MASTER’S THESIS

2006:PM04

UNIVERSITY WEST

Use of Software Verification &Validation (V&V) Techniques for Software Process Improvement

Alphonce, Magori

EXAMENSARBETE
Magister med inriktning mot programvaruteknik Institutionen för teknik, matematik och datavetenskap

MASTER’S THESIS

Use of Software Verification &Validation (V&V) Techniques for Software Process Improvement

Sammanfattning

Ingen svensk sammanfattning finns då denna uppsats skrivits av en engelskspråkig student. Se "Abstract" for mer detaljer (på engelska).

Författare:	Alphonce Magori		
Examinator:	Dr. Samantha Jenkins		
Handledare:	Christian Ohlsson		
Program:	Software Engineering, 2006		
Ämne:	Software Engineering	**Examensnivå**:	Master
Datum:	June 2, 2006	**Rapportnummer**:	2006:PM04
Nyckelord:	Software Verification & Validation Techniques, Software Process improvement & software quality.		
Utgivare:	Högskolan Väst, Institutionen för teknik, matematik och datavetenskap, 461 86 Trollhättan Tel: 0520-22 30 00 Fax: 0520-22 32 99 Web: www.hv.se		

MASTER'S THESIS

Use of Software Verification &Validation (V&V) Techniques for Software Process Improvement

Summary

Software Verification & Validation activities are applied to all phases of software processes. The aim is to improve the software processes and have higher quality software, cheap and delivered faster to the end users. There are many verification and validation techniques that already exist and people apply them to the software processes but still have difficulties in helping software companies to deliver higher quality software within the required deadlines. Research presented in this thesis suggests Verification and Validation techniques that will help software development companies or organization to develop higher quality software that is delivered as fast as possible. In the project a comparisons are used to analyze existing verification and validation techniques in addition to the use of test cases. The major finding of this work is that Verification and Validation techniques will help provide higher quality software and that is deliverable in the required time period to the customer or end- user.

Author: Alphonce Magori
Examiner: Dr. Samantha Jenkins
Advisor: Christian Ohlsson
Programme: Software Engineering, 2006
Subject: Software Engineering **Level**: Master
Date: June 2, 2006 **Report Number**: 2006:PM04
Keywords Software Verification & Validation Techniques, Software Process improvement & software quality.

Publisher: University West, Department of Technology, Mathematics and Computer Science,
S-461 86 Trollhättan, SWEDEN
Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se

Acknowledgements

I am really very thankful to my advisor Mr. Christian Ohlsson and Dr. Samantha, Jenkins examiner. Because they helped a lot for this research paper and also thankful to all the staff members of this university who were guide me in my whole stay in Sweden.

Finally I would like to thank my parents and my family for supporting me through out this research.

Innehållsförteckning

MASTER'S THESIS

Use of Software Verification &Validation (V&V) Techniques for Software Process Improvement

Alphonce Magori
magori.alphonce@student.hv.se, amagori24@yahoo.com
Department of Mathematics, Technology &
Computer Science
University West -Sweden

Abstract

Software Verification & Validation activities are applied to all phases of software processes. The aim is to improve the software processes and have higher quality software, cheap and delivered faster to the end users. There are many verification and validation techniques that already exist and people apply them to the software processes but still have difficulties in helping software companies to deliver higher quality software within the required deadlines. Research presented in this thesis suggests Verification and Validation techniques that will help software development companies or organization to develop higher quality software that is delivered as fast as possible. In the project a comparisons are used to analyze existing verification and validation techniques in addition to the use of test cases. The major finding of this work is that Verification and Validation techniques will help provide higher quality software and that is deliverable in the required time period to the customer or end-user.

Keywords: Verification & Validation Techniques, Software Process improvement & software quality.

1. Introduction

It is true that there is a need of verifying and Validating the software products before delivering the final products to the end users [2,10,14,66]. The aim of doing verification and validation of software is that we need to check if software programs or systems meet the client's needs or specification [1, 10, 12, 56]. Many software companies or organization today are focusing on delivering software products with higher quality and delivered to the user within the short time according the market and need of the software products. There is a need of improving the software processes with the aim of increasing the software quality [2, 19]. Software processes improvement is the key of understanding the software processes used at the moment and change it with aim of improving the quality of the software products [2, 5, 33].

There are many software processes suggested like traditional 'waterfall model' and 'agile methods' [2, 19, 21]. That used to develop software within the software organization. People and organization working and developing software are looking on improving software processes, this done basing on what specific goals that the software developments organization or developer plan want to achieve[16,19] .There is no specific measure on how the software process improvement is measured but this can be observed due the results that obtained after using the one of specific process improvement and then the results will be compared to the actual planned results that the software organization have been planned before starting of the software projects[16].

Basing on this research that presented in this thesis .That focus on the detection and removal of the number of bugs during the software process, there is another way of observing if the software process is improved or not .This is through perceived problems from the past software development projects[58,59] . This means the new software process are compared with the past development approach of the past software project to see if the problems obtained during the software development are solved or not . And if the problem minimized as planned then software development organization or company will be able to say that there is software process improvement [3, 39]. This improvement requires a lot the use of verification and validation activities to be applied during each phase from the requirements or specification as the early phase of the software development and before deliver of the software products to the end users [7, 52, 53].

There are many traditional verification and validation techniques existing and discussed through the different literatures [1, 10, 43, 56].Also many organization and individual people are applied them today. But some have difficulties on helping them on improving the quality of the software products, and deliver the software within the short time [59]. This means the software development organization focus on the developing the software faster and cheaper and also need to meet the software market . Therefore some difficulties they get once apply some of the software verification and validation techniques like software testing there are not sure about when they have to stop the software testing and also doing only software testing to the software project is not enough to establish the confidence of using the software product s[15,48, 53].

Therefore, due to the software project failure and software application failure. This research presented the need of verifying and validating of the software product at the early phase of software development [19]. Because it is difficult to detect all the software error basing on the software testing techniques only as the verification and validation techniques that applied to many software organization now days. Another difficult of using the software testing as the software verification and validation is that provide un clear information that all bugs are removed or not[13,15]. This take a lot of time to fix those bugs, therefore overall difficult people they get of using some of the verification and validation techniques is that take a lot of time to fix the defects of their software system ,and it is not possible to involve the stakeholder that are going to use that software project .For example using the software testing only is not enough to trust that there is no any defects in a specific software system [49, 56, 66].

Research in this thesis present verification and validation techniques that will solve this problem this means we are focusing on the verification and validation that will be able to remove number of bugs or defects at early stages and lead to the higher quality software products within the short time i.e. meet the market demands [14, 19, 66]. Many countries use software a system through different applications .And it is true that the failure of the software operations can cause physical damage, economical loss or loss of life [2]. This is true that many software applications applied to the government institution, social and critical system [2] and also software application applied to economical institutions. Therefore once there is any failure in one of among those areas can lead to the failure of the operation of the overall system and due to that this can also lead to the loss of the data or information once there is failure of operation occur in transaction system especially bank system.

Therefore in order to make sure that the overall system is working as required is very important to use verification and validation techniques that focusing at the early phase of the system development[51,56]. Take the example of the development of the critical system. The major factor that must be considered is the use of the formal methods techniques. This technique involves the formal specification [2] and this is also the verification and validation techniques that involved at the early phase of the software system development and helping to avoid the software system failure. Also it avoid the time spend during the software system development .Because this make sure that the system meet the specified requirement [10, 56]. This means that verification and validation processes applied through different areas like hospitals, bank systems, central systems and real-time systems. Therefore there is a need for verifying and validating the software systems before delivering them to customers or end users doing this will avoid the problem of system failure that can lead to the consequences of the software system operations [2, 53].

This thesis is based on the software engineering fields and highlights verification and validation techniques among those verification and validation that already exist and discussed through the different literatures [1, 2, 9, 10, 43].That will help to improve the software processes and software products will be produced at higher quality and within a short time[28,46]. There are many traditional verification and validation techniques that are already existing like software reviews, software inspections, formal methods, software testing[10,43,56], but some have some strengths and some have weaknesses once used or applied, take a lot of time for discovering bugs or removing bugs[33,34,49,53]. This research presented in this thesis proposes to use the verification and validation techniques that will be able to improve the software processes and deliver the higher software quality to the end user within a short time, developing the software products according the quality required and software market demands [8, 53].

