International Journal of Engineering & Technology, 7 (2.26) (2018) 7-12

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Determination of influencing metricsin water fall model

K. David [1] *, R. Anand [2]

[1] Assistant Professor,H.H The Rajahs College (Autonomous),Pudukkottai, Tamilnadu, India
[2] Research Scholar, Bharathiar University, Coimbatore, Tamilnadu, India.
**Corresponding author E-mail:*

Abstract

You cannot control what you can't measure - Tom DeMarco -Metrics plays a critical role in any industry. The metrics can tell the senior management the health of the activity in that organization. In the paper, we have studied the relationship between the metrics and the quality of deliverables. Once the right metrics are identified for a particular phasein the life cycle model it will help the project manager to proactive-ly detect the issues earlier and arrest the defects. This can help the project teams to deliver the product on time without defects. There by the cost of the projects can be reduced which in turn will have high customer satisfaction.

Keywords: *QOD; SDLC; ANOVA; Confidence Interval*

1. Introduction

Measurement is a key element of an effective and efficient software development process as it evaluates the quality and effectiveness of the process. Measurement is a tool through which the management identifies important issues, events andtrends that enables them to make informed decisions. Moreover, measurements help inpredicting outcomes and evaluation of risks, which in turn decreases the probability ofunanticipated surprises in different processesthat are employed in due course of developments.

Waterfall model is the mostly used life cycle model in big software projects. This works well if the client does not change the scope of the project. The strict dead line driven methodology plays well for the projects that have a clear vision of the result. The core aspect of the waterfall methodology is the detailed plan that is scheduled and executed sequentially to a final resolution. Waterfall approach is one of the heavy weight methodologies and this follows the model of requirement-design-build. Here the processes are standard and well defined; also, it has well-structuredaspects through definite phases. Every phase has a specific activity and the deliverables has tobe completed before commencing to the next phase. In this model, developers can accurately cite expected flaws and outcomes in advance.

Advantage of the waterfall model is thatit can be easily managed and the cost can be determined at the early stage. Developments can happen with the time aimed at improving the waterfall methodology. The evolution of this model has helped to bring the optimization in the waterfall method, enable developers to understand the requirements of the users better through prototyping of software, reducing market time and pressure that can improve the quality and the reusability can also be increased. Also in case of the employee turnover, waterfall strong documentation allows for minimal project impact.

2. Review of literature

There have been consistent steps taken in various process monitoring and controlling features to identify different metrics that should be used in various life cycle methods. G. Canfora a, F. Garcı´a b, M. Piattini b, F. Ruiz b and C.A. Visaggio have focused on the process modelling and introduced a set of metric for the process model that can be used as an maintainability indicator. They have proposed and empirically validated a set of representative metrics to evaluate the maintainability of descriptive software process models and these metrics are based on the main elements included in a software process models and can be used to ease the process evolution [1]. HaneenHijazi, ThairKhdour, AbdulsalamAlarabeyyat have reviewed about the software development process models and in detail have investigated the state of risk management in each of these models. They have found that development methodologies risk management and have analysed the major source of risk in waterfall model, Incremental Model, V-Model, Spiral model, Agile Development Model.Also they have specified that continuous requirement change, no overlapping between stages, poor quality assurance, relatively long stages are the major source of risk factors involved [2].P.Mahizharuvi, Dr.K.Alagarsamy has tried a novel security mechanism in system development life cycle. Authors have started their initiation process of their research and have given some suggestion to enhance the security mechanism to improve the system development life cycle [3]. Youssef Bassil has proposed a simulation model for the Waterfall development process using the Simphony.NET simulation tool. This model is used to simulate different phases of the waterfall model including software solutions to be developed, operational resources, employees, tasks, and phases. It can also assist project managers in determining the optimal number of resources required to produce a particular project within the allotted schedule and budget [4]. DeepshikhaJamwahas provided the opinion of software developers on variousaspects of developing a software development model suchas: development time, Project complexity, Implementationchallenges, Extensive and accurate documentation for small software organization [5].MihaiLiviuDespahas considered heavyweight and lightweight methodologies. It is suggested that heavyweight methodologies suits well for the projects where the requirements does not change frequently. This methodology is easy for tracking, evaluation; reporting and can showcase tangible deliverables in every stage of the project. Whereas lightweight methodologies suits well for projects in which the requirement is not clear. Despite of this the project owner is involved in all the phases of the project. It is suggested that while choosing a software development methodology project owner profile, developer's technical expertise, project complexity, budget and deadlines must be taken into account [6]. Rajesh H. Kulkarni, P.Padmanabham, K.K.Baseer have extended the proposals of integrating software engineering processes with human computer integration for product development to the waterfall model. They have attempted to optimize the software development process, which has found utmost importance in the development phase [7]. MaaritTihinen and Päivi, Rob Kommeren, Jim Rotherham has focussed on describing set of metrics in global software developmentused in the industrial practises. They have suggested that analysing of metric can be used to proactivelyact to the potential issues in the projects. Metric such as effort size cost, productivity will act an indicator [8].Unnati A. Patel, Niky K. Jain has proposed modification to the existing waterfall model. They have suggested that at every phase of the waterfall model there is a need to check with the customer with respect to their requirement. Since frequent change in the requirement of the client is the setback in this model, they have given provision to avoid this issue [9]. PoojaJha, K S Patnaikhas studied the empirical study and the importance of metrics in an organization. They have examined the impact of lower specification limits and upper specification limitson various projects and have proposed metric such as defect efforts, defect density size, defect removal efficiency, cost of quality for capability analysis metrics.In addition, these metrics can help the on-going software project [10].Mubarak Elamin and ElmubarakDaleel have considered the features such as requirement specifications,understanding Requirement,cost,simplicity, risk analysis, flexibility to change, user involvement duration, client satisfaction, resource control to analyse different life cycle models. In addition, one has to consider common factors along with risk and cost aspects to select a suitable life model [11].Adetokunbo A.A. Adenowo, Basirat A. Adenowohas suggested about waterfall model and object oriented approach software engineering methodologies for the system development. Waterfall model is simple to use but uses linear approach and is suitable for sequential and procedural design but would be difficult to rectify error at the later stage. Object oriented approaches can be independently operated and is easy to incorporate changes at the later stage. Authors conclude that both the approached are currently functional but more importance is given to the object oriented approach [12].

