

---

# **Chapter 1**

## **Introduction**

---

## 1.1 Overview of Software Engineering

Software engineering is the application of computer science that deals with building of large, complex, industrial strength software systems in arrange to discover the software that cheap to run, consistent and mechanism efficiently on actual machines.

### **Software Engineering is related to computer science and systems engineering:**

- Computer Science : Concerned with theories and methods
- Software Engineering : Practical problems of producing software
- Systems engineering: Aspects of development and evolution of complex systems specifying the system, defining its overall architecture, integrating the different parts to create the finished system

Software engineering is the organized approach that aims to provide methods & procedures for analytically developing industrial potential software. The main driving forces for software engineering are the problem of scale, quality & productivity (Q&P), consistency & change. Achieving high Q & P consistently for problems whose scale may be large & large and where changes may happen continuously is the main challenge of software engineering.

The fundamental approach of software engineering to achieve to the objectives is to separate the development process from the products. Software engineering focuses on process since the quality of products developed & the productivity achieved are heavily influenced by the process used. To meet the software engineering challenges, this development process is phased process. Another key approach used in Software engineering for achieving high Quality & Productivity is to manage the process effectively & efficiently using metrics.

Software engineering procedures are that holds the methods and tools mutually .They define the order in which the methods will be theoretical ,the deliverables such as reports, manuals, forms, are required, the controls that help quality assurance and managing changes, and milestones that are used for assessing the project expansion.

---

**The regulation of software engineering provides repeatable methodologies which make software development process:**

- More closer to client
- Nearer to the technical method
- Absent from being a talent

In other words we can say that if necessities of client are same, then by following the same methods, tools and techniques, similar result will be produced.

**Software manufacturing is significant for the reason that of:**

- The business of software development and maintenance has become increasingly competitive day by day which requires money-making and high excellence products to battle in the market.
- Due to complexity of software applications failures can result in monetary smash up level the lives of individual beings. Financial, air-traffic control, transportation, and medical applications demand high-quality software.
- Due to change of customers and users attitude toward quality. Customers are more willing for get better quality and higher productivity software.
- Today each project is managed and developed differently so it is difficult to know what truly affects quality and hoe to control it.
- We need a common framework for evaluation and quality improvement.
- Standardizing the software development and evaluation processes will promote the characterization and measurement of projects so that they can be compared more easily.

The major approach of software engineering to complete to the objectives is to separate the development process from the products. Software engineering focuses on process since the quality of products developed & the productivity achieved are heavily subjective by the process used. To meet the software engineering challenges, this development process is phased process. Another key approach used in Software engineering for achieving high Q & P is to manage the process effectively & efficiently using metrics. Software Quality Engineering (SQE) is a extensive life, cycle approach concerned with every aspect of the software product development process (SDP).

---

## **1.2 AN INDUSTRY ANGLE**

Computer systems were developed using hardware-oriented management in the early days of computing. Hardware was considered as the most important item of the system. To control hardware costs thorough analysis and design, formal methods, tools and technical standards were improvement. The software work was practically unmanaged programs were constructed in an under closely controlled manner using experiment and blunder approach.

Today, software has become the single costlier thing in the system. Because of complication, software takes a very long time for development. Hence the cost of software becomes high. Also, finding all errors and measuring growth of the software project is quite complex.

### **An Aging of Software system**

Today, there are thousands of critical software based applications that have undergone a lot of modifications. They are now practically uncontrollable. Any modification made to them now may leads to the total breakdown of the system. No one has the awareness about their internal structures. Many components of such systems require re-engineering; of they will not be competitive.

Overall, there is ever-increasing need for software because of the following reasons:

- The expenditure of the hardware continues to reduce while increasing the speed and storage capacity.
- The number of users of computers continues to increase exponentially because of continuously declining hardware cost and easy availability. These results in speedy increase in the applications as a result of new way of thinking of the user and his/her varying necessities.
- The traditional software systems developed many years ago would not fit into the present cutthroat world. So, to remain competitive and stay alive in the field, re-engineering of such systems is continuously increasing with the advancements in technology.
- With the theatrical developments in the field of computer construction, neural networks, communication technology etc., there will be a repeated birth of innovative applications.

---

So, software is **EVER CHALLENGING** building block of most of the computer-based systems.

### **1.3 Fundamental Problems of Software Engineering**

Software engineering is the methodical approach to the development, operation, maintenance, and retirement of software. There are few fundamental problems that software engineering faces. The sketch 1 shows the troubles of software production.

#### **The Problem of scalability:**

An innovative difficulty of software manufacturing is the crisis of scope; growth of an extremely great organization requires a very unrelated position of methods compared in the direction of mounting a little system. In other words, the methods so as to second-hand intended for mounting undersized systems in general do not level up to bulky systems. A diverse situation of methods has to be second-hand for upward huge software. Whichever oversized mission involves the use of machinery and tactics and mission managing for extensibility.

In good deed of software project, via expertise we denote the methods, actions, and tools with the aim of extensibility. In small projects, familiar methods for progress and organization can be worn. Despite the fact that, for large projects, equally contain to be a great deal stricter.

#### **Cost, schedule and quality:**

The asking price of escalating a method is the price of the belongings second-hand for the system, which, in the case of software, is the manpower, hardware, software, as well as the other hold up capital. Usually, the manpower module is essential, when software enlargement is principally labor-intensive and the charge of the computing systems is currently literally low.

That's why; the cost of software project is calculated in vocabulary of person-months, i.e. the cost is careful to be the full amount figure of person-months used up in the project. Agenda is an imperative issue in numerous projects. Company trends are dictating that the instance to bazaar of a manufactured goods be supposed to be contemptible; so as to is, the series time

---

Beginning idea to release ought to be little. Any business with such a requirement will also require that the cycle time for building software needed by the business be small.

Any business with such a requirement will also require that the cycle time for building software needed by the business be small.

One of the key factors dynamic any production discipline is quality. We can analysis quality of a software product as having three proportions:

Product Operation of the system (correctness, reliability, efficiency, usability and integrity)

Product Transition (portability, inter operas ability and reusability)

Product Revision and maintenance (flexibility, testability and modifiability)

**The Problem of stability:**

However high quality, low cost and small set of time are the most important objectives of any mission, for an organization there is another goal: consistency. An organization concerned in software development does not just want low cost and high and good quality for a project, but it wants these constantly.



**Figure 1: Problems of Software Engineering**

## **1.4 Software Engineering Approach**

The main goal of software engineering is to provide methods and procedures for developing high and good quality software at low cost and within time and budget.

Software engineering focuses on the development process that monitors and controls the software activities like quality, consistency, scalability, user friendly, reliability reusability, maintainability, portability, testability, modularity and productivity efficiently and effectively. The development process is essential to the software engineering and it is usually done in phases to manage and organize the development process and to achieve the objectives of software engineering.

### **1.4.1 Phased Development Process**

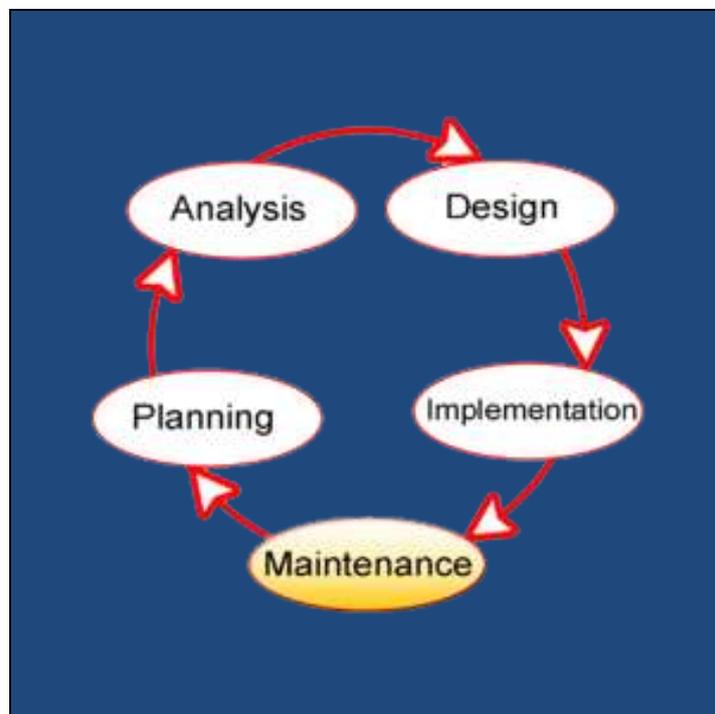
To deal with the complication improved, project tracking, quality, and to achieve reliability software development process is separated into set of phases. At the end of each phase, quality and progress checks are made without waiting till the development is complete. In general there are several models .for developing software such as waterfall model, iterative

---

model, and prototype model. An organization can follow any appropriate model; different process can have different set of activities.

In practice, any software development activity having following activities:

- Prerequisite analysis and requirement
- Blueprint and Measurement
- Code and component test
- Mixing and system test
- Release and continuation



**Figure 2: Software Development Activities**

---

## 1.4.2 Software Development Life Cycle

A application development life cycle/ expansion life series /SDLC is a rational procedure by which systems analyst, software engineers, programmers, and end-users construct in sequence systems and processor applications to resolve commerce harms and desires. We can also say it application development life cycle. The SDLC typically incorporate the subsequent general-purpose setback solving stepladder: (Figure 3 explains the perception of SDLC)

***Problem understanding and requirement specification*** - categorize the possibility and border line of the difficulty, and arrangement the enlargement approach and goals. Requirement of the imperative objects such like practical and performance requirements, extent of the product ,natural world of its crossing point to users, hardware provisions, accessible products, rapidity, rightness, the authority and expenditure of the manufactured goods and prospective reimbursement to the association etc.

**Table 1.1:** The Prerequisite specification Phase: What are the necessities of a structure?

Phase	Deliverable
Prerequisite specification	Requirements Gathering
	Assumptions
	Determine client needs
	Functional Requirements
	Non Functional Requirements
	Business Rules
	Guidelines for next phase

***Analysis*** - revision and examine the troubles, causes, and possessions. After that, categorize and study the supplies that have to be satisfied through several victorious explanations.

---

**Table 1.2:** The investigation Phase: What to manufacture?

Phase	Deliverable
Analysis	Requirements Document
	Domain Ontology
	- Things
	- Actions
	- States
	Typical Scenarios
	unusual Scenarios

**Design** – In this segment developers affecting beginning dilemma field to the resolution area.

This segment includes the subsequent attributes: Data Structure, Software Architecture, Procedural Detail and Interface Design. Design phase summarized in a Table 1.2

**Table 1.3:** The Design Phase: How to construct?

Phase	Deliverable
Design	Architecture Document
	Implementation Plan
	Critical Priority Analysis
	Performance Analysis
	Test Plan

**Development and Coding-** This stage identify the words in which the policy is to be on paper and classify any limits on use of speech features. They characterize officially permitted words structure, manner conventions, regulations for statistics structures and interfaces, and inside code certification [2].

