

Analyzing The Efficacy Of Requirements Stability Based On Function Point Modeling

D. Francis Xavier Christopher¹, Dr. E. Chnadra²

¹ Director, School of Computer Studies, RVS College of Arts and Science
Coimbatore, Tamil Nadu 641402, India

² Director, School of Computer Studies, SNS Rajalakshmi College of Arts and Science
Coimbatore, Tamil Nadu 641049, India

Abstract – A good requirement should possess the attributes such as specific, unambiguous, verifiable, quantifiable, measurable and traceable. Requirements gathering, eliciting, organizing, analyzing and managing the requirements are the key factors for the project success. Many software projects fail due to instable requirements and lack of managing the requirements changes efficiently. Software Requirements Stability Index Metric (RSI) helps to evaluate the overall stability of requirements and also keep track of the project status. Maximizing the requirements stability will obviously reduce the change impact i.e., higher the stability, less changes tend to propagate. This review paper analyzes the efficacy of measuring the Requirements Stability based on the Function Point Modeling and the regression analysis.

Keywords: Requirements Engineering, Function Point, Requirements Stability Index, Regression Analysis

I. Introduction

Software system needs to evolve. In particular, changes in the requirements may be related to the addition of new functionalities, modification to the existing ones, deleting the functionalities which are obsolete or to the improvement in the quality of service offered [1][3][4].

Software developers often start with unclear, ambiguous, and incomplete requirements with inaccurate understanding of the user needs or insufficient requirements. Therefore, requirements development and management are the starting points of the software development process [2].

Requirements changes not only cause software defects but also cause in delay of delivery of the software project. Requirements changes at the later stage can cause uncertainty in the software development. Sometimes, these requirement changes will affect the quality of the software. For requirements engineering, the challenging issue is not the requirements change. It is how to deal with the change and how to measure them. Maximizing the requirements stability will obviously reduce the change impact during the development process [5].

This paper discusses the background on Function Point Analysis and the use of function point analysis for predicting the Requirements Stability and regression analysis in order to maximize the requirements stability.

II. Background on Function Point Analysis

Function Points are the standard unit of measure that represents the functional size of the software application. Function Points (FPs) was originally developed by Alan Albrecht, who used this approach to measure the functions independent of technology and language [1] [4].

Once a project has been approved and the function point count has been established, it becomes a relatively an easy task to identify, track and communicate new and changing requirements. As requests come in from users for adding, modifying and deleting the functionalities, function point counts are calculated and updated. This result is then used to determine the impact on budget and effort. The user and the project team can then determine the importance of the request against its impact on budget and schedule. At the conclusion of the project the final function point count can be evaluated against the initial estimate to determine the effectiveness of requirements gathering techniques. This analysis helps to identify opportunities to improve the requirements definition process [9].

Function Point Analysis can be used to describe and document the functional requirements of the software application [1][3]. It analyzes the requirements in order to quantify the functional size of the project. A function point count at the requirements phase of a project can be used for estimating and for setting a baseline for managing the requirements in the scope of the project [1][3].

III. Method to Compute Function Point (FP)

Function Point (FP) Calculation can be divided into three steps [1]:

- FPs is derived based on countable measure of software's information domain and qualitative assessments of complexity of the software. The five information domain or feature domain values are External Inputs, External Outputs, External Inquiries, Internal Logical Files and External

Interface Files. Use the below table1: Feature Domain Value Complexity [3] to compute weighted factor which depends on the complexity of each domain value.

	Low	Average	High
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

Table1: Feature Domain Value

Then use the following equation to compute the Function Count.

$$FC = \sum_{i=1}^3 \sum_{j=1}^5 W_{ij} C_{ij} \text{ -----Equation 1}$$

Where, W_{ij} . Weight each of the domain values for row i and column j .

C_{ij} - Number of each function feature with complexity weight.

For example, from table 1, the weight of low complexity External Input is 3 and weight of High Complexity Internal Logical File is 15.