3. Measurement and metrics

Measurement is a simple representation of information used to establish a common understanding of status condition or, where we are in particular activity or task when executed. While the metrics provides the measuring units to depict values and thresholds A metric is a quantitative measure of the degree to which a system, component or process possesses a given attribute. In the words of Vince Law, "A metric will tell you that something is happening, while an analysis will tell you why something is happening."Any organization will have the objectives well defined. Therefore, said objectives has be measured and achieved through the metrics, also that needs has be employed. In other words, the problem defined by the management or the customer needs has to be mapped to statistical models using the metrics to have well defined and predictable solutions. Fig-1 will help us to understand the mapping of objectivise from their previous experiences to where they want to be.

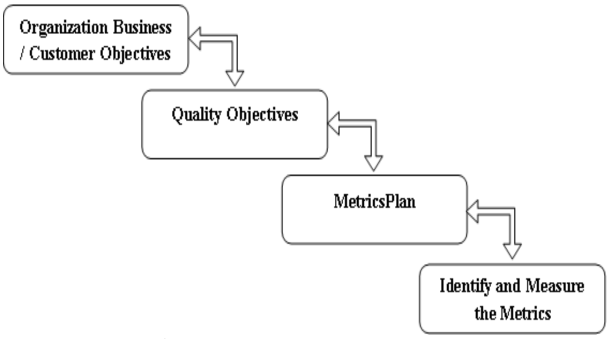

Organization Business / Customer Objectives
Quality Objectives
MetricsPlan
Identify and Measure the Metrics

Fig. 1: Mapping of Organization Business Objectives to Improvement.

The Table-1 shows the metrics classification in three broad categories that will ultimately lead the product.

Table 1: Metrics Classifications

Metrics		
People	Product	Process

People Metrics- it is a metric used to measure the experience, skill gap and domain. Some of the metrics are Staff Turnover percentage, Average Domain Experience, Average Technical Experience, Average Domain and Technical Experience of reviewer, Technical Skill Gap, Average experience in the project, Number of fresher's, Number of Project leads.

Process Metrics - process metric is a metric used to measure characteristics of the methods, Techniques, and tools employed in developing, implementing, and maintaining the software system. Some of the metrics related to the process are Requirement Study/Analysis effort, Project Planning effort, Project Plan Review effort, Test Development effort, Test Execution effort, Total Number of Test Cases, Cost of Poor Quality (Rework effort), Effort variation, Schedule variation, Product Metrics – A product metric is a metric used to measure the characteristics of the documentation and code. some of the product metrics are Number of Initial Requirements, Number of Requirements Newly Added, Requirement stability Index, Environment availability at offshore, Usage of IDE for test cases, Overall Injected Defect Density, Defect Validity Index.

4. Objective of the study

The objective of the study is to identify the metric that may impact the first time right and quality of deliverables on the life cycle of the project. We may choose the classical life cycle or testing life cycle or agile are the most common form of life cycle that are used in the industry. The other life cycles such as prototyping or RAD or SPIRAL are the models that are rarely being used in software development life cycle. The main objectives of the study is

1) To Study the impact of metrics for the quality of deliverables(QOD) of the project
2) Understanding of the current metric and propose suitable metrics for the future improvement

The Fig-2 show the diagrammatic representation about the metrics that needs to be identified, understand each of them and make improvements in the engineering process for knowing the predictable outcomes in the project.

