A TECHNIQUE FOR REQUIREMENT VALIDATION USING FUNCTION POINTS

R. Afzaal*
Department of Computer Science and Engineering, UET Lahore*, **, Pakistan
ram3109@gmail.com

ABSTRACT: *Requirement gathering is a very substantial and essential natural process in software development. Early validation of the requirement for software development was used to save effort, budget, and time. To validate the requirements by defining metrics was the main focal point of the research paper. Metrics have been proposed by using the function points. The final results are promising as the study shows that if the requirements are validated through metrics, then ambiguous requirements can be thrashed away. Role of Function Points Technique in metrics development has more trustworthy results for requirement validation. The research can be passed out for the execution in some actual applications*.

Keywords: Requirement validation, requirement validation technique, function point metrics

INTRODUCTION

SDLC (Software Development Life Cycle) is an abstract model used in project management that categorizes the methods that are extended in an information system development plan from an initial prospect study through the keep of the completed application/software. Numerous SDLC approaches have been established to run the operations involved, renting in the waterfall model (the original SDLC method), rapid application development (RAD), joint application development (JAD), the fountain model, and the spiral model. Commonly, numerous models are combined into some sort of hybrid approach [1].

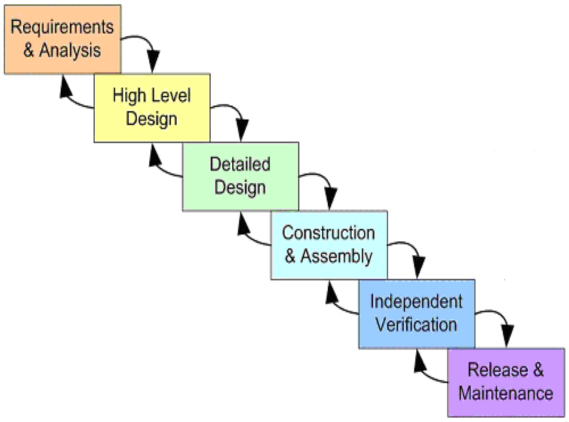

Figure: 1. Life cycle

The requirements & analysis stage emphasizes what the system do in an effort that views all stakeholders, including supporters and potential users, as substantial bases of data. In the course of this requirement stage first, create one explicit set of requirements that authenticates an understanding between all interested parties on what the system should do and also offer developers and all other interested parties with a clear conclusion of the requirements. Conclude the limits of the system. Select features to offer a home for possible iterations [2]

Validation (& verification) process of controlling whether the requirements, as identified, do not contradict the expectations about the placement of various stakeholders, and do not contradict with each other. It is Requirements Quality Control.

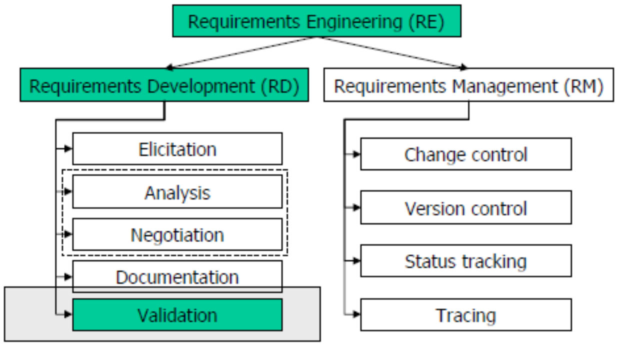

Figure: 2. The layout of Requirement Engineering

Two key assumptions that frame traditional RE:

1. Requirements always are existent 'out there' in the minds of concerned parties, and they simply need to be elicited through various contrivances.
2. The key concerned party operates in a state of goal correspondence, in which there is widespread agreement on the general goals of the system under development.

Under these conventions, validation of requirements is nothing more than checking the analysts have understood and agreed with the stakeholders' intention appropriately and have not introduced any miscalculations when writing the spec.

In software development, a metric is the measurement of a specific feature of a program's procedure or efficiency. Correspondingly, in network routing, a metric is a measure used in calculating the next host to route a packet to. A metric is sometimes employed directly and sometimes as an element in an algorithm. In software design, a benchmark includes metrics.

The requirements phase categorizes the functionality, performance steps, and other features that the product must satisfy in order for it to be acceptable to the consumer. So those Softwares that execute well is useless if it at the end of the day fails to meet business user needs. But requirements faults are the number one cause of software project failures, many organizations or consumers go on to convey requirements that are really vague, ambiguous or can say incomplete.

The problem is that organizations or consumers be unsuccessful to apply the same rigorous testing processes to requirements development as they come along with software development. This is an error, as the cost of identifying acute errors during requirements development is exponentially less than the errors that are uncovered after coding starts.

Miki Ikoma *et al*.,[2], in their research titled "Using a Validation Model to Measure the Agility of Software" paper shows that, combining the V&V model and the agile model can establish how quickly planned software items can be corroborated. This paper focuses on the validation because only the validation can confirm the quality of the final software product. Validation is more important than verification, which can be performed by tools or during intermediate processes.

Julio Cesar do Pardo Leite, *et. al*., [7], describes in their research paper entitled "Requirement Validation through Viewpoint Resolution." Suggest a specific technique viewpoint resolution as a way of offering validation of the essentials for a complex system, in this research paper basically shows that such a technique can indeed rigor and assistance to what is at present an almost completely informal part of the required validations.