- Secondly, the complexity factors need to be evaluated as described by 14 general systems characteristics such as 1. Data Communication, 2. Distributed Data Processing, 3. Performance, 4. Heavily Used Configuration, 5. Transaction Rate, 6. Online Data Entry, 7. End-User Efficiency, 8. Online Reusability, 9. Complex Processing, 10. Online Update, 11. Installation Ease, 12. Operational Ease, 13. Multiple Sites, 14. Facilitate Change. The values of each characteristic are weighted on a scale of 0-5 where 0 – Not Present, 1 – Minor Influence to 5 – Strong Influence. Based on the characteristics value, the Adjustment Factor can be computed as below:

$$AF = 0.65 + 0.01 * \sum (Fi) \text{ -----Equation 2}$$

Where Fi - denotes the weighted scale of the characteristics value

- Thirdly, compute the Function Points using Equation 1 and Equation 2 as given below:

$$FP = FC * AF \text{ -----Equation 3}$$

Where FP – denotes the Function Points

FC denotes the Function Count from Equation 1

AF –denotes the Adjustment Factor from Equation 2

Although the FP measure doesn't correspond to any actual physical attribute of a software system (such as lines of code or the number of subroutines) it is useful as a relative measure for comparing projects, measuring productivity, measuring the functional requirement changes and estimating the effort and time needed for developing a project [1][9].

IV. Predicting Requirements Stability Index using FP

Many software projects fail due to instable requirements and lack of managing the requirement changes efficiently [1][3][5]. This paper analyzes the Function Points (FPs) to measure the software requirement changes during the software development life cycle (SDLC). Based on this measurement, Software Requirement Stability can be predicted.

The Function Point of Requirements is different at different time during SDLC. The function point calculated during the design phase will be different with the function point calculated during the coding phase. Therefore Function point $FP(i)$ is associated with the timestamp τ_i . All of Function Points(FPs) will compose a sequence $\{FP1, FP2, FP3.....FPN\}$ with different timestamp τ_i .

Hence the Function Point-Timestamp will compose the ordered sequence set as below [3]:

$$FP - \tau = \{(FP1, \tau_1), (FP2, \tau_2), (FP3, \tau_3), \dots, (FPN, \tau_N)\}$$

The Software Requirement Change can be represented as transition from one Function Point State to another Function Point State as below

$$(FPi, \tau_i) \xrightarrow{\text{RequirementChange} \Delta FP_{\text{Change}}} (FP(i+1), \tau_{i+1}) \text{ ----- Equation 4}$$

From equation 4, the function point change can be represented as follows:

$$\Delta FP_{\text{CHANGE}} = \int_i^{i+1} FP dx \rightarrow FP(i+1) - FP(i)$$

The requirement change can be classified as

1. Add a new Functionality
2. Modify the Existing Functionality
3. Delete the obsolete Functionality

The related function point changes for add, update, delete functionality can be $\Delta FP_{\text{ADD}}, \Delta FP_{\text{UPDATE}}$ and

ΔFP_{DELETE} respectively. Compute the Function points change (Add, Update and Delete) as follows [3][8]:

$$\Delta FP_{ADD} = \frac{\text{Number of Functional points Added / Total Number of Initial Function Points}}{\text{at that timestamp}}$$

$$\Delta FP_{UPDATE} = \frac{\text{Number of Functional points Modified / Total Number of Initial Function Points}}{\text{at that timestamp}}$$

$$\Delta FP_{DELETE} = \frac{\text{Number of Functional points Deleted / Total Number of Initial Function Points}}{\text{at that timestamp}}$$

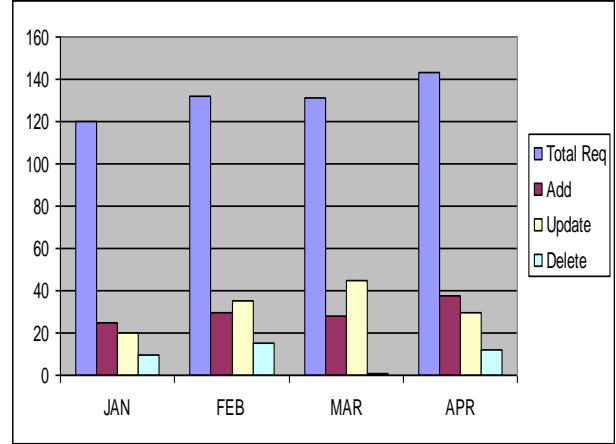


Fig1. Total Number of Functional Requirements and Changes

The Total Number of Function Points Changes during the particular timestamp is given by

$$\Delta FP_{CHANGE} = \Delta FP_{ADD} + \Delta FP_{UPDATE} + \Delta FP_{DELETE}$$