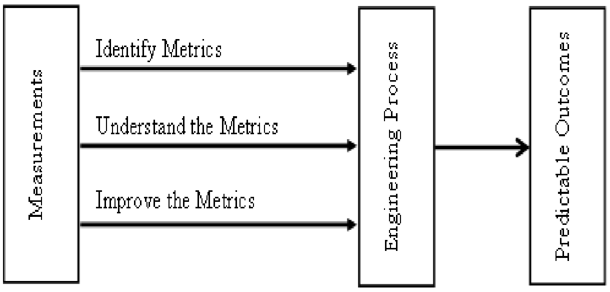


Fig. 2: Diagrammatic Representation of Metrics That Impacting the Outcomes

Some of the benefits of measuring theprojects using the appropriate metrics are as follows.

- Evaluating risks.
- Benchmarking.
- Improving planning.
- Improving effectiveness.
- Evaluating and improving product quality.
- Measuring productivity.
- Determining level of customer involvement and satisfaction.
- Comparing processes and products with those both inside and outside the organization

5. Life cycle and metrics

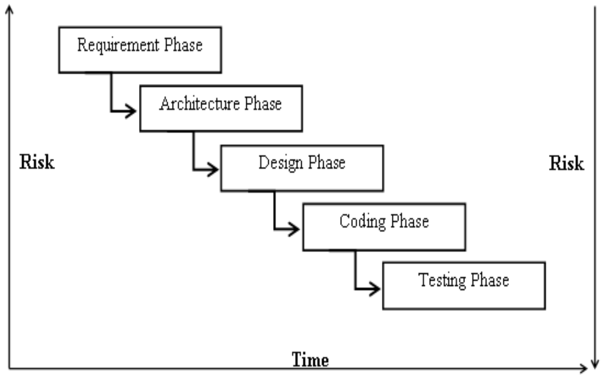


Fig. 3: Phases of the Waterfall Development Model.

The Fig-3 depicts the different phase of the waterfall model that includes the main phases such as requirement phase, Architecture phase, Design phase, Coding phase and the Testing phase. The DRE is the main metrics to be analysed at every phase and the general formulae for the DRE is

Defect removal efficiency (DRE) = (A/ (A+B))*100

A-Pre delivery defects (No of internal review defects + No of internal test execution defects (excluding product defects, but inclusive of defects of script, test case)).
B-Post Delivery Defects (No of customer reported defects)
The purpose of the DRE is to compute the effectiveness of internal defect identification and resolution processes.The Fig-4depicts the phases of the waterfall model and this diagram emphasises the need to avoid defect moving to the next phase and in order to achieve this it is important that at every phase defect prevention, defect detection, defect removal needs to be applied so that defect do not creep to the next level.

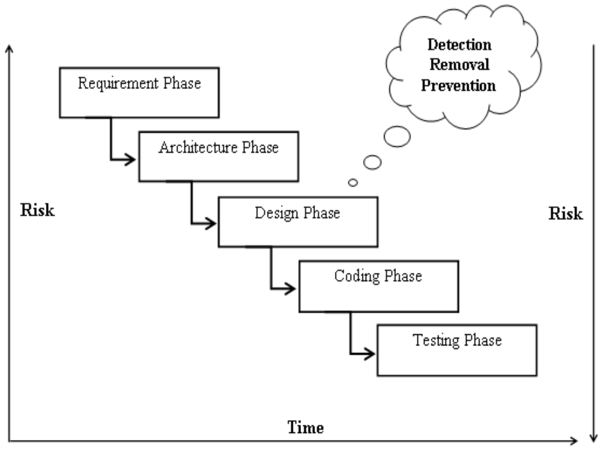


Fig. 4: Phases in the Waterfall Model for Preventing, Detection and Removal of Defects.

Identifying the suitable life cycle model is the most important aspect before starting a project in any organization. In information technology, Waterfall model ispredictive methodology thatmaybe used in most of the software projects. This model helps to deliver the project on time without the overrun cost or additional cost. Herecontinues deliverables are provided to the customer as per the planning documentright on time. In waterfall model, only one phase is active at a point of time because of this, complexity in the engineering processcan be reduced.It is extremely important to understand the need of developing software and in this regard, requirement analysis plays a vital role.It is important to capture all the requirement of the clients, walk through the same and check the feasibility of carrying it out, verify the requirement document with the state holders before moving to the next phase. Before preparing the document it is necessary to have questionnaires’, use cases and brainstorming session with the stakeholders of the project.