**Table 1.4:** The Coding Phase: What is the language construction?

Phase	Deliverable
Coding	Focus for the developers
	Language structure
	Style conventions
	Rules for data structure
	Code documentation

*Quality Assurance And Testing-* The system is tested in disagreement to system requirements to see if all requirements are precise is accomplished or not .Superiority assurance is the complete procedure of breakthrough issues (testing), documenting them, verifying their fixes, and most prominently, civilizing the enlargement and testing processes thus to facilitate fewer bugs come to pass in the expectations.

**Table 1.5:** The Testing Phase: Get better Quality

Phase	Deliverable
Testing	Regression Test
	Internal Testing
	Unit Testing
	Application Testing
	Stress Testing

---

**Implementation** - Implement the solution. It defines implementation details including programming languages, platforms, programming environments, debuggers, and many more.

**Table 1.6:** The Implementation Phase: Now put together it.

Phase	Deliverable
Implementation	Code
	Critical Error Removal

**Maintenance and Support** –

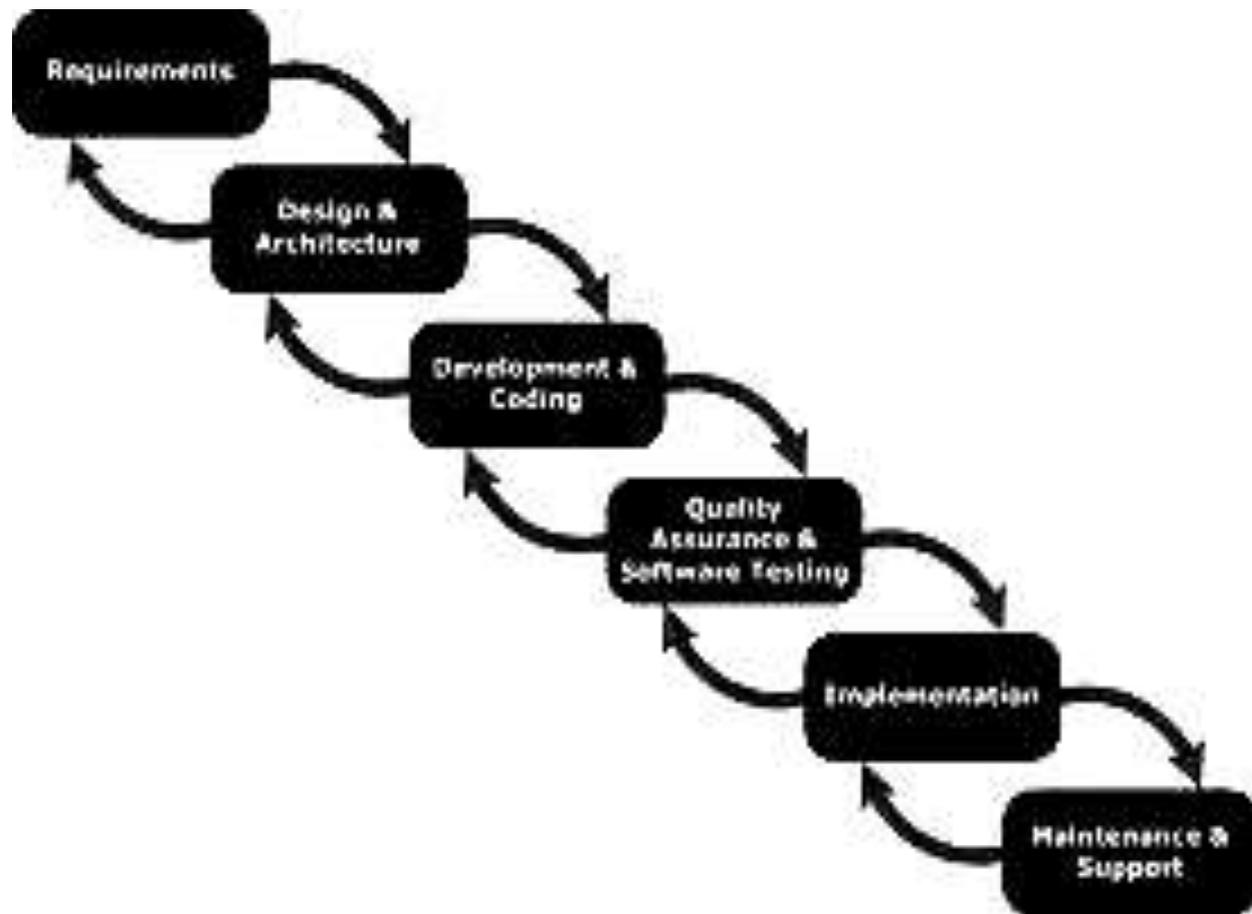
Scrutinize the implement explanation, purify the intend, and put into practice improvement in the direction of the solution. Unusual hold up situation can string reverse keen on the preceding ladder.

**Table 1.7:** The Maintenance Phase: How to scrutinize the implemented resolution?

Phase	Deliverable
Implementation	Change Management
	Lack of documentation
	Upgrade
	Enhancement
	Bug fixing

---

**Figure 3: Phased Software Process**



---

## **1.5 Principles of Software Engineering**

Principles are set of regulations; they represent the collected knowledge of many dozens of experts who have learned through experience. Software engineering principles are based on the observation of thousands of projects. This research explains the most important principles.

### **Make the Quality first objective**

Quality should be the first objective of the software. A customer always want good quality software. The quality is different for different people. To users, it must be faultless, good response time and high capacity. To developers it must be graceful design and code. For some customers, it might be satisfying all their believed or not yet real needs.

### **High Quality Software**

We can increase the quality of a product by involving the users more, simplify the design, prototyping, conducting verification at the end of each phase, and hiring the best people.

### **Give products to customers in time**

After preface study of the system, give a fast & dirty prototype that satisfies known necessities to the user, and allow them to engage with it. This leads to clear recognition of the requirements and earlier setting down of requirements. This leads to very few number of altering requirements during later stages.

### **Solve the problem**

Detecting the problem before analysis is the requirements of software project. So it should be principle of software engineering that discover all the alternatives and weigh them of costs, time and risks associated with these solutions.

### **Evaluate design alternatives**

For the proper construction of a system measurement of algorithms should be design properly. This is required to optimize the understandability of the system's external activities.

---

Different models are selected to optimize constructability, throughput, safety, modifiability, inter-opearability and functional requirements.

### **Use an appropriate process model**

There are several models for developing software like waterfall model, prototype model; spiral model etc According to the requirements of software project appropriate model should be selected.

### **Minimize distance between the developer and the user**

Software structure should be closer to the real world structure which enhances ease of maintenance of software projects. This can be achieved by OOD, JSD etc.

### **Good management**

Good management is necessary for motivates people to do their best. Project management

Manage the project internal activities such as quality planning, process definition, process monitoring etc

### **Skilled people are the key resources**

Highly expertise people with experience, talent and training are the key to success of an organization. There should be no substitute for quality while selecting employees.

### **Follow with care**

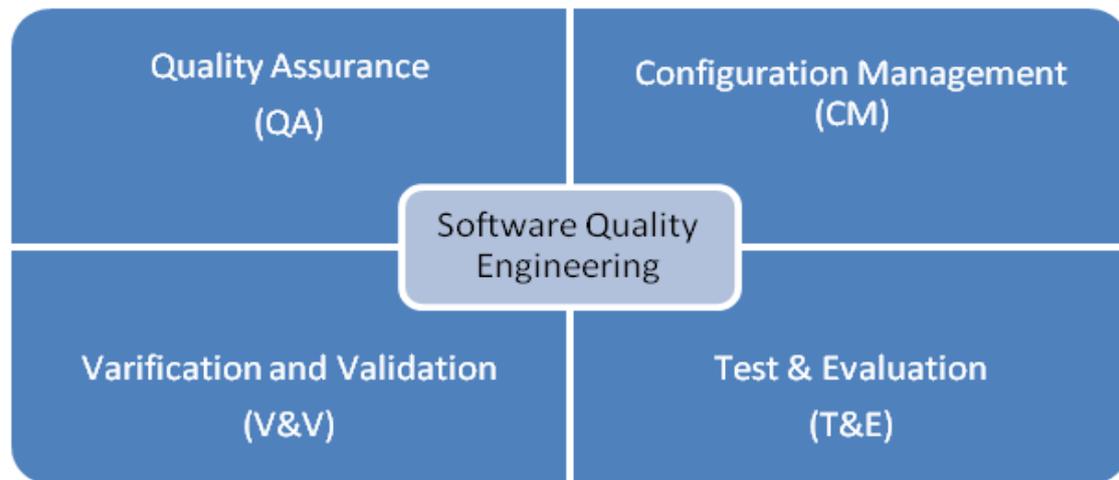
Careful assessing the work of everybody and its applicability to the present problem is most important. Different tools and advance technologies such as object oriented, modeling, reuse prototype; CASE etc. might increase quality, user satisfaction and cut down cost and time.

---

## 1.6 Overview of Software Quality Engineering

Software engineering is the organized approach that aims to provide methods & procedures for analytically developing industrial potential software. The main vigorous forces for software engineering are the trouble of scale, quality & productivity (Q&P), reliability & change. Achieving high Q & P without fail for problems whose scale may be large & large and where changes may take place continuously is the main face up to of software manufacturing.

The major approach of software engineering to complete to the objectives is to separate the development process from the products. Software engineering focuses on process since the quality of products developed & the productivity achieved are heavily subjective by the process used. To meet the software engineering challenges, this development process is phased process. Another key approach used in Software engineering for achieving high Q & P is to manage the process effectively & efficiently using metrics. Software Quality Engineering (SQE) is a extensive life, cycle approach concerned with every aspect of the software product development process (SDP). An SQE program documentation targets intended for every one constituent software expansion processes. Takes hooked on account purchaser manufactured goods necessities, purchaser superiority necessities, and company superiority contain a inclusive locate of excellence objectives quantifiable superiority attributes (quality metrics) to assess progress towards those objectives, and quantitative supplies SQE is included guarantee come close to three features of SDP: Quality, Process and Management, leading to these SQE components:



**Figure 4: Software Quality Engineering Components**

**Quality Assurance (QA):**

A quality management approach consists of individual's measures, technique, tackle and methodology practical by means of software professionals to pledge so as to software creation meets or exceeds pre-specified morality for the period of a enlargement sequence.

**Configuration Management (CM):**

Software configuration organization is a sunshade action with the intention of is practical all from side to side the software practice. CM identifies pedals, review and information modification that frequently occurs even as software is individual residential and following it has been out to a client.

**Verification & Validation (V&V):**

Verification ensures so as to each pace inside the development of structure the software delivers the accurate manufactured goods and Validation ensures so as to software life form industrial or altered satisfies practical and non -practical requirements.