The Cumulative Number of Function Point Changes during the SDLC is given by

$$CFP_{CHANGE} = \sum_{i=1}^{i=n} \Delta FP_{CHANGE}^i$$

Hence the Requirements Stability Index (RSI) based on Function Point Changes is given by

$$RSI \text{ based on } \Delta FP_{CHANGE} = (FP + CFP_{CHANGE}) / FP$$

Where FP - denotes the total number of Initial Function Points

CFP_{CHANGE} - denotes the Cumulative number of Function point changes during SDLC.

The following figure1 depicts the Total Number of Requirement changes during the SDLC. Here, the X-axis values represent the timestamp (month) and Y-axis value represents the total number of functional requirements and functionalities Add, update, delete in that particular timestamp (month)

The following figure 2 depicts the Requirements Stability Index based on the Function Point Modeling. Initially the Requirements Stability Index value will be one as the value of cumulative change will be zero initially. This is represented as “RSI” in figure 2. When changes come into the project, the Requirements Stability Index value will be changed. This is represented as “Change” in figure 2.

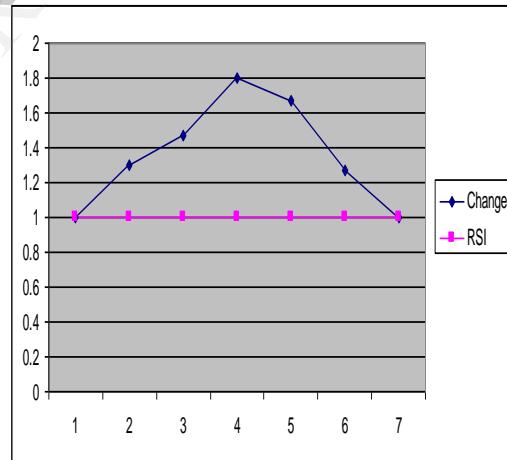


Fig2. Requirements Stability Index based on FP

V. Regression Analysis Based Model

Regression Analysis is a simple method which can be used for analyzing the functional relationships between variables. After performing an analysis, the regression results can be used to predict the dependent variable when the independent variable is known. The regression equation is of the below form

$$Y = \alpha + \beta X$$

Where X denotes the time sequence τ_i (Independent variable) and Y denotes Cumulative Function Point Changes ΔCFP_{CHANGE} (dependent variable). The Cumulative Function Point changes compose a sequence as,

$$\{\Delta CFP_{CHANGE 1}, \Delta CFP_{CHANGE 2}, \dots, \Delta CFP_{CHANGE n}\}.$$

The Variable β can be computed from the below equation.

$$\beta = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sum (X - \bar{X})^2}$$

The Variable α can be computed as

$$\alpha = \bar{Y} - \beta \bar{X}$$

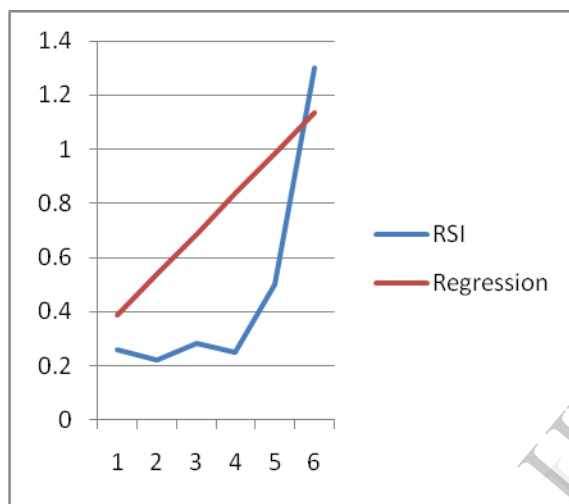


Fig3. Sample Regression Analysis Chart

Based on the cumulative function point changes, regression analysis have been done for the sample data as shown in Figure 3. For computing the Regression line, timestamp can be taken as X axis and Cumulative Function Point Change at that timestamp can be taken as Y axis. The Requirement Stability Index line can be computed based on figure 2. The Regression line and the Requirements Stability Index values can be compared to find out if the requirements are in stable state.