This paper is focusing on the different kinds of software verification and validation techniques which help for improvement of software processes [1,59]. Here we are following the software process improvement, verification and validation techniques as well as software qualities with their characteristics in the back ground section. Our paper methodology is in section three, depending on the literature survey and also based on experiments which we did in personal software process improvement. Results of this whole work are based on literature as well as experiments. The discussion is in the section five of this paper and after this we come on the section six, conclusion. The section seven is about the future planning of the paper and the references of the material are in the list of references, section nine.

2. Background

This part is depending on the different issues which are necessary for better understanding of the materials concerned the paper. The background section is divided into the following sections below:

2.1 Software process improvement

When we talk about software process improvement, we are focusing on improving the software processes for the intent of increasing the quality of the software products [2, 4, 8, 33]. This can be done through understanding the original software process and change it in order to increase the quality of the software products [21, 33]. There are many software process improvement approaches [2, 7, 19, 33]. That is focusing on increasing productivity of the software products within the organization form individual level to the organization level. Personal software process (PSP) and Team software process (TSP) are the approaches that focusing on increasing productivity for the individual work or team work that performance can be increased [19,33]. These are framework of increase performance at each stage of development. Software Engineering Institute (SEI) suggested many software process improvement approaches.

The Capability Maturity Model (CMM) [20, 41] is one of the process improvement framework suggested by the Software Engineering Institute (SEI) [2, 5, 41]. This is the model that used to determine the process maturity of the software development organization. Also the personal capability maturity model (P-CMM) [2, 20, 41] is the model for improving the level of individual within the organization this means that people working within software engineering area are required to improve their competence and have new knowledge according to change in software engineering technology.

Personal software process (PSP) and the Team software process (TSP) approaches helps a lot for engineers working alone or working on team basis to increase their performance and have higher software quality[7,19,33]. This also helps a lot on the software project management and the progress can be seen as early stages. Personal Software Process (PSP) is more applicable during the individual bases improvement. Software engineers need to see improvement of their performance during the software development increases and the higher software quality are reached [42,46].

These include the planning of the activities .choosing the software development metrics and follow them .Software metrics help on the measuring of what resources, processes and products [29, 30, 33] that involved in the software projects. The use of the software metrics helps much on how we can control the software project there are different number of metrics that help a lot software development organization to measure and also software development organization can be able to control what they have planned from the first time because it is very hard to control what have been planned if at the begin no any measure have been done before[36]. Software metrics can be categorized into three parts. These include the software resources metrics, software products metrics and software process metrics [22, 28, 46]. All these metrics have great advantage of helping the software development organization to follow what they have planned before and make sure they control the software development project[7 , 19, 28].

Therefore when we talk about the software metrics we focus on how we can measure the and monitor the software processes, software products and software resources[49,50] that will be involved in the software project, the measures of all software products and resources that can be usefully in developing the software projects. This helps a lot on relating the new development system and the existing system.

2.1.1 Software resources metrics

The software development companies use the resource metrics in order to see the progress of the software project. The resources are basing of specific development company depending of how large the software project are developed and how many number of resources that are required in order to make the software project successfully [35,38]. Therefore software resources metrics depends according to what the software development company has planned to do. And this should reflect the specific goals of that organization [25]. There are many number of the software resources metrics that will help the software development to measure and monitor the software projects [47, 50].

Research in this thesis present some of software resource metrics that help for monitoring the software project. And also it is very easy to see if there is need of improving the software process of development or not. Basing of the software resources that are allocated to accomplish the software projects .And how the planning done for developing software projects. Software resources metrics includes the general what quality of the development team they have. This means the general experience, efficiency and flexibility of the software development members that involved in the software project must be considered [30,38]. There is need of considering how the technology needed and required to make the software projects. And how the new technology should be adapted .This is very important factor to be considered because now in the world there are rapid changes of the software technology.

Therefore there is a need of software development organization to monitor the existing software technology and discuss how the new software technology will be able to replace the old technologies [24].

Software resources metrics also focus on how to manage the software projects how to size of elements and structure of software elements to be controlled and followed should be discussed in this type of software metric. That helps the software development organization to see if they reach the specified goal that planned at the beginning of the software development and planned stage [36,39].

2.1.2 Software products metrics

When it comes to software products metrics, it is true that software development companies and software developer need to estimate the size, time, cost and the standardization of the software products that they needed to be developed. And follow them in order to be easy for them to get clear picture of what products are going to be developed at the specific software project [22, 31,46]. This is to say basing of the software product metrics we can be able to estimate the size of the software project that needed to be developed. For example software developer can be able to know how many lines of code are needed during the software coding and it is easy to estimate and follow. What number of the documentation that we need to use in the specific software project, not only that using and follow the software product metrics there is number of advantage we can get for software process improvement. For example we can be able to understand what and how many components that are needed for the developing the software program or system [29] because there is possibility of using the different component from the different vendors if the product metrics well taken and followed during the software process improvement. It is true that each software component developed by specific vendors has they own standard [24,25].

Therefore following of this metrics provide also the standardization of the components that involved in developing the software system as specified from the original software projects. Also using the software product metrics help the software development company or organization to estimate the size software products, these include lines of code. Also it is very easy to understand how many test cases will be derived in specific program. Knowing all the derived test cases that involved in the software program it is easy to test and know how many test cases passed successfully and how many test cases have been failed [10,13]. This metrics when applied during the software development makes it easy to establish confidence of using the system. Because it is easy to understand what products used also these helping a lot for the software process improvement because the software developer can be able to understand how many lines of code involved in the specific program and reuse some of the line of the code for developing the new program as the concept of re using components is concerned [24,27].

Research presented in this thesis also shows that following these metrics it easy also to discover the software products that are not working as required through reviewing the software development documents that involved in the software development processes [36].

2.1.3 Software processes metrics

When we it comes to the software process metrics it is true that software development process contains different phase these includes the identifying the problem that must be solved, the software analysis phase, software design phase and software implementation phase [2,21, 28]. Therefore software process metrics provide the measure of all the software development phase from the requirement gathering to the implementation phase. Through measuring all the phase it is easy to control and follow each process [31,32]. Applicability of software process metrics it easy to understand what levels of component are needed [24], and resources that will be involved in the software development. These include the stakeholder capability and efficiency the management goals and expectation from each software development process [22,27]. Each phase of using this metrics should be allocate number of metrics to calculate and follow them doing so it is very easy to control the software development process, due to what have been suggested from each phase of the software development [7, 19, 33].

Basing on the research in this thesis that we need to improve the software process and have the higher quality software this means that once the all process included in the software processes are measured it is easy for the software development organization or developers to control their software processes. And following these metrics will avoid to the consequence of the project failure [3,7,19]. Also using the Verification and Validation techniques for the software process improvement there are number of advantage from using these metrics because, it is easy to trace the problem through reviewing each metrics that involved in the software development processes [19,33].

These metrics increase the software productivity of the software products, using these software metrics through the software process improvement approaches. It is easy to have higher quality software, also the bugs and defects can be discovered easily when injected or removed with the software system. This can be done through the use good measurement of metrics and the control of software process improvement. Many software engineers with knowledge of personal software process (PSP) can work together and improving the performance of their work using team software process approach (TSP)[5]. Team software process (TSP) helps to simplify work and people are able to exchange knowledge from different software engineering skills thus leading to higher quality software, accurately and minimize the project failure.

Many software process improvement approaches are iterative and continuous approach Therefore they use Plan-Do-Check-Act (PDCA) processes [41, 42]. This is an iterative approach of software development. Using this approach there is no time limit on application for software process improvement. It is possible to restart with planning stage of the cycle according what is required to be done in order to meet the goal of the software project .This development cycle is focusing to meet the need of the specified requirement. Also time spend for each stage of cycle depends on what activities have to be done for each stage. There is no limitation of time on the development software and this cycle focus on the problem solving. If it happens that the problem is not solved after the complementation of the all stage of the cycle. The cycle will restart gain [41].