---

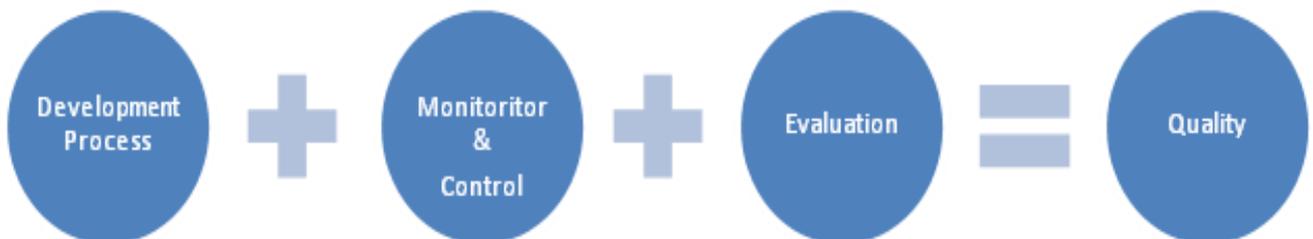
## Test & Evaluation (T&E):

### Test

A scheme or agenda calculated to attain, confirm, and create obtainable statistics to appraise, investigate, and build up; evolution in accomplishing maturity objectives; presentation and prepared budding of systems, subsystems, and workings; and utensils substance.

### Evaluation

The judgment and analysis of figures formed for the period of modern or previous test and records obtained as of test conducted next to further administration agencies and contractors, starting process and profitable understanding, or combination thereof.



**Figure 5: Software Quality Engineering Activities**

---

## 1.7 Software Quality Assurance

Software Quality Assurance is establishing in approximately every computer manufactured goods that is industrial today. Every software development company develops their individual system of standards and procedures to follow using the IEEE ISO 9000 guiding principle. They use this system to help get better the development of their software system as a complete, as reviews are made throughout and at the end of each step in the software development process.

.According to the **Jones (1985)** quality as “the absence of defects that would make software either stop completely or produces unacceptable results”.

Software Quality Assurance involves the complete software development process – monitoring controlling and improving the process, building sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with [1]. It is oriented to ‘prevention’ [1]. There are several principles that are followed in this area, concerning the most known and documented IEEE ISO 9000 software quality and management guidelines.

Software Quality Assurance can in addition be clear more in deepness as a future and prearranged move toward to the appraisal of the superiority of plus compliance to software product principles, processes, and events [2]. SQA includes the procedure of assuring that values and events are known and are followed all above the software gaining existence sequence. Accomplishment with agreed-upon principles and actions is evaluated all the way through process monitoring, invention costing, and audits. Software increase and have power over processes ought to control eminence promise endorsement points, anywhere an SQA costing of the manufactured goods might be complete in comparative to fitting principles [2].

Software Quality Assurance is found in almost every computer product that is developed today. Each software development company develops their own system of standards and procedures to follow using the IEEE ISO 9000 guidelines. They use this system to help improve the development of their software system as a whole, as reviews are made during

---

and at the end of each step in the software development process.

Significant attributes consist of functionality, consistency, usability, effectiveness, maintainability, in addition to portability. The yield, drawbacks, and risks of using a program can be resolute from exploratory these attribute. [19]

## **1.8 Standards and Procedures**

Each software maturity corporation has to launch principles and procedures to aid in the development of software, because these standards are the structure designed for which software evolve [2]. Standards are the documented criterion to which software goods are comparing. Procedures are the established criteria to which the development and control procedures are compared [2]. The whole role of Software Quality Assurance is to create certain so as to these principles and measures are residential moreover followed exactly in the development of software. Appropriate citations of principles and actions is compulsory while SQA behavior of method monitoring, manufactured goods appraisal and auditing rely taking place indisputable definitions to quantify mission fulfillment [2].

There are several different principles and dealings that can be implemented into a software development system; however, this paper stresses on SQA methodology which tailor the exact wants and uniqueness of a mission .In this structure, dimension is a nonspecific method in favor of important plus analyzing software process and product quality.

Obligation values status the shape and contented of how necessities within the organization will be clear. The regular standard in industry today is to begin a numbering arrangement meant for every foremost constraint and sub-requirements. Requirement standards will also establish a system on how to write each requirement. Exactly each system uses the method of writing a short phrase to summarize a requirement and then elaborates it into more detail. Some software companies like Rational have requirement development software to aid in the organization and writing of requirements.

Propose values make out the appearance plus pleased of how blueprint documents will be industrial. They make available policy plus methods designed for translating the software necessities keen on the software blueprint and meant for on behalf of in the design

---

certification [2]. Some software companies like Rational and Microsoft encompass intend expansion software to support in the association along with development of diagrams and charts.

Code standards denote the language in which the code is to be on paper and define any boundaries on use of language features. They define legal language structures, style conventions, rules for data structures and interfaces, and internal code documentation [2]. Coding standards can be a superior fixation however moreover numerous principles determination forces competence and artistic quality to suffer. by means of methods while ‘look closely reviews’, ‘playmate checks’, and convention examination tools container give support to locate hooked on consequence values. Some examples of what good standards would be: the decline or removal of global variables, function and method sizes be supposed to be minimize; every row of code ought to be seventy font highest, single code statement for each line, and so on [2].

Documentation standards specify form and content meant for preparation, organize, and artifact certification and make available uniformity all the way through a mission [2]. Documentation is risky in a system. It could be on paper in any form, such as electronic (e.g. comments in a program), or in paper form (e.g. a manual). Each put into practice should be acknowledged visibly so it can be frequent or altered later in the development process if needed. Qualifications, designs, business policy, examination news, configurations, code changes, test tactics, check cases, virus information, consumer manuals, etc. be supposed to everyone be recognized in a number of appearance [1]. There ought to if possible be a organization in favor of with no trouble detection and obtaining information and determining what documents resolve encompass a challenging portion of information. Revolutionize supervision for records ought to be used if promising [1].

Software superiority guarantee methodology is noticeable ladder to be followed in affecting out a development. All processes are supposed to comprise usual actions. Examples of processes for which dealings are mandatory are configuration supervision, nonconformance coverage and remedial accomplishment, trying, and recognized inspections.

---

## 1.9 Importance of SQA

With improvements in technology, we have industrial well again approaches, tools & method to expand and maintain software. These changes reproduce the rising prime of life of SQA ground:

- The business of software development and maintenance has become increasingly competitive day by day which requires money-making and high excellence products to battle in the market.
- Due to complexity of software applications failures can result in monetary smash up level the lives of individual beings. Financial, air-traffic control, transportation, and medical applications demand high-quality software.
- Due to change of customers and users attitude toward quality. Customers are more willing for get better quality and higher productivity software.
- Today each project is managed and developed differently so it is difficult to know what truly affects quality and hoe to control it.
- We need a common framework for evaluation and quality improvement.
- Standardizing the software development and evaluation processes will promote the characterization and measurement of projects so that they can be compared more easily.

---

## 1.10 SQA activities

This segment focuses on SQA actions in the direction of performed by a self-determining SQA. It should be significant to be dutiful of that SQA is not accountable for superiority. The accountability for excellence lies on the mission itself. The function of SQA is to supervise the approach the assignment performs its tasks.

There are three tools for the SQA:

- Reviews: - inspection of project documents and taking part in project reviews to discover quality troubles.
- Audits: - considered and spontaneous audits to check obedience to working procedures and to verify project growth.
- Measurements: - collect and analyzed data to follow quality and assess impact of methodological and procedural changes.

SQA group performed the following activities [3]:

- Assessment growth and excellence plans for wholeness.
- Contribute as mediator in blueprint and code inspections.
- Evaluation of test strategy.
- Appraisal a taster of test consequences to decide observance to plans.

---

## **Chapter 2**

### **Existing Tools and methodologies for Software Testing**

---

As software systems become increasingly complex, they become larger sources of risk in a project. The purpose of all assurance tools is to reduce arising from quality uncertainty. Manual methods of SA contribute to the risk via human error and lack of coverage in assuring a system, among others. SA methodology can provide increased efficiency, through automation as well as increased effectiveness, by showing that assurance activities provide more confidence in the software system. Methodologies solve the troubles relating to the expansion & progression of SQA program for small as well as large projects. In the future SQA methods will be vital in favor of every software projects for the reason that software plays a very important role in the communications of contemporary civilization; a software stop working cause significant problems.

- For effectiveness to reduce risk
- To increase efficiency to improve quality.

For the achievement for getting a reputable place in worldwide marketplace in software engineering, a company must have to produce better and high quality products. In today's competitive environment, single cannot stand correcting errors after deploying the products to the customers, so we need SQA policies, programs, methodologies, templates, standards, tools and techniques and software development environments for the success of a project.

Improving software quality requires several steps:

- A strong commitment to a product's quality.
- A strategy to achieve that quality.
- An evaluation methodology of quality throughout development.
- Quality achieved in final products.
- Analyze results
- Learning and feedback.

---

**Table 2.1**  
**Steps to improve Quality**

<b>Steps to be taken</b>	<b>Benefits</b>
Proper Management	SQA programs
Specifications of goals	Clear goals understanding
Strategy to achieve quality	Proper utilization of resources
Evaluate quality	Identifications of failure and problem area
Quality Assessment	How project satisfied its goals
Analyze results	Identify areas to improve
Learning and Feedback	Learned lesson

The major function of SQA is in the direction of maintain the excellence of the software products consequently so it is to make sure that the standards and procedures are properly followed. Software Quality Assurance methodologies are residential to facilitate organizations to accomplish elevated superiority software.

Consideration of product is not sufficient in SQA; we must consider the development process. With the help practical methods we are trying to explain if high quality productivity is used, productivity increases automatically. The representation stress the significance of sorting out tasks intended for development and SQA .Explaining quality requirements, planning quality control and performing evaluation quality control should be conducted by independent personnel. By undertaking appropriate constructive actions we can improve quality of products .Constructive actions to meet the desired results for the projects so that the corporation can get better the excellence of prospect work. We suggest that to meet future quality demands software engineers must be able to take better and more informed decisions. For fulfilled this software developer's need a framework for consistent and uniform software development activities like development life cycle model, milestones, documentation, planning and control, configuration management and reviews and audits.

The paper contains significant insights on the measurement and analysis of software quality during software development process. In this article we want to explain that without clear understanding and careful statement of goals evaluation of software quality is not possible.

---

Based on previous experiences we are annoying to assess and get better the excellence in a number of organization, we encompass finished that an extent and investigation program that extends through the entire life cycle is a must. Such a program requires a quality oriented methodology that has five essential aspects: SQA goal definition, SQA Program Development, SQA analysis, SQA program evaluation, and feedback.

## **2.1 Requirements of Software Quality Assurance Methodologies**

Software Quality Assurance methodology or system expansion methodology in software manufacturing is a structure that is second-hand to configuration, sketch, and organize the progression of mounting software projects. SQA methodologies are momentous division to any expansion project. Software applications encompass move toward to participate an extra dazzling and imperative character in day to day behavior of together the industry and confidential globe. The amplify of magnitude have pilot to several innovative maturity methodologies. It would be problematical in favor of any IT specialized to utterly be aware of all the enlargement methodologies on the other hand he or she ought to have an thoughtful of the leading ones. This would comprise considerate what the methodology entails as well as the strengths and weaknesses linked among it. With an accepting of the main methodologies reachable, and IT qualified will have an enhanced talent to be relevant unusual methodologies to diverse projects based on which will work most successfully and professionally. This research paper concerned with productivity, process control and product quality.