VI. Conclusion and Future Work

This paper concludes the importance of measuring the requirements changes for the lack of instability in the Requirements. The prediction model for Requirements Stability approach provides the solution for measuring the requirement changes based on the Function Point Analysis. This paper analyzes that Function Point Modeling for Requirement Stability will measure only Functionalities of the requirement and some of the non-functional factors such as reliability, scalability and performance based on the 14 general characteristics.

Few Non-Functional Requirements have not been measured for Requirement Stability. In future, the research can be done to develop a prediction model for measuring both functional and non-functional requirements for calculating the Requirement Stability and do the regression analysis for requirements stability.

REFERENCES

- [1] Carlos Monsalve,, Alain Abran and Alain Apri, "Measuring Software Functional Size From Business Process Models", International Journal of Software Engineering and Knowledge Engineering, Volume 21, Number 3, 2011.
- [2] Cheng Zhong, Qiu-suo Hu, Feng Yang, Meng-xiao Yin "Software Quality Prediction Method with Hybrid Applying PrincipalComponents Analysis and Wavelet Neural Network and GeneticAlgorithm", International Journal of Digital Content Technology and its Applications, Volume 5, Number 3, March 2011
- [3] Jiang Guo "Towards Automatic Analysis of Software Requirements Stability", Advances in Information Sciences and Service Sciences, Volume 2, Number 4, December 2010
- [4] Lukasz Radlinski and Wladyslaw Hoffmann, "On Predicting Software Development Effort using Machine Learning Techniques and Local Data", International Journal of Software Engineering and Computing, Volume 2, Number 2, December 2010.
- [5] Dr. Linda Rosenburg, Ted Hammer, Jack Shaw, "Software Metrics and Reliability", ISSRE, November 1998.
- [6] Massimo Felici, "Requirements Evolution Understanding Formally Software Engineering Processes within Industrial Contexts" Research project funded by Italian National Research Council, Feb 2002.
- [7] Roger S. Pressman "Software Engineering: A Practitioner's Approach", Fifth Edition, Chapter 11, pg.271-295, McGraw Hill, 2001
- [8] <http://www.psmc.com/members/default.asp>
- [9] <http://www.qpmg.com/fp-intro.htm>
- [10] Syed Ali Abbas and SaleemUllahLar, Xiaofeng Liao, Raja AftabNaseem, "Software Models, Extensions and Independent Models in Cocomo Suite: A Review" Journal of Emerging Trends in Computing and Information Sciences, Volume 3, Number 5, May 2012
- [11] Susan Ferreira, James Collofello, Dan Shunk, Gerald Mackulak, "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation", The Journal of Systems and Software, March 2009.

Bibliography



D. Francis Xavier Christopher received his B.Sc., in 1996, M.Sc., in 1998 from Bharathiar University, Coimbatore. He obtained his M.Phil, in the area of Networking from Bharathiar University, Coimbatore in 2002. At present he is working as a Director, School of Computer Studies in RVS College of Arts and Science, Coimbatore. His research interest lies in the area of Software Engineering.



Dr. E. Chandra received her B.Sc., from Bharathiar University, Coimbatore in 1992 and received M.Sc., from Avinashilingam University, Coimbatore in 1994. She obtained her M.Phil., in the area of Neural Networks from Bharathiar University, in 1999. She obtained her PhD degree in the area of Speech recognition system from Alagappa University Karikudi in 2007. At present she is working as a Director at School of Computer Studies in SNS Rajalakshmi College of Arts & Science, Coimbatore. She has published more than 20 research papers in National, International journals and conferences. She has guided for more than 30 M.Phil., research scholars. At present guiding 5 Ph.D research scholars. Her research interest lies in the area of Data Mining, Artificial intelligence, neural networks, speech recognition systems and fuzzy logic. She is an active member of CSI, Currently management committee member of CSI, Life member of Society of Statistics and Computer Applications.