The requirement Phase should givea clear idea of what the developersare expected to do. In addition, when there is a change in the requirement,the project manager uses the change management to adjust the requirement. During the requirementphase, it is necessary to track the number of times the requirement has changed during this phase (i.e. Requirement Stability Index). Defect Removal Efficiency is the main metric that needs to be checked at this phase, this is because in waterfall model defect should not move to the next level and the immediate need is to remove the defect at the early stage. Normally DRE should be [1], which means that there is no defect. If the score of the DRE is less, which means that; we need to relook our process. Effective way of the Reviewing (Review Efficiency) can reduce the error. DRE for the Requirements phase is (number of requirements related defects identified during requirements inspection/total number of requirements defects identified within the solution).The requirement stability index is one of the main metric with respect to this phase and the formulae is: [Total number of initial requirements/ (Total number of initial requirements+ (No of requirements newly added + no of requirements modified + no of requirements deleted))]*10. The purpose of this metric is to compute the extent to which requirement for the product / feature remains stable.

Architecture Phase: The Architectural Phase provides the complete specification of the system architecture including hardware and software both in development and verification process.Control and data structure of the system architecture is carried out in this phase. This phase acts like a framework for the details given in the specification document. It gives clear detailsabout the external interfaces and the tool that a project would require. It describes the different layers of the software that are not dependent on the hardware.The structural quality aspect of the defect hasto be fixed at this stage. The software architecture is based on business and technical assumptions, abstracted level of information, attributes of the software that is set based on the customer expectations. The defect removal efficiency, Architecture defect density and review efficiency are the metrics that has to be addressed at this phase to avoid defects.

Design Phase: In this phase each component of the program isbased on the specification for theusage of what data structures, algorithm and the interfaces are required for components.In addition, this phase should give a clear picture of how each phase is implemented. If the design phase is effectively done with effective tools, developer will have high flexibility in writing the code.It is important for the design architect to satisfy the customer and the development team. Defect Removal Efficiency is the important metric that needs to be analysed at this phase the formulae for this is as follows: DRE Design = (number of design and requirement related defects identified during design inspection/total number of design and requirement defects identified within the solution). Design defect density in the design phase measures the overall injected defect density of the given project for a given project density. The formula for the design defect density is: (Number of Internal and Customer Reported Defects in the design phase / Actual Effort put in for the Design).Design Stability Index can be calculated as (Total number of original Design + Number of design changed till date+ Number of design added + Number of design deleted) / (total number of original design) and this can indicate the changes made by the client during this phase.

Coding Phase: This phase uses programming languages for instructing the system to perform an action. Design document acts as the input to this phase. Every aspects identified in the design document haveto code based on the module specification. Here the developer has to follow the coding standards and write minimum number of lines to perform the maximum functionality. Defect Removal Efficiency is important metric that is needed to track the defect committed by the developer. The size of the software project is estimated with Function Point (FP) and Lines of Code (LOC). LOC is used to predict the effort that is required to develop a program thatin turn will give us the programmer productivity.The amount of business functionality is expressed in terms of function points(FP). Functional user requirement of the software is identified with one of the five types and they are inputs, outputs, inquiries, internal files, and external interfaces. The developer does unit testing, DRE for the unit test is calculated as DRE . Unit Test = (number of unit test defects identified during unit test/total number of unit test defects identified within the solution).

Testing Phase: Testing is the one of the important phase where at most care has to be taken. This is because if errors are not detected at this phase the product reaches the customers with bugs. Unlike other phases,DRE is most important metric. If the DRE is high then we can be sure that the product has reached the customer with very low error rate. DRE Integration Test = (number of integration defects identified during integration test/(total number of integration test defects identified within the solution post-unit test)Review Efficiency = (Total Number of Review defects) / (Total number of Review defects + Total number of Testing defects)*100

Even though some of the project uses agile methodologies,it is to note that waterfall is used for deeper requirement gathering and design. The divide and conquer sis the approach followed in the waterfall model. Table -2 summarises the metrics of the various phases of the waterfall model.

Table 2: Metrics and Phases of the Waterfall Model

Requirements	Architecture	Design	Coding	Testing
Requirement Stability Index(RSI) Defect Removal Efficiency(DRE) Review Efficiency(RE) Requirement Defect Density (RDD)First Time Right (FTR)Requirements Cycle Time(RCT)	Architecture Stability Index(ASI) Size of the Architecture(SOA) Architecture Defect Density(ADD) Review Efficiency(RE) Defect Removal Efficiency(DRE) First Time Right (FTR)Architecture Cycle Time(ACT)	Design Stability Index(DSI) Design Defect Density(DDD) Review Efficiency(RE) Defect Removal Efficiency(DRE) First Time Right (FTR) Design Cycle Time(DCT)	Coding Stability Index(CSI) Coding Defect Density(CDD) Review Efficiency(RE) Defect Removal Efficiency(DRE) Function Point (FP)or Lines of coding(LOC) First Time Right (FTR) Coding Cycle Time(CCT)	Testing Stability Index(TSI) Productivity (for test case preparation) Test Case Effectiveness(TE) Testing Defect Density(TDD) Review Efficiency(RE) Defect Removal Efficiency(DRE) Test Case Defect Density(TCDD) First Time Right (FTR) Testing Effectiveness(TE) Testing Cycle Time(TCT)

The Fig-5 shows that the project team should analyse the impacting metrics at each phase in the life cycle model, so that the project managers can take appropriate actions to provide on time defect free deliverables and attain customer satisfaction.