2.1.4 Plan-Do-Check-Act (PDCA)

The cycle consists on four steps and as explained in details below:

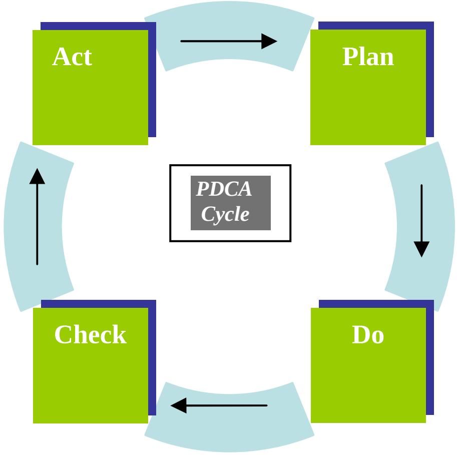


Figure 1: Plan-Do-Check-Act cycle [65]

2.1.4.1 Planning

This usually used to suggest and identify the processes that should be used for the software process improvement; also this step is used to define the problems that should be solved and finding the source of Problems.

2.1.4.2 Doing

This section includes the implementation of the Processes suggested and implementing the suggested solutions for the planning section. This is important part where by the implementation of the suggested solution should be done carefully for better solution.

2.1.4.3 Checking

This part is used to check if the solutions have been implemented correctly or not. These include the steps of reviewing the measures that have been taken before and reviews can be done before. The aim of reviewing is to see if the specification is related to the solution implemented or not.

2.1.4.4 Acting

During the acting process we focus on following up of all implemented solution(s) as the suggested solution(s) above.

Plan-Do-Check-Act (PDCA) cycle helps to know what inputs and outputs are used during the development in software products. This means it shows if the solutions obtained are working as required, efficiently and reliable for the suggested systems [6, 41]. Based on the software processes improvement, we are focusing on selecting appropriate processes and proper methods that should be used for improving the quality of the software products. This includes the understanding of the software process that have been used at that moment and changing it to increase the performance [6, 7, 19, 33].

The aim of this software process improvement used within the software companies or industry is to help them to improve the performance and quality of the products according to the needs of demands [2]. Also improve the individual skills and performance of doing work within the software engineering areas.

2.2 Software Verification & Validation (V&V)

When it comes to software verification and validation, it is true that there is a need of Verifying and validating of the software products before delivering them to the end-users [1, 10, 56]. Doing this avoids the consequences of the software system failure [1,2,15]. Verification and Validation activities are applied through all phases of software processes during the software Development [21,2]. Verification and validation are working together but they are not the same. Validation is used to show if the’ right product has been produced’ [1,.2] This means the products meet the specified requirements. Verification is used to show the ‘performance of the software products’ [2,15]. But both verification and validation show the correctness and accurateness of the software products produced. Verification and Validation are applied with intent to show if there are presence of bugs or defects. This is to say Verification and Validation of software done with intension of finding the presence of faults in the software systems [9,11]. For example the missing bracket functions or semicolon during the software development [1]. After observation of faults or defects they need to be corrected. Verification and validation are mostly used to minimize number of faults in the software systems.

There are many software Verification and validation methodologies and techniques discussed through the literature [1, 43] and used in many software companies or industries now days like software inspections, software reviews, formal method, static analysis, software testing and etc but all are intent to show the presence of faults or bugs and prevent them [1]. Many organization or companies dealing with software development apply the verification and validation techniques existing today during the software development and system development Other verification and validation applied during all stages of software and still other are applied before the end products delivered to the end users. But the choice of verification and validation techniques is very important in order to reach the specific goal. Software development companies are organization are focusing on using verification and validation techniques that leads to the higher quality software products and detect and remove faults at the early stages of the software development processes, within a short time, error can be corrected as soon as observed once the verification and validation process are taken serious at the early phase of the software development [15,56]. For example software process improvement use the Plan –Do-Check –Acting phase, after the checking stage of the cycle it is observed that there is presence of the bugs in the specific software system it is easy to prevent and remove the errors at the acting stage but this cycle can restart again depending on the need of the software system.

In order to avoid the software or system failure the verification and validation should be able to prevent the number of faults at the early stage of the development so that to avoid spending a lot of time during the software testing. Therefore the use of verification and validation techniques is the key of success of the software project increasing the number of faults prevented and avoided and leads to the higher quality software quality [44].

2.3 Software quality & characteristics of quality software

Quality of software is very important for the software market now days. When we talk about the software quality we mean that the ‘specified software products or systems meet the specified requirements or specifications’ [8, 37]. This avoids the consequences or problems of the system failure like socio-technical systems and safety critical system in which these types of the software system depend on the software, hardware and people are also the part of the system. Therefore once this system is developed there is need of meeting the specified requirement. That is to say when these system meet the specified requirement usually are grouped as the software quality products [40,46]. Let us take for example air craft operation as one type of critical systems, where software which once one part fails to operate well can lead to the failure of all the systems and eventually can leads to the lose of life of people [2]. Software quality depends on two major attributes or factors. These are software processes that can be used to develop the systems and product that is used to make the specified computer system. In short we call them the ‘process related quality’ and the ‘product related quality’ [2, 17, 40]. These two attributes are very important for software developing companies to have higher quality software [2, 4, and 8]. Products process related quality and product related quality attributes contain a number of the different attributes that are very important during software development and are very important factors to be considered for producing the quality software products within a short time [4].

Process related quality addresses issues such as cost of developing the software, the time that the system or software started to be developed and the finishing time, project lead time and the timelines of the software project. When it comes to the product related quality, we have also numbers of attributes we are focusing to consider. These include efficiency, reliability, usability and maintainability of the software products. Therefore, the two attributes should be balanced [2, 22, 27]. Doing so, the higher quality software can be reached [17]. It is true that for many software companies or industries developing software systems focusing on delivering higher quality software to the end user or customers and within the short time [17]. It is not recommended to wait to deliver the software products to the end user customers until the demands of that software product is passed or the demand of that software product is out of the market demand [2, 4, 8]. Cost of the software projects and the timeliness are the major factors for the success of the quality software products [4, 8].

S.	**Process Related Quality**	**Product Related Quality**
1.	Timelines	Maintainability
2.	Cost	Usability
3.	Development Time	Reliability
4.	Project lead Time	Efficiency

Figure 2, Processes & product related quality [64]

3. Methods

The methods are following below:

3.1 Literature Survey

The literature survey in this thesis focusing on the peer – reviewed materials this including the different books and articles from different authors focusing on the Verification and Validation techniques used to provide the quality inputs of this research presented in this thesis[62]. Approximately 40 books where reviewed and this used as the primary literature survey method. Many quality inputs obtained from here from different number of authors who wrote about the Verification and validation processes, software process improvement approaches, software quality and software testing.

Collection of the different articles basing on the research presented in thesis reviewed and input information obtained to provide information that used to support the research presented in this thesis.

Also in order to have quality inputs of information basing on the Verification and Validation techniques, software testing, software quality and software quality. This research used to compare different ideals from the different authors using the existing publication using the keywords from the source database connected to the search engine like www.google.com.

Generally the relevance materials and quality input basing on this research presented in this thesis. Where obtained when the Software Verification and Validation books reviewed, pre-knowledge of the Verification and Validation process .Also different publication that obtained from the source database connected to search engines used in this thesis to provide the quality input of information [63].

3.2 The Experiment

Research Presented in this Thesis not used full implementation but the experiment of individual project using the Software process improvement used to support the results obtained in thesis. The experiment done with intension of improving the quality of the software program through following different software metrics like time metrics, testing metrics, design review metrics and etc [61, 63].

4. Results

We derived some results from literature survey as well as from experimental which are given below:

4.1 Over view of the Literature survey

Here we are categories in different parts which are below:

4.1.1 Software testing Technique

Software testing is the one of the techniques of the software Verification and Validation. As we know the Verification and Validation processes applied to the software processes to verify that the software systems or programs meet or consistent with the specified specifications of the suggested software program or systems[43,56] . Software testing is one of the verification and validation techniques that involve the running of the implementation of the software programs where by data have to be tested[11-15] .Usually software testing used to show the presence of bugs or defects that introduced to the software programs[11,52].

Many software companies and software developer apply software testing as one of the verification and validation techniques for intent of showing the presence of bugs to their software systems and find the means of removing errors from the software systems. Verification and validation applied as dynamic verification and validation techniques. This means changes according to the time and the specific software to be tested at that moment [2, 10, 13]. This consists of two types of software testing that many software companies apply them today these includes.

4.1.1.1 Validation Testing

The validation used to show that the software that developed meet the clients or customer needs and meet the specified requirements according to the market demands[2,40] .