Additional strengths of an SQA methodology comprise however are not partial to:

- Customers' desires should meet
- Better and advanced liveliness
- Sensible and practical client expectations and prospect
- Motivated and provoked Development Team
- Productive Development Team
- Polished and refined Processes
- First-class superiority Software
- Improving and civilizing estimates
- Taking place occasion & financial statement
- Near the beginning notice of troubles and Bugs
- Resulting Milestones
- Supervision visibility

---

## 2.2 Profit and Purposes of excellence guarantee and Testing

Due to speedy modernism and opposition, software organization be obliged to be equipped to extend sky-scraping and first-rate excellence software. According to **Alfonso Fuggetta ,” the eminence of a software invention deeply depends lying on the citizens ,organization and measures used to fashion and distribute it”**

## 2.3 Benefits and Purposes of Quality Assurance and Testing

Quality assurance (QA) is a developmental process enables a software project and relevant parties to:

- Originate, elucidate and eloquent its assignment, dream, goal, and objectives of software projects.
- identify the unique and/or distinctive features of the organization
- receive feedback and to take action towards the problems, surroundings cooperative priority, and distribute high-quality follow;
- Recognize needs and resources;
- Develop and execute action planning;
- Monitor and evaluate the SQA activities.
- Categorize strengths and weaknesses of a SQA activities;
- Enhance the professionalism of development staff through regulation, self-regulation, and self-organized development;
- Get better the superiority of the labor, the knowledge and attainment of the intended outcomes expand a methodical, exact state of mind/frame of mind towards, plus means of looking at, planning, delivering and evaluating its work, and to do this methodically, collaboratively and collegially;
- Have a say to the continuing process of declaration and augmentation of quality in the work in addition to their coverage cycle so as to may live for review and civilizing the work;
- Improve, develop and ensure that the quality of the work undertaken is at the highest level that they be in service in effect, with are impacting on top of the value of the work undertaken.

- 
- Spot anywhere the effort in put into practice is, moreover is not, coordinated in the direction of its declared intentions;
  - enhance communication, improve morale and a sense of working towards a common aim of the best performance and operation of the work in the
  - Mount up disciplined and total certification of the effort take on, such so as to inventive associate of workers be able to be grateful for, as well, surely robust keen on, the by means of utmost effortlessness and smallest amount instance; this be able to be practical if employees earnings is a subject.

## **2.3 List of Top Testing Tools**

**The record include open source over and above approved performance testing tools.**

**There are some Performance Testing Tools**

### **1. Apache JMeter**

**Open source load testing tool:**

Fundamentally it is a Java display place relevance, chiefly well thought-out as a performance testing tool and it can in adding be built-in among the test plan. In adding together to the load test plan, you can also produce an efficient test plan. This tool has the capability to be burdened keen on a server or network so as to make sure on its presentation and examine its prepared under dissimilar state of affairs. It is of impressive employ in testing the functional performance of the resources such as Servlets, Perl Scripts and JAVA objects. Require JVM 1.4 or higher to run.

**Apache JMeter System Necessities:** It works in UNIX and Windows OS

### **2. NeoLoad**

**Load and performance testing software:**

Device designed for measuring and analyzing the presentation of the website. The presentation and the ending product can be evaluated by means of this tool and any added ladder can be taken. This helps you in improving and optimizing the performance of your web application. We can study the show of the web application by raising the traffic to the website and the performance beneath serious weight can

---

be determined. You can obtain to be acquainted with the ability of the request and the quantity of users it can grip at the similar time. This tool was industrial through a French corporation named as Netosys and it be printed in JAVA. It is easy to get to in two unusual languages; English along with French.

NeoLoad System\_Necessities\_ : Well-suited on operating systems like Microsoft windows, Linux and Solaris.

### **3. LoadRunner**

LoadRunner is one of the important HP product which be able to exist used as a performance testing tool. This can be buy as a HP product starting its HP software separation. Furthermore, it is extremely a great deal obliging in understanding and influential the performance and outcome of the method what time present is actual load. Single of the explanation elegant features of this testing tool is that, it can make and hold thousands of users at the similar instance. This tool enables you to bring together all the requisite information by admiration to the presentation and also based lying on the infrastructure. The LoadRunner comprises of unlike tools; that is, Virtual User Generator, Controller, Load Generator and Analysis.

**The Necessities of LoadRunner System are:** Microsoft Windows plus Linux

### **4. WebLOAD**

Present are several accessible performance trying tackle accessible for measuring the performance of the exact web request. WebLoad is single such instrument second-hand for load testing and stress testing. This instrument is able to use in favor of Load testing at every one internet applications such as Ajax, Adobe Flex, Oracle Forms and much more. All the way through this tool, you have the potential to calculate the prepared performance and also

---

its reply to the users. This be industrial by the Radview Software for the **testing of Web applications**. This tool was renowned and tribute attractive software of the Radview Software. This tool is widely second-hand in the environment anywhere present is a high condition for highest Load testing. We know how to get obvious in order on the functionality in addition to the authentic capability of the net applications.

**WebLOAD System\_Necessities:** Cross Platform.

## 5. Loadster

Loadster is a desktop based complex HTTP load testing tool. The network browser is proficient to evidence the scripts which are easy to employ and outline. Through the Graphical User Interface you can modify the essential script by means of energetic variables to authenticate reply. Among administer in excess of network bandwidth you can simulate huge practical client support intended for your request strain test. Following test is operated HTML statement is generated for investigation. This tool is finest to recognize presentation bottlenecks in your submission.

**Loadster System Requirements:** Windows 7/Vista/XP

## 6. LoadImpact

This is a load testing tool which is more often than not used in the cloud-based services. This also cooperative in website optimization and improvise the functioning of any web application. This tool generates travel to the website by simulating user so as to find the pressure and greatest load it can work. This LoadImpact comprise of two key parts; the load testing tool and the page analyzer. The load testing can be divided hooked on three types such as Fixed, Ramp up and Timeout. The operational of page analyzer similar to a browser and it gives in sequence concerning the operational and figures of the website. The acknowledgment of raising this load testing tool belongs to Gatorhole AB. This is a fermium examine which means that, it can be acquired for free of charge obtainable as well as for quality price. But, you

---

encompass the profit of many options and features at what time you purchase them on behalf of finest cost.

System Requirement: Windows OS and Linux.

## **7. Rational Performance Tester**

This is a computerized presentation testing instrument which be able to be used for a web application or a server based application where there is a technique of contribution and production is worried. This tool creates a taste of the innovative operation procedure among the client and the net examine. Through the finish of it all the arithmetical in order are gathered as well as they are analyzed to amplify the usefulness. Either get away in the website or the server can be recognized and rectified directly with the support of this apparatus. This device can be the supreme option in building a effectual and mistake free cloud computing examine. This Rational Performance tester was industrial by IBM (Rational software division). They have inwards up by means of a lot of versions of this automated testing tool.

**Rational Performance Tester System Necessities:** Microsoft Windows and Linux  
AIX

## **8. Automated Ranorex Testing Tool**

The Ranorex toolset enables automating our UI testing by demo our UI events lacking the requisite to put pen to paper any code. Although, every one recording be able to contain a system folder in the background, which gives you the probability to make specially your test scripts if you desire to. In adding together, its outstanding object detection enables you to construct out the UI essentials of your purpose, in spite of whether it's a net or Windows based UI.

## **9. Testing Anywhere**

Test Anywhere is a computerized testing implement which can be successful in favor of testing the presentation of whichever net sites, web applications or any other bits

---

and pieces. Many developers and testers build employ if this implements to discover away any bottlenecks in their web request and put right them for that reason. It is a enormous implement which can check more than a few applications mechanically. The testing anywhere instrument takes on 5 straightforward ladder to engender a test. They are thing recorder, superior mesh recorder, SMART test recorder, Image detection and Editor through 385+ explanations. Originally, this testing software be industrial by San Jose based Automation Anywhere Inc. At the present time, there are more than 25000 users in support of this product.

**Testing Anywhere System Necessities:**\_ Compatible among every version of Windows OS.

## 10. OpenSTA

The full form of OpenSTA is Open System Testing Architecture which is based on GUI, it is a performance tool mostly used by application developers intended for load testing and analyzing. This is unspecified to be a difficult instrument in the middle of the all extra performance testing tools. It has confirmed potential in the earlier period and the present toolset is competent of performing the weighty load test and analyses used for the scripted HTTP and HTTPS. By the side of this position, the testing is approved away by means of the recordings and easy scripts. To hold away the test successfully, consequences and additional data are in employ all through a variety of test runs. These statistics plus fallout be able to be later export to software for create in sequence. This is a free of charge testing implement and it is dispersed below GNU GPL .This instrument be first of all residential by Cyrano, which was after that in use in excess of by Quotium.

**OpenSTA System Necessities:**\_ Opens works merely on Windows operating system.

---

## **2.4 Test Methodologies**

### **Quality Assurance Methodologies**

Software Quality Assurance Methodologies for managing quality assurance activities and optimization of its available resources effectively and efficiently within time and cost constraints.

This methodology is explained in white paper titled as “Triage Regression Testing Guide”

### **Leveraging Test Tools**

Customization of test tool methodologies to leverage automated function testing tools like Quick Test Professional and other automated testing tools. The advantages of this tool are:  
Easier to read and maintain.

Do not require any programming.

### **Effectively Deployment Staff**

This methodology focus on deployment of staff (training and educational program) on a QA projects in order to get better quality results at a reasonable cost and time.

---

## **Chapter 3**

### **Literature Review and Objectives**

---

### 3.1 Introduction

A great amount of literature exists on the subjects of Software Quality Assurance: SQA is discussed regularly in publications and conferences on software engineering such as International Conference on Software Engineering. In addition, a growing number of special issues and conferences are dedicated to distribute to reduce risk and improve quality of software, such as issues of IEEE Software furthermore the global symposium on Global Software Engineering. The articles we selected for this issue present new approaches and ideas for successful SQA and reports on the applications of some SQA techniques in industry.

#### **A Management Methodology to diminish hazard and get better excellence, Published by IEEE Computer Society December, 2007.**

In this article the authors classifies a process organization tactic that uses capacity and development models combined with risk and analysis factors to reduce risk levels to optimize quality.

The authors outline key processes that minimize the project risks in specific industry consequential in the require to get better passion processes to make the most of quality. This paper also represents a important get through in specifying and civilizing business process efficiency and increasing safety integrity.

#### **Implementing Quantitative SQA: A practical Model,Victor R.Basili and H.Dieter Rombach , University of Maryland, IEEE 1987.**

According to the author consideration of product is not sufficient in SQA, we must consider the development process. With the help a practical model they are trying to explain if high quality productivity is used, productivity increases automatically. The representation stresses the consequence of sorting out farm duties on behalf of development and SQA .Explaining quality requirements, planning quality control and performing evaluation quality control should be conducted by independent personnel. The author concludes through responsibility

---

fitting positive events to get better quality .Constructive actions to meet the desired results for the projects so that the corporation be able to get better the excellence of upcoming work.