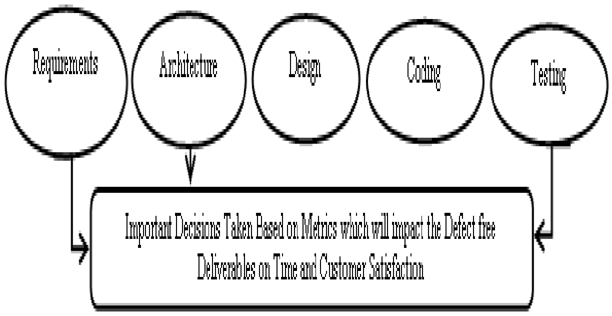

Requirements
Architecture
Design
Coding
Testing
Important Decisions Taken Based on Metrics which will impact the Defect free Deliverables on Time and Customer Satisfaction

Fig. 5: Decisions Based on Metrics.

6. Data collection methodologies

Data is collated from five different projects of various phases in the waterfall life cycle model. The impacting metrics in every phase are identified and appropriate statistical calculations were done to achieve it.Statistical Techniques Employedis:

1) ANOVA with assumptions

The one-way analysis of variance is used to test the claim that 3 or more population means are equal. The assumptions for ANOVA are:

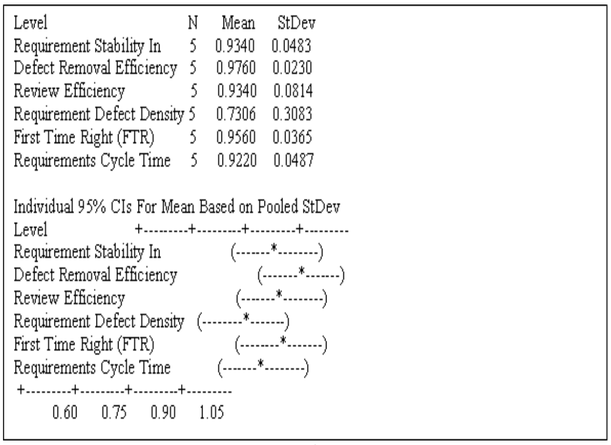
- The Data should be randomly sampled.
- The variance of each sample is assumed equal.
- The Residual are normally distributed.

2) Confidence interval (CI):

A confidence interval estimate of a parameter consists of an interval of numbers along with a probability that the interval contains the unknown parameter.The level of confidence in a confidence interval is a probability that represents the percentage of intervals that will contain if a large number of repeated samples are obtained. The construction of a confidence interval for the population mean depends upon three factors: The point estimate of the population, the level of confidence and the standard deviation of the sample mean. Phase wise implementation for the different phases is as follows:

i) Requirement phase

The data is collected from five different projects for the requirement phase. The important metrics such as requirement stability index, defect removal efficiency, review efficiency, requirement defect density, first time right, requirements cycle time are considered and ANOVA is carried out. After computing confidence interval, it is clear that the Defect Removal Efficiency, first time right, requirement stability index are the metrics the project manager should focus on. The Fig-6 shows the appropriate calculations for the requirement phase.



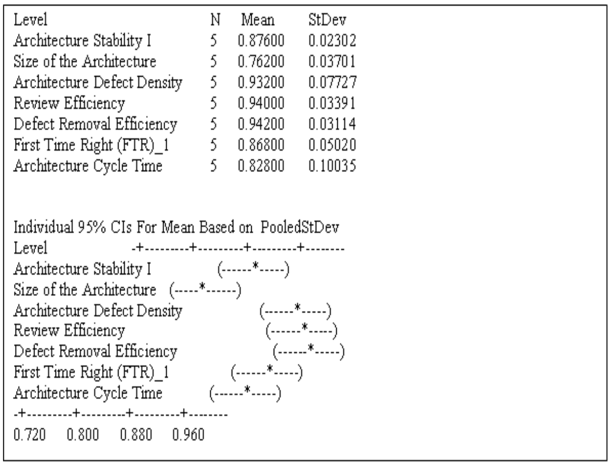
```
Level                       N    Mean    StDev
Requirement Stability In    5   0.9340   0.0483
Defect Removal Efficiency   5   0.9760   0.0230
Review Efficiency           5   0.9340   0.0814
Requirement Defect Density  5   0.7306   0.3083
First Time Right (FTR)      5   0.9560   0.0365
Requirements Cycle Time     5   0.9220   0.0487

Individual 95% CIs For Mean Based on Pooled StDev
Level                      +---------+---------+---------+---------
Requirement Stability In              (-------*--------)
Defect Removal Efficiency                  (-------*-------)
Review Efficiency                      (-------*--------)
Requirement Defect Density  (--------*-------)
First Time Right (FTR)                 (--------*-------)
Requirements Cycle Time             (-------*--------)
 +---------+---------+---------+---------
      0.60      0.75      0.90      1.05
```