4.1.1.2 Defect Testing

This is one of the software validation techniques that applied to show the general inconsistence of the software systems that produced or the software system that developed and the meet the requirements that specified before[2,13,15] . Generally used to show the inconsistence between the system produced and the requirements that have been specified to build that system. Defect testing and validation testing are working together as the software testing approaches as well as the verification and validation techniques. But the deference to this two techniques is that during the validation testing it is easy to find the presence of defects or bugs of that system and using the defect testing it is not easy . As we know the key success of the software testing techniques is to be able to detect the presence of bugs or defects to the software system or program [11, 15, 48].

There is no important to perform the software testing if there is no possibility of discovering the presence of bugs or defects to the specified software [11]. Software testing as the one of the verification and validation techniques applicable to test many software system application like critical systems, Real-Time system and many software system application in order to remove the errors that can lead to the faults of the software system . Failure of much software system can lead to the consequence in life [1, 2].

Therefore software testing is very important techniques for the verification and validation techniques that help to detect the number of defects and then its is easy to take measure on removing the defects discovered from the software system[53,59]. Testing of the software system are categorized into different levels basing on what to be tested . Research presented in this thesis present number of levels of the software testing the can help to find number of errors within the software products .

4.1.1.3 Software Unit testing or Component Testing

This is one of level of software testing [9-13] used in testing the software system where by the component are tested before being used Or individual component are tested that involved in the design of the software system . Applying this technique it is easy to see if the component used or the component that will be used in designing the software application will be able to meet the requirement specification[10,11,14]. It is true that many software organizations now apply the component technology [24]. In order to establish the confidence of using component developed by different provider these techniques is more useful in this thesis as one of the Verification and Validation techniques and generally included in the software Techniques. Using this software technique in software process improvement help on avoiding the failure of the software application system like critical and socio-technical systems [2]. This means the component are easily to be tested and confirmed if are working properly .This is one of the verification and validation and verification techniques basing of the software testing that help on finding if the component are working proper or not[1].

4.1.1.4 Software Integration Testing

Software integration is one of levels of software testing [13, 48]. Also is verification and validation technique that have number of advantage on software process improvement .This techniques involves the many components and many components must be tested to see if no problem occurred during the building of the system or during the system integration[11, 13,15]. This is to say the applicability of this technique to the software verification and validation basing on how the software system are developed and components how connected to form the overall system. This technique applied to many software organization to check if there is any faults occurred during the system building or not and each component that involved in the software development are tested. Doing that keep the confidence of the using the software system later on [53, 59]. And also avoid the failure of the software system because overall system is tested and checked if there are errors occurred during the integration of the components that involved on building the system[2,14] .

4.1.1.5 Software Acceptance testing

This level of the software testing techniques [13, 48] and involves large number of customer or end user that use the system. The end user use to test if the systems are going to use is meeting the requirement that specified. This involves the testing of the system functionality and this help to make confidence of what system is usefully for their applications. Also this is the one of the software verification and validation techniques that focusing on checking how system is performing work. This technique involves many stakeholders that use the specified system. It is easy to see if the software system meet the specified requirement and this help the customer to accept or reject the final project that developed by the developer.

Testing the software system before the use of it is very important and this give the clear knowledge as soon as possible if the final software product is going to be reached or not .This can help on improving the process of development because if there is observation that the software system not meet the requirement that needed by the user or not meet the specified requirements. It easy for the developers to check what was wrong and this done not for the purpose of punishing the software developers or the make complaining to the software developing company but this help on improving the software process to the next development. Also focusing to have the higher quality software in the future [33, 46]. Therefore the type of the software testing works also as the verification and validation techniques to make sure customer get what they want and help on making the customer to trust software system that are going to use .

4.1.1.6 Software System and Regression Testing

Software system testing and software regression testing are the two levels of software testing [26].Apart from the software component or unit testing and software integration testing[11,13] that surveyed in this research presented in this thesis .This is to say apart of testing the individual component ,also the overall system can be tested to see if there is no problem during the exchange of the information from one component to another .Software system testing make sure there is flow of the information among the connected component used to form the entire software system . Doing that is very easy to avoid miss communication among the component that used to form the specific software system [24, 53] .Let take example of the airplane if there one operation fail to operate properly it can cause accident. Therefore the software system testing as discussed from literature sources when performed help to understand if there is passage of information among the system components[11] . Also this is Verification and Validation techniques because software testing is also the Verification and Validation techniques .Apart from the software system testing level there is software Regression software testing .This level of testing software is basing on finding if there any probability of change of the software functionality when the change occur among one of the software component[11]. This is the critical issue because once one component among the different component used to form the specific system change there is possibility of change of operation of that system . Because during the change there is possibility of missing of the software requirement and this can lead to the failure operation of the software system.

Therefore applicability of the software regression testing is to make sure it discover if there is functionality change of the software system when there change occur among the software component, that used to form the software component [11-13] .Basing on the research question presented in this thesis these two level of software testing does not give the clear picture on how the software defects can be prevented but provide the means of testing the software system . Software testing is not enough to establish the confidence of using the software system therefore this techniques help on improving the software process but not give information on how the software defects can be prevented and voided . These two levels of the software testing provide only the way on how to detect the software defect. But these two levels not provide the means of preventing the software defects or bugs within the software engineering project.

4.1.2 Software Design review Technique

Software design review is used as verification and validation techniques to the software process improvements .This technique used by software development to review the design and the requirements[8,60].Design review techniques applied to the early phase of the software processes[19] .It is easy to detect the defects of the software design if there is some missing function using this techniques as soon as possible when all the documents of the design are review at the early phase of the software processes . Using this technique it is easy to avoid using the large number of resources to develop the software. That will not provide software products that required by the customer or according to the market need [10, 56].

This technique is performed by number of people .This includes the developers and all the stakeholders that involved to that software system . Doing that help development team to know the progress of the software system .This is to say using the design review technique it is easy to manage and evaluate the development software projects. Also it is easy to confirm that the software system will meet the customer needs at the early stage of the software development before the final products are achieved [66]. Design review approach which is one of software verification and validation techniques help a lot on showing the completeness , the correctness and accuracy of the software system .Doing so it is easy to establish the proper basis to the next phase of the software development[21,43].

4.1.3 Software code review Technique

This is one of the verification and validation techniques applied to the software process improvement in order to help the software process improvement This approach is used to many software developing companies to have high quality products of the software[54,56,60] , system or programs and software products delivered to the customers or clients within a short time. After code being written they should be reviewed this is applied as one of the quality assurance activities as well as the Verification and Validation processes to the software process improvements [9, 10]. In order to have higher quality software ,code review technique used to make sure that code written by the developer or software engineers are correct and in order to avoid spending a lot of time during the software testing this techniques prevent the introduction of defects during the software testing at the early phase of the software development .Code review establish the confidence of codes before tested . Testing software is very important but in a real sense using the software testing as the verification and validation techniques is not enough to the software to be sufficient for the use of the software system [12,15]. In order to establish the confidence of using that software system there is need of reviewing code that written before tested and doing so the early progress of the of the software system developed can be seen .This technique is going parallel with the software design review and software inspection techniques that used to prevent error , detect and remove error at the early phase of the software development[2].

4.1.4 Software inspection Technique

Software inspection is one of the one software verification and validation techniques and used to the software process improvement. This technique involves the inspection of the all the documents , requirements and implementation of the software programs or system[14, 51] .This usually involves large number of the stakeholder that involve to the software projects. These include the software developer, customer and software engineers. This technique also is called software peer review is performing work as software review techniques but this take large space of application compared to the software design review and software code where by one specific area is inspected but software inspection technique is applicable for all part of the software development and involve of the stakeholder of that software system, using this technique each phase of the software development is inspected and reviewed at the same time, instead of waiting to test the complete software system[54,60].

Due to some research that have been conducted by some software organization that use this techniques it shows that using this technique it is easy to improve the standard documentation of the developing software projects for future. It is true that people changes but standard that used to develop the software system can remain the same for number of years. Therefore this techniques help on improving the documentation for intent of improving the quality of the software product.

Also using this technique it is easy to see the visibility of the software project that developed before the final products are delivered to the customer. This means using the technique all the stakeholders are involved in software Project development This it can be easy for the end user to trust the system their going to use [2,51]. Therefore software inspection techniques when used as verification and validation process help on improving the software process. Also doing that it is easy to have higher quality software products [12, 40].