**How to improve SQA in developing countries, Advanced computing: An International Journal, Vol.3, No.2, March 2012.**

In this paper the author points out the quality problems in developing countries, author is trying to tell us why failure occurs, here is discussed that how infrastructure, processes and documentation problems can cause lack of software quality. Inside this investigate numerous interviews are conducted to discover the answers. Some quality problems exposed via this investigate are coding issues, pattern management, security, bug reporting issues, communication problems, lack of specialists, requirement gathering, developer's attitude problem and documentation. Finally proposed solutions to manage by means of persons evils and look up the software eminence.

**The Economics of Software Quality Assurance: A Simulation –Based Case Study, IEEE, September 1988.**

The main purpose of this manuscript to conclude the tradeoffs flanked by the financial reimbursement and costs of QA. A System Dynamics Model of Software Development second-hand to spot the most favorable QA spending height along with its allocation throughout the project's lifecycle.

The model is comprehensive since it integrates the numerous functions of the software advance progression, together with equally supervision nature functions and software invention type behavior. The reproduction also captures the dynamics of inaccuracy age group as well as the QA tricks of error discovery and rectification. Investigational consequences reported in this piece of writing demonstrate so as to QA strategy has a noteworthy collision on scheme expenses.

**Quality Assurance in Future Development Environments, IEEE 1987**

---

The author suggests that to meet future quality demands software engineers must be able to take better and more informed decisions. For fulfilled this software developer's need a framework for consistent and uniform software development activities like development life cycle model, milestones, documentation, planning and control, configuration management and reviews and audits.

This paper presents four overlapping processes in the evolution of SQA: inspection, quality control, quality improvement and quality by design. Depending upon the problem, environment tools will solve the problem automatically or alert the proper manager. Future software development environment will contribute to increased software quality. They will enhance and support the existing SQA process, but they will not solve all the problems.

### **A Measurement –Based Approach for Implanting SQA & SCM Practices, IEEE 2000**

In this paper, the author examines two software process areas, Software superiority declaration (SQA) and Software Configuration Management (SCM). A measurement –based approach intended for specifying, monitoring and evaluating the software superiority for the duration of expansion whose principle is to give surety the release of invention according to the quality requirements defined initially. In this editorial author nearby a suggestion for by means of SQUID (Software Quality In the Development Process). Author show that SQUID is flexible for managing the excellence of implanting a situate of detailed tackle for SQA and SCM inside software organizations.

### **Software Standards IEEE 730: A guide to writing successful SQA plans**

According to author creating a Software Quality Assurance plan is more important than following standards. The most important factor knows the audiences that will be affected by the SQA plan, and coordinating with them as the plan and coordinating with them as the plan is being written. In this article there is examine the needs of audiences, corporate execute managers, project managers and the engineering or development staff. The creator of the SQA plan must consider the executive manager's need to distribute the costs and benefits of the work over many projects. Project managers focus on getting their current project finished and its product out the door. The engineering staff must be convinced that the SQA plan will result in a high quality product.

---

### **Tailoring SQA to fit your own life cycle, IEEE March, 1988**

The author presentation contains significant insights on the measurement and analysis of software quality during software development process. In this article author said that without clear understanding and careful statement of goals evaluation of software quality is not possible. Based on previous experiences they are trying to evaluate and improve the quality in several organization, they have concluded that a measurement and analysis program that extends through the entire life cycle is a must. Such a program requires a quality model that has three essential aspects: Define & Planning activities, Data Analysis & Learning and Feedback.

### **Enhancing imperfection Tracking Systems to make easy Software Quality Improvement, IEEE**

The author worked with the company to get better their imperfection Tracking Systems (DTSs) — with a view to facilitate SQA and SPI. Author paying attention mostly on moreover revising the principles of accessible imperfection categorization attributes in an accessible DTS or introduces innovative attributes. They required bringing in project managers and developers added in improvement, related, straightforward and effortless to use study flaw facts for assess software superiority and verdict impending SPI trial in cost useful approach. The revision also reveals countless boundaries with the intention of normally decrease the reported data's superiority.

### **In the direction of constant procedure growth: scheming SQA Utility for Large Organisation Needs, 1998, IEEE**

The author presents a definition and goals for SQA also shares experiences of scheming an SQA purpose to permanent Switching unit of Nokia Telecommunications. Author outlines that scheming and implement an SQA purpose in a huge association is a complex task that requires careful planning, capable resources and a powerful commitment. The merely people being talented to get better superiority of the products are the single who are actually

---

producing them. So it is very important to bind these people as tightly as possible to design and develop the SQA process.

### **Software Quality Assurance –Concepts and Misconceptions, 1998 IEEE**

Author explain the Software Quality Assurance (SQA) solution development field in the Capability Maturity Model (CMM) is explained i.e. activities for monitoring and controlling process activities, this paper eliminate the misconceptions by comparing SQA concept with quality management and the differences and similarities. Author shared some experiences beginning the completion of software concepts in two different software companies (start-big and start small). According to the author SQA is suggested to be performing in the undersized approach because of better benefits as compare to big start. Start with limited resources, quick start up time and allow the purpose grows as the desires and capability of the organization grow.

### **A systematic Literature Review, IEEE 2012**

According to the author by Software Process Improvement (SPI) approach we can boost the competence and helpfulness of a software maturity institute and to augment software products. The endeavor of this paper is to classify and exemplify valuation strategies and capacity used to evaluate the collision of dissimilar SPI initiatives. In this article, there is a methodical writing evaluation include 148 documents in print among 1991 and 2008. The author highlight the benefits the benefits of SPI .The aim of software process improvement is to increase product quality, to reduce time-to market and production costs.

### **Studying the Impact of Software growth uniqueness on development Goals: A informal reproduction, the Open Software Engineering Journal, 2010**

The authors discuss the problem of Global Software Development (GSD) due to various factors like limited resources, cost, language differences, physical distance, cultural difference, local government, problems in coordination, communication and control, lack of trust, productivity, security etc. This paper presents literature review, aimed goal and design of the study and identifying failure factors in GSD which resulted in a casual model .The

---

model contains many well-known factors and influences which provide information to researchers and practitioners for identification of impacts of factors on a global projects or distributed projects.

### **Related Work:**

Models related to Software Quality Assurance are discussed below:

#### **PARNAS MODEL**

Parnas[5] explained the significance of scrutiny in dipping superiority harms in SQA. Leading of all be explained the condition of check in SQA to lessen dilemma, how it helps to discover errors in the software. The leader methodologies after that to the backside the scrutiny activity are extrication and overcome.

The investigate moreover peak out that scrutiny is a time uncontrollable observe though it assure to find errors in the process.

Many proceeds are the termination of this investigates; he explained with the intention of present is no administrator want for the leadership of check. A software engineer does necessitate a good schooling so present does no need documentation for the inspection. A software engineer does not in actual fact need qualifications for the inspection. An assessment does need a correct preparation so there is need for a corporation to expend it. So it will provide many proceeds in spite of of overwhelming besides large amount material goods of a group. The explanation reimbursement of the examination is to payback errors in the starting place code but it can also facilitate to find many ambiguities in the development stage similar to it can just reveal that good strategy are followed or not like commenting etc. creator thought so as to it is not essential to do examination at the finish of the scheme but it can be done at any point of the mission and make known the errors from the basis code.

#### **David Model**

David [17] explained in their research about the free software quality and factors affecting them. He explained that the quality of the free software is higher than the other projects. Many reasons behind the improved quality are explained in this research and some

---

comparisons are done between free projects and other projects. The quality of this free software is high because of the open development models used in the development process.

### **Peer Reviews Model**

Peer reviews acting very imperative responsibility to improve the eminence [18] of these projects, user contribution is also very important for the response and on the origin of this opinion software superiority can be enhanced especially with no trouble. Client gives their criticism on the origin of their understanding and this disapproval knows how to facilitate to pick up the superiority of the software.



**Figure 6: Main Objectives of SQA**

---

### 3.2 Conclusion and Objectives

With the help of all papers we conclude that the considering the product is not sufficient in SQA –we must also consider the development process. The effective development & maintenance of high-quality software is increasingly important due to cost-effective development & maintenance of high quality software is increasing day by day. Thus, we conclude the following points:

1. Development of SQA programs so that SQA activities can be executed systematically and effectively.
2. Productivity increases automatically if a high-quality development process is used.
3. We must consider the development process in SQA.
4. Measurement –based methodology is adopted designed for specifying, monitoring and evaluating the software invention eminence for the period of development of a software project.
5. Defect tracking system facilitates Software Quality Improvement.
6. Meta SQAP manuals would be very useful to software group leaders & for evaluating the software.
7. SQA is a process that must be measure and continuously improved.

---

## **Chapter 4**

### **Problem Formulation and Analysis**

---

In our dissertation, we design SQA Meta Methodology, named **SQOM Software Quality Oriented Methodology**. At the completion of this thesis we will be able to:

- **Describe goals of the SQA Program:** The main objective of SQAP to develop & set into carry out an SQA agenda which is most favorable for the project.
- **Critical issues of the SQA software:** Identify the goals of a software project, lack in domain knowledge, and lack of skilled professionals, develop & set into apply an SQA agenda which is best for the organic projects, sufficient information of software technology, vital moment & attempt for preparation & organization.
- **Identify the responsibility of management activities.** Product evaluation and process monitoring are the SQA activities that promise the software development and control processes. Described in the project's management plan are correctly carried out and that the project's procedures and standards are followed.
- **Improving the software Quality by Management Actions:** SQA administration behavior run an SQA agenda which defines the procedures & data for the project SQA activities. Activities involved SQA goal definition, SQA program development, SQA program operation, SQA program evaluation & feedback.
- **Project SQA behavior:** The development SQA actions are clear as analysis, configuration running, and testing & complexity supervision.
- **Verify and revise quality assurance standards as needed:** Verify & revise worth manages practice as required is a measurement of mainly flourishing production ventures. Maintaining tall excellence direct principles be able to help progress corporation standing and create the place of work safer and additional agreeable for human resources
- **Improving the software Quality by Integrated Environment:**

Software Development Environment integrates software tools and methodologies appear to the user as a single, organized system of hardware and software that supports the entire life cycle. In this type of environment tools work together synergistically and make the alteration

---

smooth between phases. The environment has a single operational manner, as well as shared data, communication media and interfaces.

- **SQA Tools and Methodology:** SQA consists of methodologies and techniques of assessing the software expansion processes and methods, tackle, and technology used to warranty the excellence of the urbanized software. SQA is typically achieved from side to side the employ of specific regular practices, together with tools and processes, meant for excellence organize to promise the honesty and trustworthiness of software. This strange concern serves as a display place for researchers and practitioners to nearby theory, fallout, familiarity, and additional advances within SQA.