Fig. 6: ANOVA and CI for Requirement Phase.

ii) Architecture Phase

The data is collected from five different projects for the Architecture phase. The important metrics such as Architecture Stability, Size of the Architecture, Architecture Defect Density, Review Efficiency, Defect Removal Efficiency, First Time Right, and Architecture Cycle Time are considered and ANOVA is carried out. After computing confidence interval, it is clear that the Defect Removal Efficiency, Review Efficiency, Architecture Defect Densityare the metrics the project manager should focus on. The Fig-7 shows the appropriate calculations for the architecture phase.



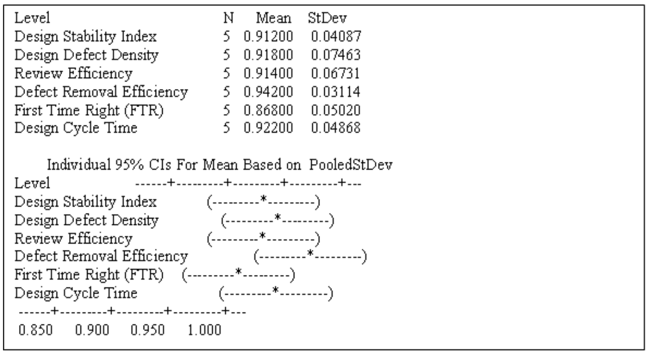
```
Level                        N   Mean     StDev
Architecture Stability I     5   0.87600  0.02302
Size of the Architecture     5   0.76200  0.03701
Architecture Defect Density  5   0.93200  0.07727
Review Efficiency            5   0.94000  0.03391
Defect Removal Efficiency    5   0.94200  0.03114
First Time Right (FTR)_1     5   0.86800  0.05020
Architecture Cycle Time      5   0.82800  0.10035


Individual 95% CIs For Mean Based on  PooledStDev
Level                  -+---------+---------+---------+--------
Architecture Stability I            (------*-----)
Size of the Architecture   (-----*------)
Architecture Defect Density                (------*-----)
Review Efficiency                            (------*-----)
Defect Removal Efficiency                      (------*-----)
First Time Right (FTR)_1              (------*-----)
Architecture Cycle Time             (------*-----)
-+---------+---------+---------+--------
0.720     0.800     0.880     0.960
```

Fig. 7: ANOVA and CI for Architecture Phase.

iii) Design phase

The data is collected from five different projects for the Design phase. The important metrics such as Design Stability Index, Design Defect Density, Review Efficiency, Defect Removal Efficiency, First Time Right, Design Cycle Time are considered and ANOVA is carried out. After computing confidence interval, it is clear that the Defect Removal Efficiency, Design cycle time, design defect density are the metrics the project manager should focus on. The Fig-8 shows the appropriate calculations for the design phase.



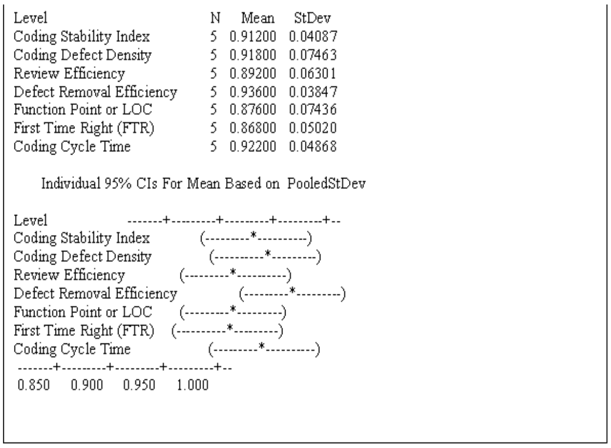
```
Level                       N   Mean     StDev
Design Stability Index      5   0.91200  0.04087
Design Defect Density       5   0.91800  0.07463
Review Efficiency           5   0.91400  0.06731
Defect Removal Efficiency   5   0.94200  0.03114
First Time Right (FTR)      5   0.86800  0.05020
Design Cycle Time           5   0.92200  0.04868

     Individual 95% CIs For Mean Based on  PooledStDev
Level                  ------+---------+---------+---------+---
Design Stability Index       (---------*---------)
Design Defect Density          (---------*---------)
Review Efficiency            (---------*---------)
Defect Removal Efficiency          (---------*---------)
First Time Right (FTR)   (---------*---------)
Design Cycle Time              (---------*---------)
 ------+---------+---------+---------+---
 0.850     0.900     0.950     1.000
```

Fig. 8: ANOVA and CI for Design Phase.

iv) Coding phase

The data is collected from five different projects for the coding phase. The important metrics such as Coding Stability Index, Coding Defect Density, Review Efficiency, Defect Removal Efficiency, Function Point, First Time Right, Coding Cycle Time are considered and ANOVA is carried out. After computing confidence interval, it is clear that the Defect Removal Efficiency, coding cycle time, coding defect density are the metrics the project manager should focus on. The Fig-9 shows the appropriate calculations for the coding phase.