4.1.5 Comparison between the software design review and software code review techniques

Both software design review and the software code review are applied as verification and validation techniques for the software process improvement .And for improving the quality of the software products[2,54] . Therefore both always used to be applied at the early phase of the software system development. Software design review used to review all the requirements and design .Also improve them for the intent of obtain the higher quality software products and developing the software system within a short time.

The software code review used to show that the code written without error by considering the design that have been made before [54]. Therefore software design review and the software code review are the depending each other .when the design is done correctly it is easy to map the design into coding [1,2] . This is to say the better the design the better the code produced. Therefore both techniques used to improve the software processes .And doing so higher quality of the software products are reached. Also the progress of the software development can be observed at the early stage of the software development [12].

4.1.6 Comparison between software code review, software design review and software testing Techniques

In order to establish the confidence of the software program or software system we should apply the software code review and software design review techniques that will help through preventing the introduction of bugs or defects from the early phase of the software development processes instead of waiting at the end of software development[15, 51]. Software design review and software code review techniques are working together. Both are applied with intent of detect and remove the large number of the bugs or defects at the early phases of the software development product [9] .This help to check if the correct code have been written before complied .Doing so help the software testing process [19, 54]. And this will not take a lot of time during the testing software. Therefore the software testing depends on the sufficient work that done by the software design review and software code review and other verification and validation techniques that applied at the early phase of the software development process[1,21] . Doing the software testing and driving the software test case is not enough to making sure that all the defects have been removed but software testing intent to show the presence of bugs or defects that introduced to the specific software program or system [13,49].This will consume a lot of time if large number of the defects will be found during the software testing phase and sometime leads to the software failure [2,11].

Therefore in order to minimize the time spend during the software testing it is advice to take serious measures on the verification and validation techniques that used at the early phase of the software development .This is to say the number of defects can be discovered during the software testing depends of the what have been done before if the software design review and software code review have done correctly there is possibility of escaping the software testing[19]. But if the verification and validation process are not applied correctly at the early phase of the software development processes there is possibility of detecting the larger number of the bugs in the software program and sometime lead to software project failure[1,53, 59].

4.1.7 Comparison between software inspection and software testing Techniques

Software inspection is applicable as software review techniques. This means used this technique all the documents are reviewed to check if there are any defects injected and prevent [2, 51, 60]. Also software inspection techniques involve the all the software process activities and involve also the stakeholders of the specified software system .Using the software inspection as the verification and validation techniques. It is easy to see the progress of the software project at the early phase of the software development. And it is easy to prevent the large number of bugs or defects at the early phase of the software development[2,19,54] Coming to the software testing techniques is that software testing technique depends of the good job done by the software inspection during the development software system[19].The better the software inspection job the better the software testing. This is to say once all software documents are reviewed at the early phase of the software development and also the large number of defects once are prevented at the early stage of the software development This help to decrease large number of defects that can be found during the software testing[51,52,53,60] . Doing the software testing is not enough to establish the confidence of using software system but doing the software testing if to show the presence of bugs that are involved in that system[53,56,59].

Therefore in order to avoid to spend a lot of time for finding and fixing larger number of bugs during the during the software testing. It is advised in this research paper that the use if software inspection technique must taken serious at the early phase of the software development process .And this must applied as the verification and validation techniques to help the organization to deliver the higher quality software and within the short time instead of spending the large amount of time on testing the software system[19,60] . In order to, show the presence of bugs without knowing how to prevent them. But both software testing and software inspection are used as verification and validation techniques to help software developing organization to detect , prevents and avoid the bugs or defects of the software system during the system development and before delivering to the customer[1,10,15 ,56].

4.2 Result from the literature survey

Results obtained after the surveying of the deferent literature sources . And by comparing the different existing software Verification and Validation techniques . It is Show that the Verification and Validation techniques that applied at the early phase of the software development have great advantage on improving the software development processes. And the higher quality of the software products are easy to be obtained from the early phase of the software development [9,15,66] .

Research presented in this thesis also observe that once the Verification and Validation techniques applied at the early phase of the software development it is easy to see the progress of the of the software projects as soon as possible[8,54,66] . Also result from the literature survey show that the software testing is one of the software Verification and validation techniques. But doing software testing of the software is not enough to establish the confidence of the using the software system, because software testing is focusing show presence bugs that is not a key factor of improving the software process [51,56].

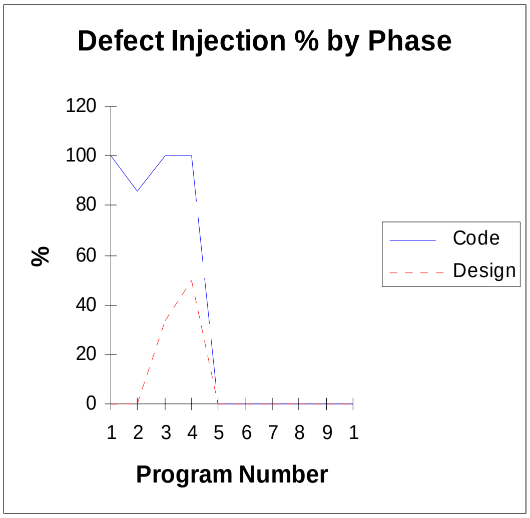
Therefore after reviewing the literature survey we came to conclude that the software inspections , software review , software code review formal specification and software design review have big advantage on helping the software developing organization to obtain the higher quality software. Because once the Verification and Validation techniques are applied through the all phases of the software development. This help on increasing the understanding of the software system .Many error that detected at the development phase of the software development is very costly and expensive the be corrected compared to the detection of the software error at the early phase of the software development [1,19,21] . And also the software inspection involves all stakeholders that are going to use the software products of the developed system [14, 60]. Also after reviewing the software testing techniques we see that software testing is the best way of showing the presences of defects on the software system but doing software is not enough to establish the confidence of using the software system[9-13]. Because we are not sure just after stopping the software testing all the defects are removed or not [48, 53].

Therefore this observed that it consume a lot of time on doing software testing compared to the software inspections and software review Techniques[19,53,56, 59] ,

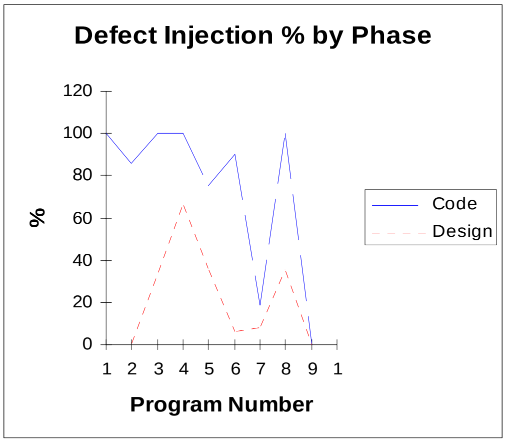
4.3 Results from Experiment [See Appendix A-E, 61]

This research presented in thesis is not using full implementation but the experimentation basing on the personal software process improvement project [7,19, 61] of the software engineering area used to support the results obtained. And we have obtained many conclusion from the results obtained. We used the personal software process improvement approaches. Where each process checked and some verification and validation where compared due to their applicability and efficient on helping the software processes improvement approaches to produces the higher quality software products and within the short time [19, 40, 42] . Experiment in this Thesis basing on development of deferent software programs with intent of improving the quality of each program .Through following of software metrics and improving each metrics This includes the time metrics, testing metrics ,code review metrics design review metrics and etc [19, 47,49,50] for each phase of the software development process[49, 50] .

Results of the experiment in this thesis are shown through the deferent graphs below. That obtained during the software development, graphs shows how the number of defects injected and removed through each phase. And from each graphs there are numbers of observations that have been stated.

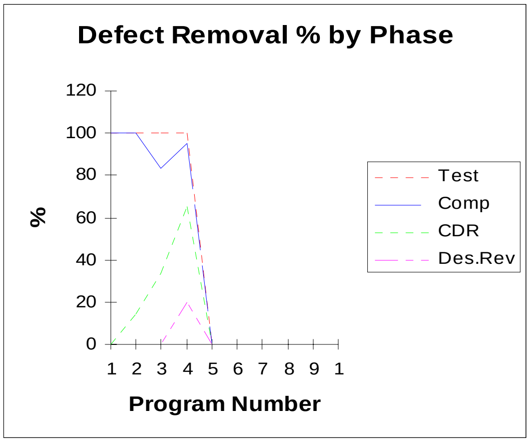


Graph 1, The percentage of the defect injected on design & code level by Phase with program number, it show the defects by each programs up to first four programs . But the zero is also showing in this graph because we have total four programs. This means program (05) was not developed in this experiment. Therefore no data recorded for its defects found [61].