#### 4.1 Introduction

Any software organization can only acquire a reputable position in worldwide marketplace focus on quality. Software Quality Assurance (SQA) methodology is an important factor to every software development, designed to guarantee that quality & productivity requirement are fulfilled. We put this special issues together to help people understand the increasing importance of SQA methodology as an essential part of software projects, outline some new ideas (Integrated SQA) and proposes a measurement based methodology for specifying, monitoring & evaluating the software product quality .We are proposing a new Software Quality Oriented Methodology (SQOM), which is flexible to integrate new changes in the software industry and provides detailed guidelines and templates for real world implementation and customization. In this paper, we characterize the software development process, the product & development methodology and goals for quantification.

There are number large of methodology have launched that can be used to assess the quality of software systems during development. These includes (1) modeling & investigation methodology such as model checkers, theorem proves & code analyzers,(2) quantities methodology such as software reliability growth methodologies & test coverage analyzers,(3) traceability analysis methodology,& (4) methodology for assessing product & process

---

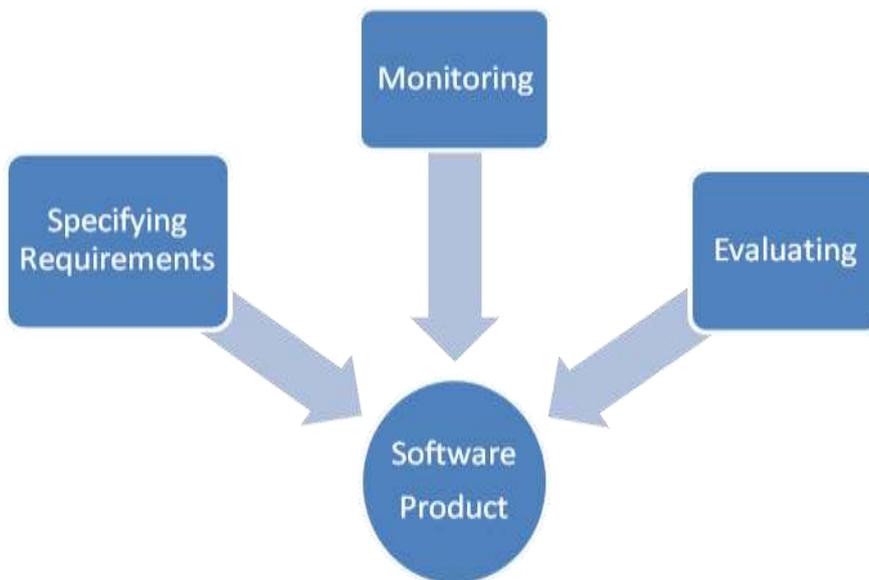
agreement to standards. There appears to be considerable interest in assurance tool development and assessment research. We have originated many organizations that determine software projects, but most of these measurement programs are ineffective.

Evaluating SA methodology is just as important as using the methodology themselves. Without a common basis for evaluation, quality enhancement will happen slowly and unsystematically. Common environment integrate software tools and methods architecturally and include common and consistent user interface and a project database. Method evaluation is essential to knowing which is the best tool for a given assurance task. Evaluation was not possible without understanding and careful statement of goals .so we next several years trying to understand the NASA software development process the product and development methods and goal of a project. Our come within reach of has been upward a position of generic methodology, such like guides templates, testing & benefits for SQA.

We are trying to develop a methodology so as to dimension to state, observe and appraise the superiority of a software product during the development process. We show that our methodology is flexible enough to be used for, managing the quality of producing any product. This article is organized as follows: First we explain the concept of Meta methodology; next we tell SQA activities and finally we proposed a new Software Quality Oriented Methodology to evaluate and improve the quality in several organizations. The Meta SQOM is evaluated by Ranorex testing tool for checking whether the project is success or not is determined by the extent to which the actual results fulfill the targets for product quality. The object should be SQ methodology is used to create quality control systems.



**Figure 7: Quality Control Systems**

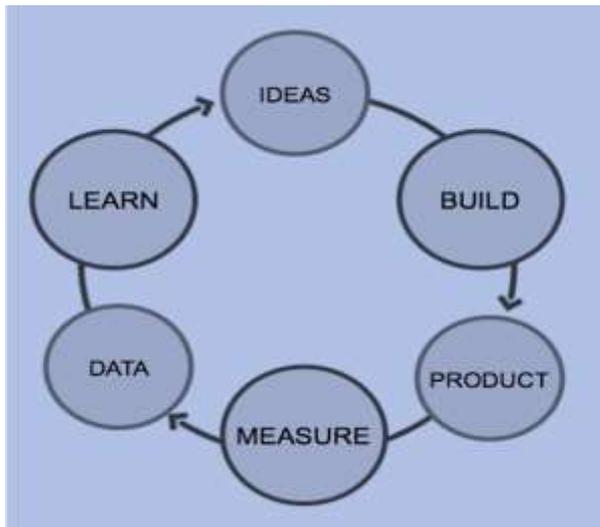


**Figure 8: Software Quality Oriented Methodology**

---

## 4.2 Meta-Methodology Concepts

Meta-methodology should be developed and placed on performance consequently so as to software superiority guarantee actions execute systematically and effectively in software projects. Meta Methodology provides guidelines and information to the managers and group leaders that how to plan, build, collect data, measurement and analysis and learning of SQA activities.



**Figure 9: Concept of Cyclic Meta Methodology for SQA**

**Ideas:**

How ideas work for arrangement, calculation and be in command of, at what time and how the information wanted for compute all actions must be composed, what methods and tackle be supposed to be second-hand.

**Build:**

According to the requirements build the system. The reason of this segment is to discover the most excellent explanation toward the problems.

---

**Product:**

Any modification required by the user during initial operating period is usually done in this period.

**Measure:**

Performing the evaluation of tools and methods is the meaning of measure. The result of the costing is recorded & reported as criticism for expectations projects.

**Data:**

Data collection is necessary to measure quality and help management understand the project. When data is collected, the process is performed manually.

**Learn:**

If the consequences of estimation for an SQA curriculum be fine, the feedback data is referred to during the development of a new SQA program. The fallout of the examination & the understanding segment can be nourish reverse to the institute to alter the method it do commerce based on openly strong-minded successes & failures.

### **4.3 Meta Methodology SQA activities (Recommended Methodology)**

Meta methodology is based upon three types of activities:

**SQA Project activities:**

These types of actions are explained through project members for the duration of the growth and preservation of a mission such as review, configuration management, testing and problem management etc.

---

### **SQA Management activities:**

These types of behavior are approved by a scheme head. Define goals, plan, build, execution, verification, learning and achievement concerning the scheme SQA actions are operated in series. This includes development, organization, establishment of rules & regulations, definition of principals and project evaluation which are usually done by division manager.

### **SQA Integrated activities:**

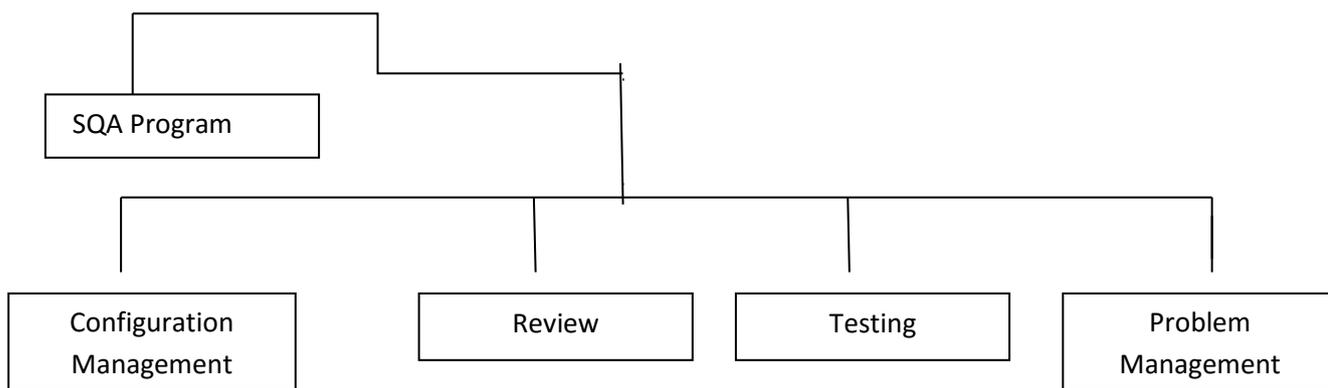
Software Development Environment integrates software tools and methodologies appear to the user as a single, organized system of hardware and software that supports the entire life cycle. In this type of environment tools work together synergistically and make the alteration smooth between phases. The environment has a single operational manner, as well as shared data, communication media and interfaces.

### **4.3.1 Project SQA Activities**

Project SQA is defined to be performed by the following 4 categories of activities in Meta-SQAP. Meta-SQAP provides an association and measures intended for projects SQA activities. The measures are primarily clear by IEEE standards.

1. Review: - This grouping of actions includes propose evaluate, walk-through, convention scrutiny and authentication. Arrangement is reviewed by members of the assignment numerous periods in every segment. Every software maturity segment is well planned finished following confirmation.
2. Configuration management:- The core behavior in this group are configuration recognition and pattern organize.
3. Testing:- This group of actions includes the growth of well-planned certification for testing, testing statistics growth based on testing technologies and tools, and the scrutiny of test consequence, implementation the test segment.

- 
4. Problem management:- This group of behavior includes complexity account supervision, completion of countermeasures meant for individuals difficulties, and testing to stop the reoccurrence of difficulties.



**Figure 10: Standard SQA Program**

### **4.3.2 Project Management SQA activities**

It is necessary to set up processes, procedures, techniques, tools and clear methodologies to reduce complication of software. Doing so will guarantee that the end-product has been exposed to determined and accurate industry-wide testing techniques and procedures. This will translate into a high scale of assurance that a software system passes the test for correctness and reliability. SQOM (Software Quality Oriented Methodology) is engineering the same as a meta-methodology which sort out the problems concerning the development & operation of SQA programs so that SQA activities can be successfully & methodically executed in software projects. The SQOM is successfully operated via the SQA administration sequence actions, consisting of SQA goal definition, SQA program development, SQA program operation, SQA program evaluation and feedback.

