and effective for the system.

REFERENCES

[1] R. Kaur and J. Sengupta, "Software Process Models and Analysis on Failure of Software Development Projects," *International Journal of Scientific & Engineering Research*, vol. 2, no. 2, Feb. 2011, ISSN: 2229-5518.

[2] Asmita, Kamlesh, and Usha, "Review on Comparative Study of Software Process Model," *International Journal of Science, Technology & Management*, vol. 4, Special Issue 1, Mar. 2015, ISSN: 2394-1537.

[3] A. Chopra, A. Prashar, and C. Saini, "Software Projects Models," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 1, no. 4, ISSN: 2347-4289.

[4] A. P. Gosavi and J. B. Pandav, "Analysis of the Different Approaches for Software Project Management," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 1, pp. 421–425, Jan. 2015, ISSN: 2277-128X.

[5] A. Kaur, "Comparative Analysis of Reliability Models—Based on Uncertainty Factors," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 2, pp. 481–488, Feb. 2015, ISSN: 2277-128X.

[6] J. Haweliya, M. Sharma, and B. Nigam, "Comparative Analysis of Model Based Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 4, pp. 356–360, Apr. 2015, ISSN: 2277-128X.

[7] R. R. Raval and H. M. Rathod, "Comparative Study of Various Process Model in Software Development," *International Journal of Computer Applications*, vol. 82, no. 18, pp. 16–19, Nov. 2013, ISSN: 0975-8887.

[8] Asmita, Kamlesh, and Usha, "Review on Comparative Study of Software Process Model," *International Journal of Science, Technology & Management*, vol. 4, Special Issue 1, pp. 68–75, Mar. 2015, ISSN: 2394-1537.

[9] J. K. S. Jasmine and R. Vasantha, "A New Process Model for Reuse Based Software Development Approach," in *Proc. World Congress on Engineering (WCE)*, vol. 1, Jul. 2–4, 2008, ISBN: 978-988-98671-9-5.

[10] A. B. Sasankar and V. Chavan, "SWOT Analysis of Software Development Process Models," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 5, no. 3, pp. 390–399, Sep. 2011, ISSN: 1694-0814.

[11] H. Natarajan, R. K. Somasundaram, and K. Lakshmi, "A Comparison Between Present and Future Models of Software Engineering," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 2, no. 2, pp. 47–50, Mar. 2013, ISSN: 1694-0814 (Print), ISSN: 1694-0784 (Online).

[12] A. Alshamrani and A. Bahattab, "A Comparison Between Three SDLC Models: Waterfall Model, Spiral Model, and Incremental/Iterative Model,"

IJCSI International Journal of Computer Science Issues, vol. 12, no. 1, no. 1, pp. 106–111, Jan. 2015, ISSN: 1694-0814 (Print), ISSN: 1694-0784 (Online).

[13] V. Massey and K. J. Satao, “Comparing Various SDLC Models and the New Proposed Model on the Basis of Available Methodology,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 4, pp. 170–177, Apr. 2012, ISSN: 2277-128X.

[14] N. Jain and A. Jain, “Software Development Life Cycle: A Detailed Study,” *International Journal of Advanced Research in Computer Science*, vol. 2, no. 3, pp. 261–264, May–Jun. 2011, ISSN: 0976-5697.

[15] S. Chauhan and D. Jha, “A Formation to Recover Software Process Improvement Model for Small Organization,” *International Journal of Advanced Research in Computer Science*, vol. 4, no. 4, pp. 153–154, Mar.–Apr. 2013, ISSN: 0976-5697.

[16] R. Kaur and J. Sengupta, “Software Process Models and Analysis on Failure of Software Development Projects,” *International Journal of Scientific & Engineering Research*, vol. 2, no. 2, Feb. 2011, ISSN: 2229-5518.

[17] M. Sharma, “A Survey of Project Scenario Impact in SDLC Models Selection Process,” *International Journal of Scientific & Engineering Research*, vol. 2, no. 7, Jul. 2011, ISSN: 2229-5518.

[18] A. Mishra and D. Dubey, “A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 5, pp. 64–69, Oct. 2013, ISSN: 2321-7782 (Online).

[19] R. K. Vaghela, “A Comparative Analysis of Software Development Life Cycle Models,” *International Journal of Scientific Research*, vol. 4, no. 6, pp. 512–515, Jun. 2015, ISSN: 2277-8179.

[20] S. Srivastava and A. Awasthi, “Software Process Improvement,” *International Journal of Computer Science & Information Technology*, vol. 4, no. 9, pp. 154–160, Sep. 2014, e-ISSN: 2231-2471, p-ISSN: 2319-2224.