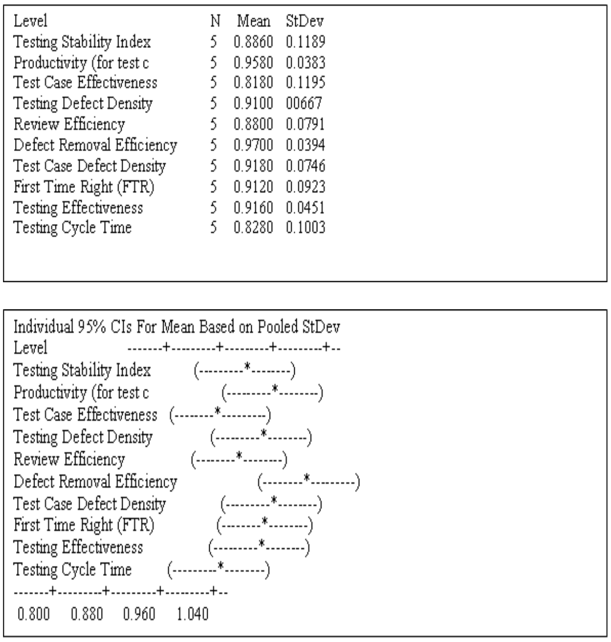


```
Level                       N   Mean     StDev
Coding Stability Index      5   0.91200  0.04087
Coding Defect Density       5   0.91800  0.07463
Review Efficiency           5   0.89200  0.06301
Defect Removal Efficiency   5   0.93600  0.03847
Function Point or LOC       5   0.87600  0.07436
First Time Right (FTR)      5   0.86800  0.05020
Coding Cycle Time           5   0.92200  0.04868

     Individual 95% CIs For Mean Based on  PooledStDev

Level                  -------+---------+---------+---------+--
Coding Stability Index          (---------*----------)
Coding Defect Density             (----------*---------)
Review Efficiency          (---------*----------)
Defect Removal Efficiency            (---------*---------)
Function Point or LOC      (--------*--------)
First Time Right (FTR)   (----------*--------)
Coding Cycle Time                 (---------*----------)
 -------+---------+---------+---------+--
 0.850     0.900     0.950     1.000
```

Fig. 9: ANOVA and CI for Coding Phase.



```
Level                       N   Mean    StDev
Testing Stability Index     5   0.8860  0.1189
Productivity (for test c    5   0.9580  0.0383
Test Case Effectiveness     5   0.8180  0.1195
Testing Defect Density      5   0.9100  00667
Review Efficiency           5   0.8800  0.0791
Defect Removal Efficiency   5   0.9700  0.0394
Test Case Defect Density    5   0.9180  0.0746
First Time Right (FTR)      5   0.9120  0.0923
Testing Effectiveness       5   0.9160  0.0451
Testing Cycle Time          5   0.8280  0.1003
```

```
Individual 95% CIs For Mean Based on Pooled StDev
Level                  -------+---------+---------+---------+--
Testing Stability Index      (---------*--------)
Productivity (for test c          (---------*--------)
Test Case Effectiveness  (--------*---------)
Testing Defect Density          (---------*--------)
Review Efficiency           (--------*--------)
Defect Removal Efficiency               (--------*---------)
Test Case Defect Density          (---------*--------)
First Time Right (FTR)           (--------*--------)
Testing Effectiveness           (---------*--------)
Testing Cycle Time       (---------*--------)
-------+---------+---------+---------+--
 0.800     0.880     0.960     1.040
```

Fig. 10: ANOVA and CI for Testing Phase.

Software metrics aid in the identification of these issues in a complex infrastructure utilizing multiple technologies. Additionally, software metrics help organizations determine team or vendor productivity regarding the amount of work put into an implementation and to check whether product meets end user requirements. Table-3 shows that defect removal efficiency is the most impacting metric in the SDLC phase. In addition, it is equally important to check the other metrics related to that phase.

v) Testing phase

The data is collected from five different projects for the Testing phase. The important metrics such as Testing Stability Index, Productivity(for test case preparation), Test Case Effectiveness, Testing Defect Density, Review Efficiency, Defect Removal Efficiency, Test Case Defect Density, First Time Right, Testing Effectiveness, and Testing Cycle Time are considered and ANOVA is carried out. After computing confidence interval, it is clear that the Defect Removal Efficiency, Productivity (for test case preparation),Test Case Defect Density are the metrics the project manager should focus on. The Fig-10 shows the appropriate calculations for the requirement phase