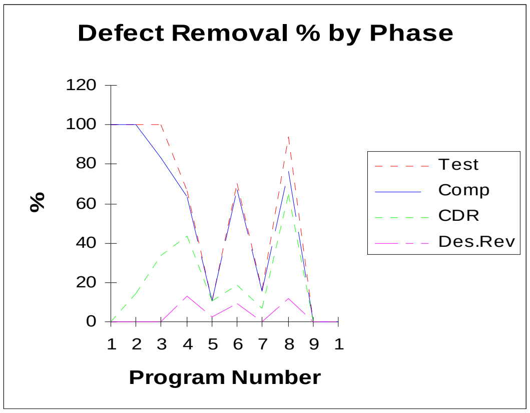


Graph 2, the percentage of the defect injected on design & code level by Phase with program number, it shows the defects by each program of all programs. But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its defect found [61].

The above graph1 and graph 2 are the result that obtained during the Software engineering project of developing the software using the Personal software process(PSP) [7,19, 61] for the aim of improving the quality and performance of the software [4, 8, 42,46], also the graphs shows how the defects was injected in coding and design phase[3,19] , The above graphs shows that many defects injected in code , and the percentage of defects injected through the design phase is not some much, therefore this shows that we spend a lot of time in coding and this indicates that the early phase of the system development where not taken in serious that leads to many number of bugs or defect during the coding phase therefore this graphs as result obtained concludes that we did not spent a lot of time for the design review and software inspection that is why we spend a lot of time during the software coupling and many defects injected during the compiling of the software code



Graph 3, the percentage of the defect removed on test, compile, code review & design review level by Phase with program number, it show the defects by each programs up to first four programs .But the zero is also showing in this graph because we have total four programs. This means program (05) was not developed in this experiment. Therefore no data recorded for its defects found [61].



Graph 4, the percentage of the defect removed on test, compile, code review & design review level by Phase with program number, it shows the defects by each program of all programs .But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its defects found [61].

Graph 3 and 4 above are the results obtained during the software design review , code review testing and compiling of the software program when experiment where conducted . And the results shows that many defects where injected during the testing phase of the software program and compiling phase of software program . Graphs 3 and 4 above show that we did not take serious measures during the program design review phase and the program code review phase as the early processes of the software verification and validation processes that is why many defects was found during the software testing phase and the software code compiling phase .

5. Discussion

It is true that there is a need of software quality among the software system that developed by software development organizations or companies. That is to say they need to develop the software system that meet the customer or clients needs according to the market demands and time [8, 17]. Many software process improvement approaches have been investigated and suggested by the software Engineering institute(SEI)[19,20, 33] to help software companies to improve their processes in order to increase the quality of the software products[19,33].

The verification and validation processes have positive advantages on helping these software processes through detecting and avoiding the bugs or defects introduced in the software products [10,56]. Verification and validation processes have number of applications that improve the software process and detecting bugs injected and remove from the software system for the aim of improving the quality of the software quality [37, 40]. Basing on the domain of this thesis as the software engineering area the results shows that verification and validation that applied at the early stages of the software development processes are the one that have the great advantages to make software projects succeed and at the end provide high quality software project [17, 46]. The use of the verification and validation techniques is very important and this thesis suggest the use of the verification and validation techniques as soon as the development of software system starts from the software requirement gathering phase [1,19,56]. Doing so will avoid the consequence of the software project failure. This theory of using the verification and verification techniques at the early stages of the software processes is very important factor especially during the development of the software critical, socio-technical and Real Time system [2, 59] .Doing so help a lot to avoid the failure of those systems that are very important software system applications and applied to many application like in economical, social and government sectors. It is already proved that if these system are not designed carefully they can lead to consequence to the society if one part fails to operate as required this means they can cause the physical damage , economical loss or loss of life[2,59].

Therefore the use of the verification and validation techniques that applied at early stage of the software development like software inspections formal specifications and design reviews will be the key of success to these systems. The progress of these system will be shown as soon as possible and the customer or end use we have high confidence to these system s . Many software engineering project failure is not because of the poor knowledge of the software engineers or the software developer .This is because of the poor software process improvement and the use of the verification and validation techniques at the reasonable time[1,19] .

The aim of developing the software system is not just for the software development companies or developer only. But this involves the consideration of the markets demands and the customer needs at the reasonable time [24]. This is to say the quality of the software products is very important and also the time management is the key factor for making sure that the higher quality software products are derived to the end user within short time[2,40,42,46] . Domain of this thesis is software engineering and is addressing how the organization or software developing companies will be able to have higher software products and also those products have to be delivered to the customer within the short time with the support of the verification and validation techniques to the software process improvement[19, 37,56] . This is the research question presented in this thesis the answer to this research question is that the early the better .This means that the verification and validation techniques that applied at the early phase of the software development are the ones shows the progress of software projects at the early phase of the software development[1,10,15,19,53,56] .This is to say the visibility of the software projects are observed from the early phase of the software development and it is easy to improve the software process once defects are obtained at the early phase of the software development. Research presented in this tried to compare the different existing verification and validation techniques [1, 10, 13, 15]. From the different literature sources, books and the implementation that done through project study from one of the software process improvement. And the result shows that the Verification and Validation techniques once used and once the software projects verified and validated at the early phase of the software development it is easy to provide the quality software products.

When the software developed without verified and validated at the early stage of software development many fails [1, 19]. Therefore this thesis suggest the use of the verification and validation techniques for the software processes from the start of the software development .Some of techniques that applied at the early phase of the software development, includes the use of software design review , software inspection and software code review [1,2,19,439] . These techniques once applied as the verification and validation processes provide the way to have higher quality products and many defects can easily detected and prevented at the early phases of the software development[1,2,15,53]

6. Conclusion

Software development organization , software engineers and all software development companies will be able to have higher quality software only if number of defects from the software products introduced through each phase of the software development will be prevented as soon as possible during the each phase of the software development process[8,14,19 ,21,40,42] .This means that the use of the verification and validation processes should be taken seriously at the early phases of software process .This will help to prevent larger number of defects and within a short time instead of using the software testing only at the end of the software implementation[15, 19, 43].once the verification and validation process applied as soon as the software development starts will reduce the amount time spending during the software testing phase and sometimes it is not easy to detect and avoid all defects or bugs using the software testing only[19,53,56].

Therefore in order to establish the confidence of using the software system. And in order to make user to trust software products [52, 53]. It is very important to check if the system developed will meet the need of the user and the requirement specified .Doing that will avoid the consequence of the software project to fail to perform operation as required [2,9,48]. Therefore the use of the verification and validation techniques at the early stage will help a lot software organization to reach their goals and improvement of the software process[1,19] .The software process improvement can not provide higher software system products without of the support of the verification and validation techniques [19, 46, 50].This means that verification and validation techniques are focusing on defect detection , preventing and avoiding defects within the software system .Also we can say that software testing applied to many software organization but it not enough to prove that the software system contains no defects . Because the use of software testing is to show the presence of bugs [2, 19, 53, 59]. The research presented in this thesis suggest that in order to avoid larger number of burgs to the software products there is need of apply verification and validation processes as soon as possible[1, 11, 13, 15,19,49] . And doing so the higher quality software will be delivered to the customer and within short time [8, 40].

Domain of this thesis presented in this paper is software engineering area and help the software engineering companies or organization to improve the quality of the software systems. Through the use of the software verification and validation processes[1,8,6]to improve the software process and to make sure that the higher quality of the software products are delivered to the customer within the locate number of time and focus on the market demands[2,19,37].Also the end user must trust the software system that are going to use. The involvement of customers at the early stage of the software development give them confidence of what system their going to use and it is easy to trust the system delivering it from the developers[54,60].

Research presented in this thesis explains the current application of the verification and validation techniques to the software process improvement. Considering this situation we have discussed the different existing validation and verification techniques from different literature sources [1,2,10,15,56] . we have been using the comparison approach for some existing verification and validation techniques like software inspection, software code review , software testing and software design review . We have made experimentation for deriving the results of this research , this done as the project study using the person software process approach (PSP) [3,19]. And the results presented in the form of the graphs a lot of information can be analyzed from the graphs of the results obtained .