Further findings of Julio Cesar do Pardo Leite, *et. al.*, [7], help others to prove that it is extremely desirable to have early validation in the software construction process. Its strategy for conflict detection between views is supported by a running program, the static analyzer.Uzair Akbar, Raja, [5], in his research entitled "Empirical Studies of Requirements Validation Techniques" in this paper, he provides an overview of requirement validation techniques

Kristian J. Cruickshank, *et al.,* [6], in their research paper titled "A validation Metrics Framework for Safety-Critical Software-Intensive System." Define that validation of the requirements has become exceedingly more difficult. Then they offer a new technique which is Safety-critical software-intensive systems.

Refer to Christopher Fuhrman, [4], a paper titled "Software Verification and Validation within the (Rational) Unified Process". In this paper firstly we compare these two validation and verification process frameworks and secondly to establish whether that is directly borne out, partially supported, or not tolerated by the Unified Process..

The next Section explains the proposed solutions by applying metrics. The section after that will corroborate the outcomes. The last segment will conclude the solution and consequences by giving future recommendations.

MATERIALS AND METHODS

Propose Technique for requirement validation

Requirement validation is a construction unit in software evolution. The requirement should be very light and consistent so that the quality of the software should be authentic. In this view, we have introduced a new metric.

The measured impact on the quality of the software. Because we want to evaluate the importance, validity of information so that it relays and update our knowledge according to it.

Our new metric is the Function points for this. Function point counts that are affiliated with either projects or applications. A function point is a unit of software measure. Function points are reckoned withm all the stages of a project from requirement up to execution.

The metric calculation is as follows:

Two cases of function level as follows:

1: Transactional function point (EI, EO, EQ)

EI (The data may attain from a data input screen or another application)

EO (A fundamental process in which derived data licenses across the limit from inside to outside.)

EQ (An exterior inquiry not having calculated values or derived data. This distinctive discriminates an exterior inquiry from an external output.)

2: Data function point (ILF, EIF)

ILF (A user distinguishable group of logically associated data or control facts and figures)

EIF (Data referenced by SW application, but proclaimed by another SW application**)**

Steps for function point:

1: Select any one of the types of count

Development (All function impacted by the project activities)

Enhancement (All functions being added, altered, and removed, but boundary remains same)

Application

2: Found boundary to establish its choice.

3: Identify & rate transactional function point

EI count (DETs & FTRs), EO count (DETs & FTRs) , EQ count (DETs & FTRs)

DETs (**Data Element Type**) All GUI control are DETs or say user view things are DETs

FTRs (**File Type, Referenced***)* A FTR are a file form that referenced by a operation/transaction.

4: Identify & rate data function point

ILF count (DETs & RETs), EIF count (DETs & RETs)

DETs (**Data Element Type)** All GUI control are DETs or say user view things are DETs.

RETs (**Record Element Type***) A* RET is a consumer identifiable sub-group of data essentials within an ILF or an EIF.

5: Determine the value adjusted factor

VAF = 0.65 + (0.01 * Total general characteristics)

6: Determine adjusted function point

AFP= Value adjusted function * Undefined function point.

RESULTS AND DISCUSSION

Steps for performing function point:

1: Type of count Growth Enhancement

Application

2: Establish boundary

3: Identify & rate transactional function point

EI count (DETs & FTRs), EO count (DETs & FTRs), EQ count (DETs & FTRs)

4: Recognize & rate data function point

ILF count (DETs & RETs), EIF count (DETs & RETs)

5: Determine the value adjusted factor

VAF = 0.65 + (0.01 * Total GSCs)

6: Determine adjusted function point

AFP= VAF * UFP

CONCLUSION AND FUTURE DIRECTION: Requirement validation is a research question which we set about in this area. We proposed the metric like Requirement validates through function points. Metrics validation has been done using mathematical expressions. Results have proven our research is valid for the validation of requirements. System developers are the major benefits of this research. This research in the future be helpful or extended by taking a real-life application and applying these metrics for requirement validation in that application. The resolutions and maybe then elaborated and maybe just about new metrics have to fix.

REFERENCES

1. David Cohen, Gary Larson and Bill Ware "Improving Software Investment through Requirements Validation ", 2009

2.. Miki Ikoma, Takahiro Tanida, Masayuki Ooshima , Michiko Oba, Sashiro Sakai "Using a Validation Model to Measure the Agility of Software Development in a Large Software Development Organization" IEEE 2009.

3. Alessandro Cimatti, Angelo Susi, Marco Roveri, and Stefano Tonetta "From Informal Requirements to Property-Driven Formal Validation", Springer Verlag Berlin Heidelberg 2009.

4. Christopher Fuhrman, Fatime Djlive and Edgar do Palza "Software Verification and Validation within the (Rational) Unified Process", IEEE 2003

5. Uzair Akbar Raja "Empirical Studies of Requirements Validation Techniques" 2002.

6. Kristian J. Cruickshank and James Bert Michael, Man-Tak Shing "A validation Metrics Framework for Safety-Critical Software-Intensive System." IEEE 2009.

7. Julio Cesar do Pardo Leite and Peter A. freeman "Requirement Validation through Viewpoint Resolution."

8. James Bert Michael and Man-Tak Shing "Metrics Framework for Safety-Critical Software-Intensive System." 2009