A suitable methodology should contain the entire system's development life cycle. Most current methodologies combine the use of several development tools and techniques. A meta-

---

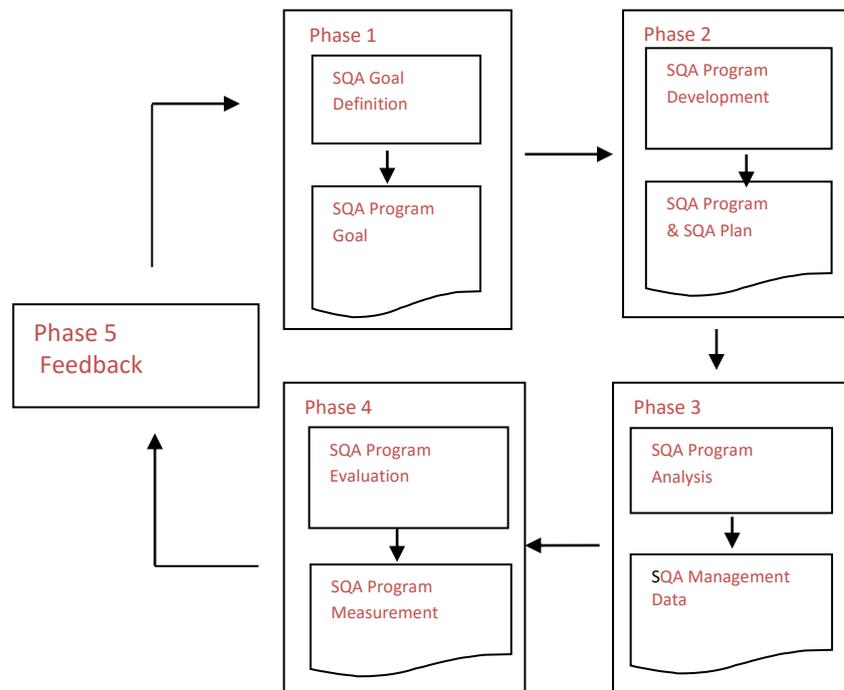
methodology intended for software superiority declaration have be industrial and set into carry out so that software quality assurance behavior can be efficiently and methodically executed in software projects. Meta-SQOM provides strategy and in order with reference to the three levels of SQA actions: the assignment SQA, the SQA managing rotation, and the included SQA movement levels. In Meta- SQOM (software quality assurance program), scheme SQA behavior, such seeing that review, configuration control, and testing, are healthy well thought-out and intended away as a SQAP. The SQAP is efficiently operated via the SQA supervision series behavior, consisting of SQA goal definition, SQA program assessment, and criticism. Every part of the projects are support by the separation SQA behavior in each sharing, such as decision-making renovation and in order control in the middle of projects.

- **SQA Goal Definition:** Internal (skills & experience of the project staff) & external (scalability of a system, user requirements etc) factors should be measured in setting SQA goal. There are two types of quantitative goals are recognized.
  1. Activity Quality: - SQA activities should be involved as the basis for SQA goal definition. These activities are evaluated according to goals of SQA.
  2. Product Quality:- Goals about product quality are set as the basis for analyzing the results of SQA activities.
  
- **SQA agenda expansion:** The SQA curriculum urbanized in this segment is premeditated to bring about the goals which were situate in the earlier segment through planning, designing, & coordinating. The SQA program development is having three steps:
  1. Planning: Planning includes defining what criterion is experimental in conditions of not direct events, how these indirect measures can be used for prediction & control, when & how the data needed for computing all measures should be collected and methods & tools should be used. Selecting appropriate indirect measures requires that we have knowledge of the project's particular development or maintenance process.

- 
2. Designing: In designing the SQA activities operation rules & management rules are defined. For this we need appropriate rules, procedures, technologies and tools are considered for developing software projects.
- Coordinating: Coordination with the user and external sources is essential for proper execution and operation of a successful SQA program.
  - **SQA Program Analysis:** On the basis of detailed plan SQA activities & SQA program is reviewed & modified at of development phases. During & after the project we must conduct data analysis. The information should be disseminated to the responsible organization. The operational definitions of quality provide traceability from goals to metrics and back. This lets you interpret the measurement in context, ensuring a focused, simpler analysis. The goal driven operational measures provide a framework for the kind of analysis we need.
  - **SQA Program Evaluation & Measurement:** The SQA programs are evaluated in this phase; the result of the assessment is recorded & reported as response for prospect projects. This phase includes two activities 1) measurement, in which the methods and techniques specified during the planning phase are applied to gather the actual values for all defined direct & indirect measures & distributions and 2) Evaluation: In which direct measures are compared to the quality requirements & indirect measurements are interpreted to explain or predict the values of direct measures. Evaluation also involves deciding if the requirements were met for each quality characteristics & for each quality characteristics and for entire set of project requirements.
  - **Feedback:** If the consequences of assessment used for an SQA curriculum were first-class, the response of data is referred to during the development of a new SQA program. The fallout of the scrutiny & the elucidation part is able to be nourishing flipside to the association to transform the technique it do production based on openly resolute successes & failures. On behalf of, considerate so as to we allow faults of oversight bypass from end to end the check up process and be

---

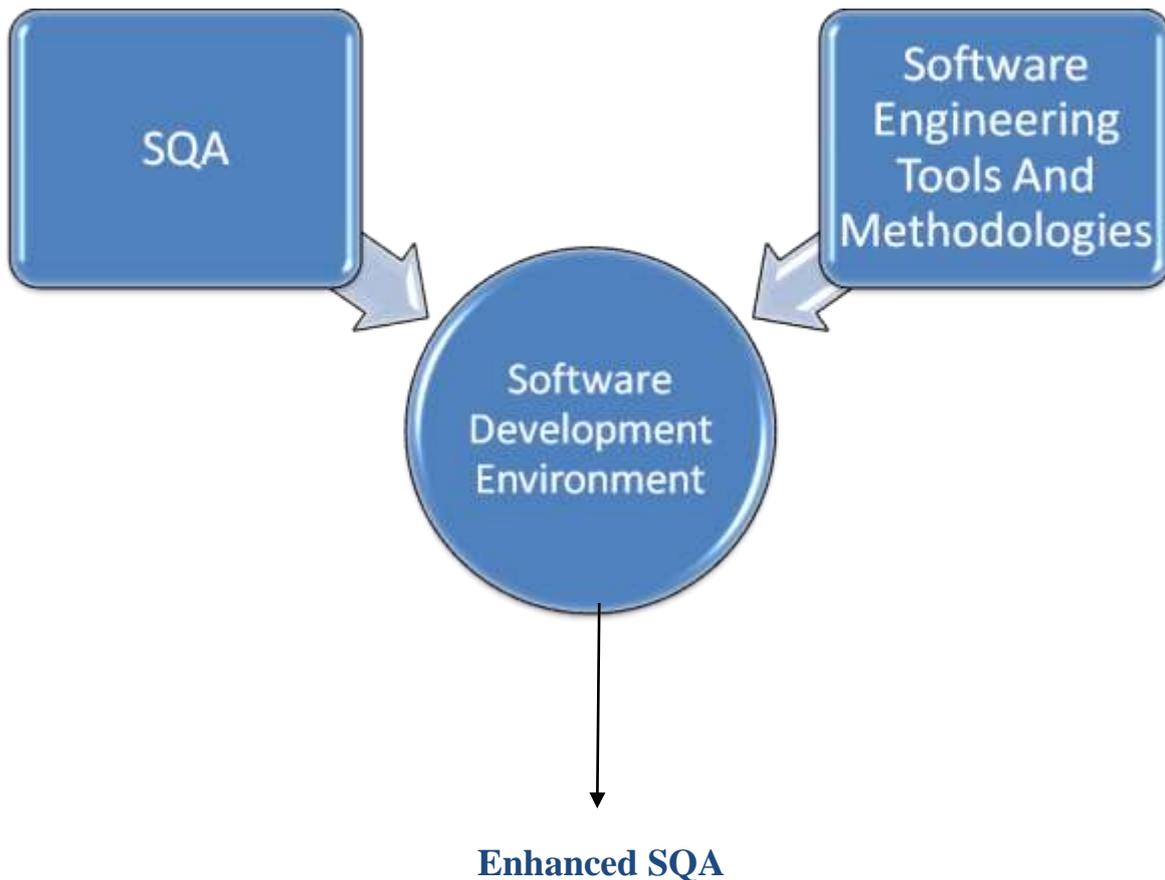
wedged in system test provides in sequence on how to transform the scrutiny process. Quantitative histories can perk up that route. Inside this technique knowledge is propagated the whole time the association .We can study how to perk up superiority & efficiency & how to pick up meaning & appraisal of goals. This footstep involves organizing the prearranged awareness hooked on an in sequence to assist advance scheduling, organizing, enlargement and estimation.



**Figure 11: SQA Project Management Activities**

---

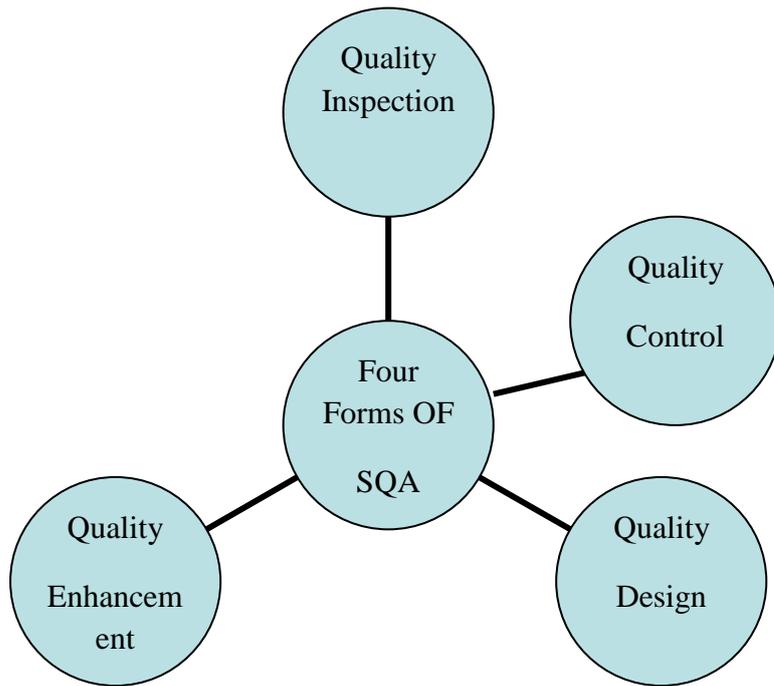
### 4.3.3 Integrated SQA activities:



**Figure 12: Integrating SQA**

In figure 15 we can see Software Development Environment integrate software tools and methodologies appear to the user as a single, organized system of hardware and software that supports the entire life cycle. In this type of environment tools work together synergistically and make the transition smooth between phases. The environment has a single operational style, as well as shared data, communication media and interfaces.

Integrated SQA activities provide several capabilities to increase Software Quality and offer solutions to help improve other SQA problem.



**Figure 13: Four Forms of SQA**

**Quality Inspection:**

Quality inspection means before shipping the product simple evaluation should be done to detect obvious faults in a product.

**There are some problems in an inspection:**

- Testing too limited and late in the development life cycle.
- Test environment not sufficient, resources have been wasted from the start developing a product.
- Quality cannot be obtained by testing only.

**Potential Environment Solutions:**

- Verification throughout software development
- Select appropriate use of tools and methods to correct the problems.
- Advanced quality techniques should be applied.

---

**Quality Control:**

Quality control means reviews, examines individual development areas like process parameters, parts and partial assemblies. Quality controls use statistical techniques to determine whether the development is producing a product with acceptable quality.

