Integrated Intelligent Research (IIR)

International Journal of Computing Algorithm
Vol 02, Issue 01, June 2013, Page No.16-18
ISSN: 2278-2397

Comparison of Statistical Calculations for Software Model Analysis

N.Sasikala[1], G.Rasitha Banu[2], K.Rangarajan[3]
[1, 2] Research Scholar, Mother Theresa Women's University
[3] Hod, MCA Department, Bharath University
Email: ngskala@gmail.com, rashidabanu76@gmail.com

Abstract- Software engineering is the latest and most complicated discipline of engineering. Current software engineering is based on empirical practices, while theoretical research and investigation into foundations of software engineering have long been left behind. Theoretical software engineering studies the nature of software, mathematical models of software architecture, mechanism of software behaviors, methodologies of large scale software development and laws behind software behaviors and software engineering practices. The main part of software development is using a software models to find a systematic and disciplined way of the solutions for a given project. This paper focuses on fundamental theory of different lifecycle models with the cross comparison of statistical calculation for software model analysis with emphasis on the dependency of activities on phases. We used a mechanism for finding the dependency matrices between the different phases of different models. In this paper we devised a methodology for estimating the occurrence relationship between the activities with its correlation and regression. Finally we have given the comparison of statistical calculation for dependency values among different models.

Keywords: Association matrix, activity relations, association map construction, occurrence relationship, statistical comparison.

I. INTRODUCTION

Software is a special type of behavioural information of computing and a means of interaction between the abstract world and the physical world. The nature of software makes software engineering a unique discipline, which is innately the most complicated engineering branch that human ever experienced, and inherently the most overarching trans-disciplinary field in both theory and applications. The software development model approach attempts to provide a set of guidelines for the design and implementation of software at system and module levels. However, this approach has been focused on technical aspects of software development lifecycles.

1. Software Development Process

A software development process is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Each model contains several phases of activities like the ones given below.

2. Requirements analysis

The most important task in creating a software product is gathering requirements or requirements analysis.

3. Specification

Specifications are most important for external interfaces that must remain stable.

4. Architecture

The architecture of a software system or software architecture refers to an abstract representation of that system.

5. Design, implementation and testing

Implementation is the part of the process where software engineers actually program the code for the project and run or execute the project

6. Deployment and maintenance

Deployment starts after the code is appropriately tested, is approved for release and sold or otherwise distributed into a production environment. Maintenance and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software.

7. Software Design models

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view.

The software design models can be classified as follows:
Water-fall model, Spiral model, Iterative model, RAD model, V-model and Prototype model

II. WEIGHTAGE ASSUMPTIONS

In the field of software development, the development cost is the cost incurred during Requirement analysis, development, coding & testing etc. So we can determine the effort distribution for different phases of the software development. Let us have the effort distribution for different phases as Requirement 10%, design 20%, coding 20%, Implementation & testing 50%.This is called the weightage assumption for different phases of a software model.

For example let us have the effort distribution for V- model.

S. No	Phases	No.of activities	Weight age	Weight per Activity
01	Reguirement Analysis	4	10%	0.0059
02	System Design	3	20%	0.0118
03	Architecture Design	4	20%	0.0118
04	Model Design	3	20%	0.0118
05	Coding & Testing	3	30%	0.0176

Table 1: Comparative of various software design model

Name of models	Requirement specification	Design	Construction
1.Water fall model	1.All possible requirements of the system to be developed are captured in this phase. 2.Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.	1.The requirement specifications from first phase are studied in this phase and system design is prepared. 2. The system design specifications is created to serve as input for the next phase of the model.	1.The work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality;
2.Spiral model	1.The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.	1.A preliminary design is created for the new system.	1.A first prototype of the new system is onstructed from the preliminary design. This is usually a scaled-down system and represents an approximation of the characteristics of the final product. 2.A second prototype is evolved by a fourfold procedure: (1) evaluating the first prototype in terms of its strengths, weaknesses, and risks; (2) defining the requirements of the second prototype; (3) planning and designing the second prototype;(4) constructing and testing the second prototype.
3. Iterative model	The requirements for the software are gathered and analyzed. Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.	a software solution to meet the requirements is designed. This may be a new design, or an extension of an earlier design.	---------------
4.RAD model	Business Modeling: The information flow among business functions is defined by answering questions like what information drives the business process, what information is generated, who generates it, where does the information go, who process it and so on.	Data Modeling: The information collected from business modeling is refined into a set of data objects(entities) that are needed to support the business. The attributes (character of each entity) are dentified and the relation between these data objects (entities) is defined.	Process Modeling: The data object defined in the data modeling phase are transformed to achieve the information flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting or retrieving a data object.
5. V-model	1.the requirements of the proposed system are collected by analyzing the needs of the user(s). 2.The user acceptance tests are designed in this phase.	1.System engineers analyze and understand the business of the proposed system by studying the user requirements document. 2.They figure out possibilities and techniques by which the user requirements can be implemented.	1.The designed system is broken up in to smaller units or modules and each of them is explained so that the programmer can start coding directly. 2. The unit test design is developed in this stage.
6.Prototype model	Identify prototype	Agree to a plan	Create the prototype

Construction of Activity Association Matrix

If a phase consists of set of activities P1A, then P1A contains of set of activities which are determine by the phase as per the model.$P_1A = \{ P_1a_1, P_1a_2, P_1a_3 \ldots, P_1a_n \}$

Likewise second phase of activities denoted as P2A. It contains the activities which are specified in the second phase of the software development model.

$P_2A = \{ P_2a_1, P_2a_2, \ldots, P_2a_n \}$All the activities of each phase to be defined as a set.

PxA where X denotes the phase of the model .