Table 3: Summary of Defects and Its Impact in the Different Phases

SDLC PHASE					
METRIC / PHASES	REQUIREMENT	ARCHICTURE	DESIGN	CODING	TESTING
1	Defect Removal Efficiency	Defect Removal Efficiency	Defect Removal Efficiency	Defect Removal Efficiency	Defect Removal Efficiency
2	First Time Right (FTR)	Review Efficiency	Design Cycle Time	Coding Cycle Time	Productivity (for test case preparation)
3	Requirement Stability Index	Architecture Defect Density	Design Defect Density	Coding Defect Density	Test Case Defect Density

7. Conclusion

The objective of this paper is to identify the suitable metric that affects the phase and project subsequently. It is necessary to analyse the metric in every phase and the relationship between them. The project team has to focus on metrics at every phase so that defect can be predicted in advance so that the project teams work accordingly. In the Requirement analysis phase metrics such as requirement stability index, defect removal efficiency, review efficiency, requirement defect density, first time right, requirements cycle time are considered. In architecture phase metrics such as Architecture Stability, Size of the Architecture, Architecture Defect Density, Review Efficiency, Defect Removal Efficiency, First Time Right, and Architecture Cycle Time are considered. In design phase metrics such as Design Stability Index, Design Defect Density, Review Efficiency, Defect Removal Efficiency, First Time Right, and Design Cycle Time are considered. In coding phase metrics such as Coding Stability Index, Coding Defect Density, Review Efficiency, Defect Removal Efficiency, Function Point, First Time Right, and Coding Cycle Time are considered. In testing phase metrics such as Testing Stability Index, Productivity (for test case preparation), Test Case Effectiveness, Testing Defect Density, Review Efficiency, Defect Removal Efficiency, Test Case Defect Density, First Time Right, Testing Effectiveness, and Testing Cycle Time are considered. The below table depicts the most metrics that needs to be focused by the project manager to achieve the goal.

The focus of the project managers should be on first Time Right, On Time Delivery, Customer Satisfaction, Profit Margin, Enhanced Business Operations, Costs, Project Compliance Index. Defect. Removal Efficiency, Defect Density is metrics that are closely related. When defect removal efficiency is high, the defect density becomes low at every phase. This paper can help us to reduce the rework and reduce the cost. In addition, extend this work further to other life cycle model such as Agile.

References

[1] G. Canfora a, F. Garcı´a b,*, M. Piattini b, F. Ruiz b, C.A. Visaggio, “A family of experiments to validate metrics for software process models”, *The Journal of Systems and Software* 77 (2005) 113–129.

[2] HaneenHijazi, ThairKhdour, AbdulsalamAlarabeyyat, “A Review of Risk Management in Different Software Development Methodologies”, *International Journal of Computer Applications* (0975 – 8887) Volume 45– No.7, May 2012

[3] P.Mahizharuvi, Dr.K.Alagarsamy, “A Security Approach in System Development Life Cycle”, *International Journal of Comp. Tech. Appl.*, Vol 2 (2), 253-257

[4] Youssef Bassil, “A Simulation Model for the Waterfall Software Development Life Cycle”, *International Journal of Engineering and Technology* Volume 2 No. 5, May 2012.

[5] DeepshikhaJamwa, “Analysis of Software Development Models”, *International Journal of Computer Science and Technology* Vol. 1, Issue 2, December 2010.

[6] MihaiLiviu DESPA, “Comparative study on software development methodologies”, *Database Systems Journal* vol. V, no. 3/2014.

[7] Rajesh H. Kulkarni, P.Padmanabham, K.K.Baseer, “Critical Review of Extended Waterfall Model”, *International Journal of Scientific & Engineering Research*, Volume 6, Issue 11, November-2015.

[8] MaaritTihinen and Päivi, Rob Kommeren, Jim Rotherham, “Metrics and Measurements in Global Software Development”, *International Journal on Advances in Software*,vol 5 no 3 & 4, year 2012.

[9] Unnati A. Patel, Niky K. Jain, “New Idea In Waterfall Model For Real Time Software Development”, *International Journal of Engineering Research & Technology (IJERT)*Vol. 2 Issue 4, April – 2013.

[10] PoojaJha, K S Patnaik, “Proposed Metrics for Process Capability Analysis in Improving Software Quality: An Empirical Study”, *International Journal of Software Engineering and Technology (IJSET)* Vol. 1, No. 3, December 2016.

[11] Mubarak Elamin and ElmubarakDaleel, “Software Engineering Development and Analysis of Life Cycle Models”, *International Journal of Computer Applications* (0975 – 8887) Volume 133 – No.10, January 2016.

[12] Adetokunbo A.A. Adenowoand Basirat A. Adenowo, “Software Engineering Methodologies: A Review of theWaterfall Model and Object-Oriented Approach”, *International Journal of Scientific & Engineering Research*, Volume 4, Issue 7, July-2013.