The major conclusion is that there is a need of using the verification and validation techniques in order to improve the software process and to obtain the quality software products [8, 27,53]. In this research paper we suggested to use the verification and validation that help to avoid failure of the software projects and help the software development organization to manage and see the progress of the software products that they need to develop .And as well as the early determination of the software project failure or success can be visualized as soon as the software development starts [33,51,60].

We have been discussed the use of the software metrics also we discussed how software metrics helps the software development organization to control and measure what the want to produce [22, 23, 30, 35, 38, 50].

7. Future work

Research presented in this theses in future we are going to focus on the Verification and Validation techniques that will help the software process improvement when there is change of functionality of the requirements of the software system. Through literature survey of many literature source as the methods used in this theses. We came to know that it is importance for the one Verification and Validation to be flexible due to changes of the software requirements during the software development. Therefore this research in future will address the use of Verification and Validation approaches that will solve this problem.

We will address how the adaptability of Verification and Validation should be done in order to limit the failure of the software operation when any change of the software requirement occur . It is true that many software Verification and Validation process applied in software process but some are not flexible. Therefore this Research in future will focus on how the Verification and Validation process that will be able to adapt the changes of the software functionality or requirements.

Also this research present in this thesis in future will be going to focus on how the addition of the some processes will be help the Verification and Validation techniques existing today to be flexible and adjustable when the functionality changes during the software development . This adaptation of the use of Verification and Validation activities to the software improvement that this research will going to address in future will be done according to the functionality of the system and what the software product are required to be produced . Also this will focus on the general risks that can be occur during the use of the software Verification and Validation techniques during for software development

8. Acknowledgements

I am really very thankful to my advisor Mr. Christian Ohlsson and Dr. Samantha, Jenkins examiner. Because they helped a lot for this research paper and also thankful to all the staff members of this university who were guide me in my whole stay in Sweden.

9. References

[1] "IEEE Standard of Software Verification and Validation ", IEEE std 1012-1998, March 1998

[2] I.Sommerville, "Software Engineering 7th ed", Addison – Wesley publishing company, Boston, 2004.

[3] W.Scaccchi, "Process Models in Software Engineering", in J.J marciak (ed), encyclopedia of software Engineering wiley, New York, 2003

[4] M.s Deutsch and R.R Williams, " Software Quality Engineering : A total Technical and Management Approach ", Printicel Hall international London , 1998.

[5] S.L Pfleeger , " Software Engineering Theory and Practice, (2ed)", Prentice Hall, Upper saddle River, 2001.

[6] C.N Johnson "The Benefit of PDCA", Quality Progress Vol 35, pp 120-121, 2002.

[7] W.S Humpyrey, "A Displine for Software Engineering First (ed) :", Addison- Wesley Publishing company, Reading, 1995.

[8] Daniel, Galin, "Software Quality Assurance", publishing company, Addison- Wesley, new York, 2003

[9] Hertzel, WilliamC, "The Complete Guide To Software Testing 2nd (edition):", Wellesley, MA: QED informatics science, 1998

[10] G.Gordon Schulmer, Garth R.Mackenzie, "Verification and Validation for Modern software intensive systems :" , Prenticel, Hall saddle, River, 2000

[11]Paul C.Jorgensen, "Software Testing", CRC Press, Rockford, Michigan, 2002

[12] Myers, G.j, "controlled experiment in program testing and code work through/inspections",Publishing Company John Wiles & Sun , Inc , New york , 1978

[13] Myers, G .J. "The art of Software Testing", Baltimore, USA, I979.

[14] Fagan , Michael E, " Design and Code inspections and Process Control in Developing Program :" , IBM –TR.00-73 , June , 1976

[15] Rakitin, Steven, R "Software Verification & Validation, A practical guide", Boston: Archech House, 1997

[16] M.Leszack,D.E Pery and D.stoll :, " A case Study in Root Cause Defect Analysis :", Presented at 2nd International Conferences on Software Engineering , Limeric , Ireland , 2000.

[17] Card , DavidN and Glass, Robert, " Measuring Design Quality :" , Englewood Cliffs, NJ: Prenticell hall Publishing Company , 1990

[18] Marcinisk (ed) , Encyclopedia of Software Engineering , Vol 2 , Wisey , New York 1994 .

[19] Humphrey, W, "Introduction to Personal Process Improvement", Addison- Wesley, MA, 1996

[20] "Capability Maturity Model Integration (CMMI) version 1.1, CMU /SEI, 2002- TR- 028, August 2002

[21]. Craig Larman, "Applying UML and Patterns", Prentice Hall PTR, October 30th 1997.

[22] Bache, R., Bazzana, G "Software Metrics for Product Assessment", McGraw-Hill, 1992

[23] Marchiniak, J.J.: Encyclopedia of Software Engineering. Publisher Company John-Wiley & Sons (Vol. I), 1994

[24] Clements, Szy perski, "Component Software," publishing company Addison – Wesley, New York, 2004

[25] J.Cheeseman and J. Dauds, "UML Components", publisher Addison, New York, 2004

[26] Onoma,Akira, ,etal " Regression on Testing Industrial environment :", Communication of the ACM,ACM ,New York , vol .41. No 5, May 1998, PP .81-86

[27] NASA: "Software Measurement Guidebook",. Maryland, 1995

[28] Pressman, R.S, "Software-Schock - The Danger & the Opportunity." Dorset House Publ., 1991

[29] DeMarco, T., Lister, T.: "People ware." Dorset House Publ., 1987

[30] Arthur j.j "Improving Software Quality ", An Insider's Guide to TQM. John Wiley & Sons, 1993

[31] Bache, R.; Bazzana, G, "Software Metrics for Product Assessment". McGraw-Hill, 1994

[32] Grady, R.B.: "Practical Software Metrics for Project Management and Process Improvement" Prentice Hall, 1992

[33] Humphrey WS" Managing the Software Process", publishing company - Addison-Wesley, 1990

[34] Schmauch, C.H., "ISO 9000 for Software Developers",. ASQC Quality Press, Milwaukee, 1994

[35] Ami, “(application of metrics in industry) - a quantitative approach to software management.” CSSE, London, 1993

[36] Greenberg, S “Computer-supported Cooperative Work and Groupware”, Academic Press, 1991

[37] Brinkworth, J.W.O “Software Quality Management”, Prentice-Hall, 1992

[38] Hollocker, C.P.: “Software Reviews and Audits Handbook.” , publishing company John Wiley & Sons, 1990

[39] Kan, S.H.: “Metrics and Models in Software Quality Engineering.” Publisher Addison-Wesley, 1995

[40] Arthur, L.J, “Rapid Evolutionary Development.” , publishing company John Wiley & Sons, 1992

[41] Paul, MC, Curtis B, Chrissis MB, “ The Capability Maturity Model”, publisher Addison-Wesley,Reading , 1995

[42] Shew hart WA, Deming WE, “ Statistical Method from point of view of Quality Control ,” publisher Dover ,Newyork , 1986

[43] Deutch,Michael S : “ Software Verification and Validation :” , Eng Le wood , Cliffs , NJ : Printice Hall, 1982

[44] j.B Goodenough and S.L Gerhart , “ Toward a Theory of Testing Data Selection ,”, IEEE Trans .Software Eng . SE – 1(2) , 156-173 (1995)

[45] Nasto , John “ A software Product and Maturity Model in Cross talk :” , publisher Software Technology support centre organization:, Hill, Vol 10, , No 8 , August 1997, pp 22-26

[46] Singh , Raugh , Gordon : “ Standardization of the Quality Assurances 3^{rd} (edition)”, upper . Saddle River, NJ: Prentice, Hall Inc, 1998

[46] Key of practices of the Capability Maturity Model of Software , Version 1.1, SEI- 93-TR-24, Carnegie Mellon University , Software Engineering Institute , 1993

[47] K-Beck “ Extreme Programming Explained : Embrace Change :” , publishing company Addison Wesley , Reading , 2000

[48] R.H Thayer and E.S .HINTON, “ Software Reliability –A Method that works”, Proceeding of the 1975National Computer Conference , Montrade, N.J.:,Afips Press, 1975, pp-877-883

[49] J.E Sullivan, “Measuring The Complexity of the Computer Software”, MTR 2648, Mitre Corp., Bedford Mass, 1973.