First Level Association Matrix :

The first level association matrix represents the relationship impacts between phase x with
phase x+1.

Nth level association matrix

The nth level association matrix represents the relationship impacts between phase x with x+n. Let us assume that phase value of x is 1. the first phase of the water falls model is feasibility study. The second x+1 phase of the water falls model is requirement analysis.

III. INDEPENDENT ACTIVITY ASSOCIATION MATRIX

The Requirement analysis set (P1A) considered as column and System Design set (P2A) considered as row then a two dimensional association matrix framed with the following condition.

Condition 1:The activities of Requirement analysis set (P1A) is created an impact with activities of System Design set (P2A) then the value is set as ‘1’. these activities are considered as related activities/dependant activities .

Condition 2:The activities of Requirement analysis set (P1A) is not create on impact on the activities of SystemDesignset(P2A) then the value is set as ‘0’. These activities are considered as Isolated activities/independent activities

Condition 3: If the activities of Requirement analysis set (P1A) partially created the impact on System Design set (P2A) then those activities to be called as partial dependant activities. If its impact is dominant on the process then the value to be considered as approximate equivalent to the dependant activity else it is treated as an isolated activities. In certain cases these partial activities are treated as X don't care condition.

requirement analysis\ system design	system Analysis (p2va1)	system design (p2va2)	system test design (p2va3)
Requirement gathering(p_1va_1)	1	0	0
Requirement analysis(p1va2)	0	1	0
Software requirement(p1va3)	0	1	1
Requirement acceptance testing(p1va4)	0	0	1

Cyclic Avoidance:
The activities relationships are determined only for the phase x with the next phase x+1. There is no determination of activities within the phase itself. So there is no cyclic path for the relationship determination.

Multi level Relationship:
If there is a relation between the phase x with phase x+2 through the phase x+1, then a multi level relationships will occur.

Feed Forwarded approach:
The activities relationships can be determined only for the phases x, x+1,x+2 etc. as a forward approach. Here there is no determination of relationships for the x+1 phase with its previous phase x.

IV. ACTIVITY WEIGHTAGE CALCULATION

(a) Individual phase weightage: The individual phase weightage can be ascertained using the nuactivities in each phase , total no.of activity and percentage of weight of each phase.
IPW (individual phase weightage) = ((Number of activity in phase)/(total activities of model))* (percentage of weight of each phase)
(b) Individual activity weightage:
Weight age per activity = Individual phase weightage / number of activities in that phase
iaw = ipw / 3

V. CONSTRUCTI ON RELATIONSHIP WEIGHTED MATRIX

In this step I am constructing the relationship weightage matrices between the phases x and x+1.

Association map construction:

(a)Direct relationship function

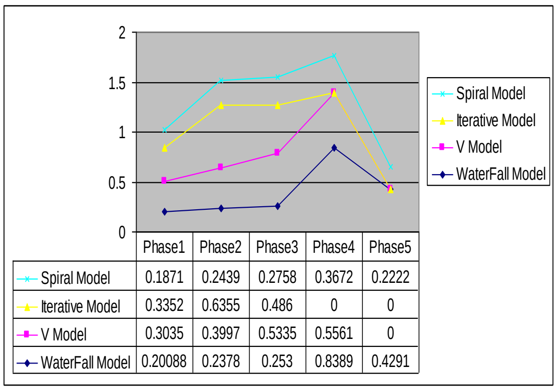
Association map is constructed for the phases x and x +1 if there is a direct relationship between them. If the phase x is having relation with x+1 then there is a path existing between the x and x+1 phase in the association map.

(b)Routed relationship function
(i) Single hidden phase

This type of routed relationship is constructed if there is a path between the phases x and the phase x+2 through the phase x+1 which is in between those two phases

(ii) Multilevel hidden phase

This type of relationship occurs only when there is a relationship between x and x+3 or x+4 phase which is having the path existence through x+1 and x+2 etc. The dependency values are calculated for all the models among different phases and minimum dependency of each model with each phase was selected and given below in the line chart:



	Phase1	Phase2	Phase3	Phase4	Phase5
Spiral Model	0.1871	0.2439	0.2758	0.3672	0.2222
Iterative Model	0.3352	0.6355	0.486	0	0
V Model	0.3035	0.3997	0.5335	0.5561	0
WaterFall Model	0.20088	0.2378	0.253	0.8389	0.4291

VI. CONCLUSION

With reference to the study of different models it can be concluded that there are several advantages and limitations in each software design models. These issues affect the solutions of software design approach. To overcome the above specified issues the software development model to be well defined. The sophisticated architecture can be achieved through practices. But the impact and the efficiencies are measurable through the attributes. The efficiency can be determined through the statistical tools. With the above comparison analysis chart, it has been derived that spiral model has less dependency on other phases and activities than the other models.

REFERENCES

[1] Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process, Addison-Wesley, 1999.
[2] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, 1991.
[3] M. Aksit and L. Bergmans, Obstacles in Object-Oriented Software Development, Proceedings OOPSLA '92, ACM SIGPPLAN Notices, Vol. 27, No. 10, pp. 341-358, October 1992.
[4] W. Tracz, DSSA (Domain-Specific Software Architecture) Pedagogical Example, In ACM SIGSOFT Software Engineering Notes, vol. 20, no. 4, July 1995.
[5] Tekinerdogan, Syrnthesis Based Software Architecture Design, Ph.D. thesis, University of Twente, The Netherlands, March 2000.
[6] B. Tekinerdogan and M. Aksit, Classifying and Evaluating Architecture Design Methods, to be p ublished in Software Architectures and Component Technology: The State of the Art in Research and Practice, M. Aksit (Ed.), Kluwer Academic Publishers,2000.