[50] Boehm, B.W.: “Software Risk Management”, IEEE Computer Society Press, 1989

[51] N.Anderson and B.Shneiderman, “ Use of Peer Rating in Evaluating Computer Program Quality :” , IFSM-TR-20, University of Maryland ,1997

[52] Myers , Glenford , “ The Art of Software Testing :”, Publishing Company John Wiles & Sun , Inc , New york , 1979

[53]E.A, young, “Human Errors in Programming:” int J. Man- Machine Stud. 6(3), 361- 376(1974) [Testing]

[54] J.Ascoly etal, “Code inspection Specification”, TR-21.630, IBM System Communication Division, Kingston, N.y, 1976

[55] Elemaaik, Mc curey J, “Modeling the like Hood of The Software Process Improvement”, 2001

[56] C.R , Magil , “ The Role of an Independent Software Validation Agency :” , Re-Port 76-1, Defence Systems Management School, Fort Belvor , va , 1976

[57] Raiston, Anothon & Really, Edwin (editors): “Encyclopedia of computer science “, New York, Van No strand Reinhold, 1993

[58] N.Whitter , “ Managing of Software Development Project : Formula for success , 2^{nd} (edition) :”, publishing company , Wiley, New York, 1995

[59] Royce, ww, “ Managing The Development Of Large Software Systems : Concepts and Techiques .”, Proceeding of the IEEE ,WESCON, August , 1970

[60] G.J Mayers, “A controlled Experiment in Program Testing and Code WorkThrough/Inspections,”, Communication .ACM,21(9) , 760-768(1978)

[61] Alphonce,Magori “ Managing Software Engineering program development Using Personal software process approach :”, Project work 2006: PPC010 at University – West ,Department of Technology, Mathematics and Computer science , March

[62] C.Robson “ Real World Research 2rd(ed) :”, Blackwall publishing company, comwall ,2002

.[63] F.J flower : “ Survey Research methods , 3^{rd} (ed) :”, sage London , 2002

[63] F:J Flower , “ survey methods (3^{rd} ed)”, sage ,London ,2002

[64] J.A MCall , “ Quality Factor : in J.J .Marcinisle (ed), encyclopedia of Software engineering :”, wiley , New york , 2003

[65]ASQ – American Society for Quality is a society, Excerpted from Nancy R. Tague's The Quality Toolbox, Second Edition, ASQ Quality Press, http://www.asq.org/learn-about-quality/project-planning-tools/overview/pdca-cycle.html, 2004, pages 390-392.

[66] Forrest Shull et.al., "What We Have Learned About Fighting Defects." Eighth IEEE Symposium on Software Metrics (METRICS '02) 2002.

10. Appendix

This appendix includes the general materials that used to support the results obtained from the research Presented in this thesis.

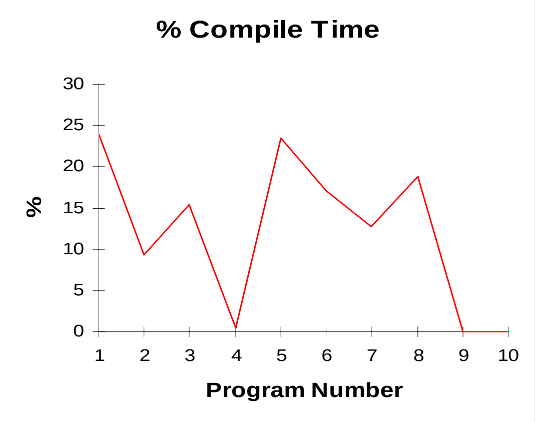
A. The experimentation part in this paper conducted by following the Personal software process approach where several software program developed using the different metrics like time metrics , design metrics , Testing metrics , Code review Metrics , Design Review Metrics and etc.

B. Experimentation done also analyze number of defects injected and Removed from each phase of the software development from software analysis to the Testing Phase and data basing on the Time used for each phase and defect where recorded by following the allocated Metrics [61].

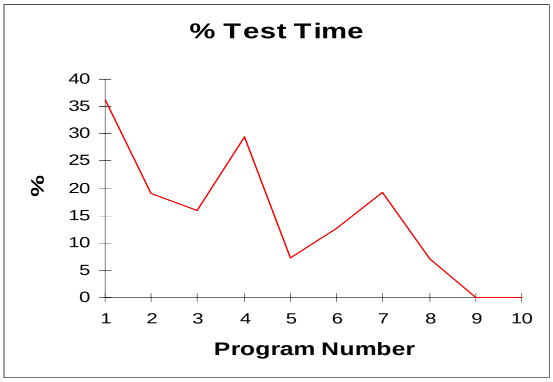
C. All data used for developing different number of the of software program where recorded and stored in form of database record and after the final development of the number of the different program. We obtained the results from the data we recorded in form the database. where analyzed in the form of the graphs and the conclusion where obtained through the use of the Verification and Validation techniques for the software process improvement.

D. Generally basing on the Research Question of this research Presented in this Thesis the experiment of the software engineering project done to provide real example of how the Verification and Validation techniques are more important for improving the quality of the software products[61].

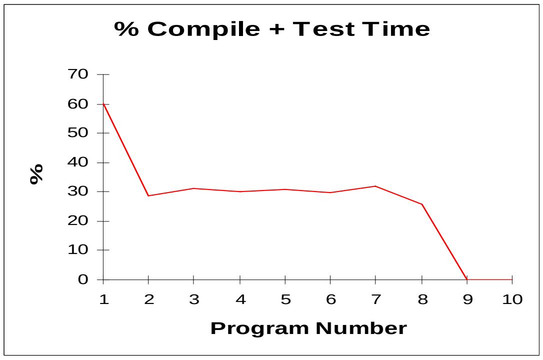
E: This appendix include the graphs that show the results of the percentage of the testing and compiling time of the software programs , percentage of the time that used to test the software program ,and the percentage of the of the planning time that used during the software program development . This appendix show the examples of the results after the final program development using the software process improvement(PSP) .This research is not a full implementation but this results from the experiment is used to support this thesis basing on the use of the Verification and Validation techniques for the software process improvement . Using the information that obtains from the graphs obtained after recording the input information in form of the database. .And through the use of the software metrics like time metrics , testing , design metrics for the software process improvement . Therefore the results presented in form of graphs of the time spend for the each phase will provide information can be useful for the improving the software process later on .Basing on the past software project . Using information's obtained from the graphs also will help the software verification and validation process when taking place for checking the software products if developed correct or not.



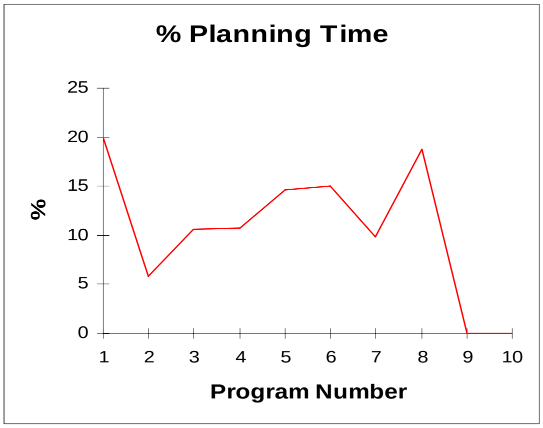
Graph 5. The above graph show percentage of time used to compile program (1-8). Using the Personal Process improvement approach (PSP) .But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its compile time [61].



Graph 6, the above graphs show the time in percentage .Used to test each of the programs (1-8] during the software program development using the personal software process improvement approach. But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its testing time [61].



Graph7, The above graphs shows the actual development percentage time of compiling and testing each of the software programs (1-8) using Personal software approach .But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its compile and test time [61].



Graphs 8, The above graphs show the percentage planning time that indicated for the development of software program(1-8) Using the Personal software Process improvement (PSP). But the zero is also showing in this graph because we have total eight programs. This means program (09) was not developed in this experiment. Therefore no data recorded for its planning time [61].

Note these above given graphs were calculated automatically through the automated excel sheets. The calculation of data is starting from planning phase to the testing phase, in the graphs. In the graphs the data is showing about the eight different programs. And also how much time taken to develop these programs, are also showing in these graphs during the planning, testing and compiling. These graphs are explaining the result that how much programmer/developer improved his/her skills. But the zero is also showing in each graph on nine number programs because we have total eight programs. This means program (09) was not developed in this experiment [61].