**Deficiency in a quality control is:**

- Labor demanding problem.
- Final inspections, reviews suffer from being too late and too limited.
- Fail to budget proper resources.
- May not be treated seriously by management

**Potential Environment Solutions:**

- Provide more automation tools.
- Integration of SQA and Software Development Environments.
- Support engineering decisions
- Provide decisions support tools.

**Quality Enhancement:**

Quality improvement means eliminating the causes of problems with a feedback mechanism. This requires consistent tracking of process and product data.

**Troubles in quality improvements are:**

- Lack of definitions
- Manual data collection
- Fragmented necessary data sources
- Lack of feedback.

**Potential Environment Solutions:**

- Apply and improve existing definitions.
- Automate data collection
- Make data collection visible and reliable
- Show managers effects of their decisions.

---

## **Quality by Design:**

Identifying the key parameters allows the modeling and realistic prediction of quality. Additional studies them to determine the best settings for control label parameters. In software development today, quality by design concepts remain more a research topic than a practical applications.

## **We have find out some problems in this form of SQA:**

- Trialing is difficult because of lacking knowledge of practical applications.
- Variable optimization in skill and experience level of the people working on project.
- Lack of information

## **Potential Environment Solutions:**

- Provide confirmation from past projects.
- Set up practicable thresholds and baselines.
- Provide significant information by analysis of past and present projects.

## **Implementation & Guide**

Based on previous experiences we are annoying to assess & perk up the excellence in more than a few organizations by SQA guidebook. Meta –SQAP should be explicating to software project leaders & managers as a part of SQM (Software Quality Measurement) seminar program by means of the guidebook.

### **4.4 Meta-SQA Methodology guidebook**

We need SQO methodology manuals for developing good quality software projects place Meta-SQOM into achievement. This manual explains the meaning of Meta SQA & how to set into process the software scheme. The settlement of the regular SQOM/SQA program is:

1. Project SQA activities well planned & be consistent in the middle of projects & separation from side to side the SQA Methodology manuals.
2. Project SQA activities accurately performed because of planned operational activities.

- 
3. SQA program be able to be enhanced by repeating 5 phases of SQA actions. These are SQA goal definition, SQA program development, SQA program analysis, SQA program evaluation & feedback.
  4. Software Development Environment integrates software tools and methodologies appear to the user as a single, organized system of hardware and software that supports the entire life cycle. In this type of environment tools work together synergistically and make the transition smooth between phases. The environment has a single operational style, as well as shared data, communication media and interfaces.

#### **4.5 Division of the standard SQA program**

Project SQA is defined to be performed by the following 4 categories of activities in Meta-SQAP. Meta-SQAP provides an association and measures intended for projects SQA activities. The measures are primarily clear by IEEE standards.

1. Review: - This grouping of actions includes propose evaluate, walk-through, convention scrutiny and authentication. Arrangement is reviewed by members of the assignment numerous periods in every segment. Every software maturity segment is well planned finished following confirmation.
2. Configuration management: - The core behavior in this group is configuration recognition and pattern organizes.
3. Testing: - This group of actions includes the growth of well-planned certification for testing, testing statistics growth based on testing technologies and tools, and the scrutiny of test consequence, implementation the test segment.
4. Problem management: - This group of behavior includes complexity account supervision, completion of countermeasures meant for individual's difficulties, and testing to stop the reoccurrence of difficulties.

---

#### **4.6 Practical use of SQA Meta-Methodology**

Meta Methodology will provide useful hold up for the project achievement managers with tools that categorize tasks, milestones, configurations, schedule, costs, and responsibilities. This Methodology also works as a roadway growth, changes and problems areas.

All Meta-Methodology of SQA activities are defined in this SQOM manual which gives right direction to staff members. When a goal is defined at the beginning of a project, strategy for the scheme SQA behavior is accessible. Every optimistic result of the SQA activities integrated into normal SQA agenda when development is based on Meta-SQOM. The better distribution of standard SQA program then becomes the new source used for each and every activity in project.

This manual is very much helpful to manage the SQA program including report documents, monitoring, controlling, verifying & growth of separation of standard SQA programs. It encourages the use of Meta-SQOM by distributing manuals & as long as tutoring for managers & mission heads.

We have applied our methodology in Ranorex Testing Tool for check its efficiency and effectiveness, which is used to automate all kind of Windows applications and web sites.

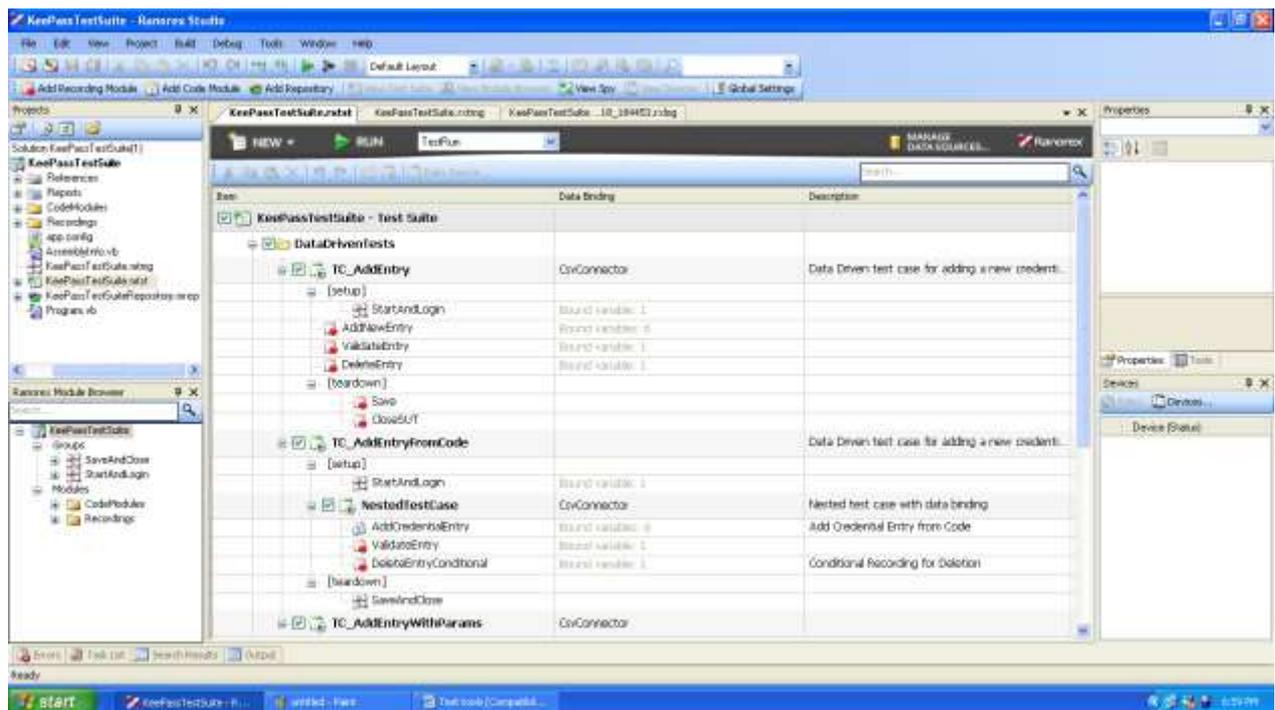
This research includes a description of the method as well as sample session reports, and a tool we used that produces successful project reports. This methodology helps to analyze software test projects.

## 4.7 Analysis of Meta-SQOM

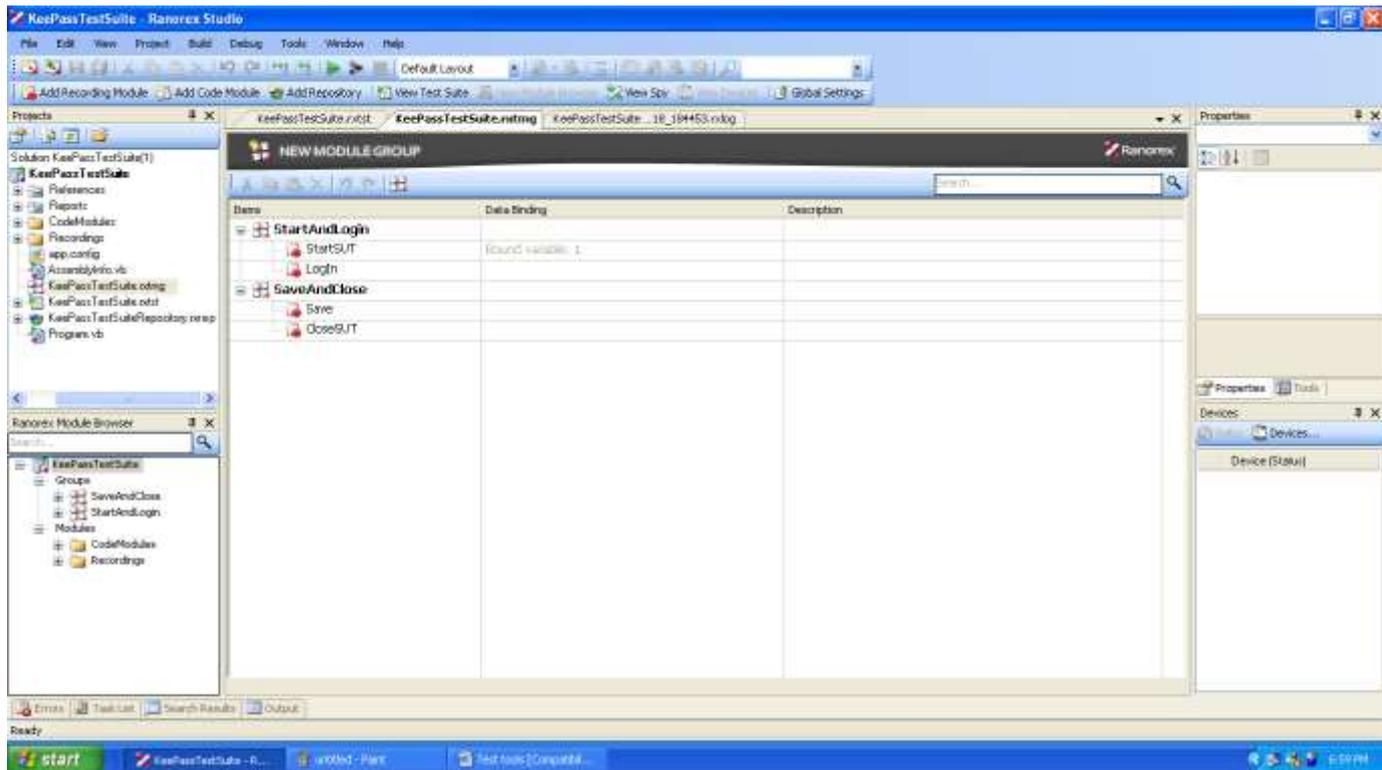
The Meta SQOM is evaluated by Ranorex testing tool for checking whether the project is success or not is determined by the extent to which the actual results fulfill the targets for product quality

### KeePass Password Safe: System under Test

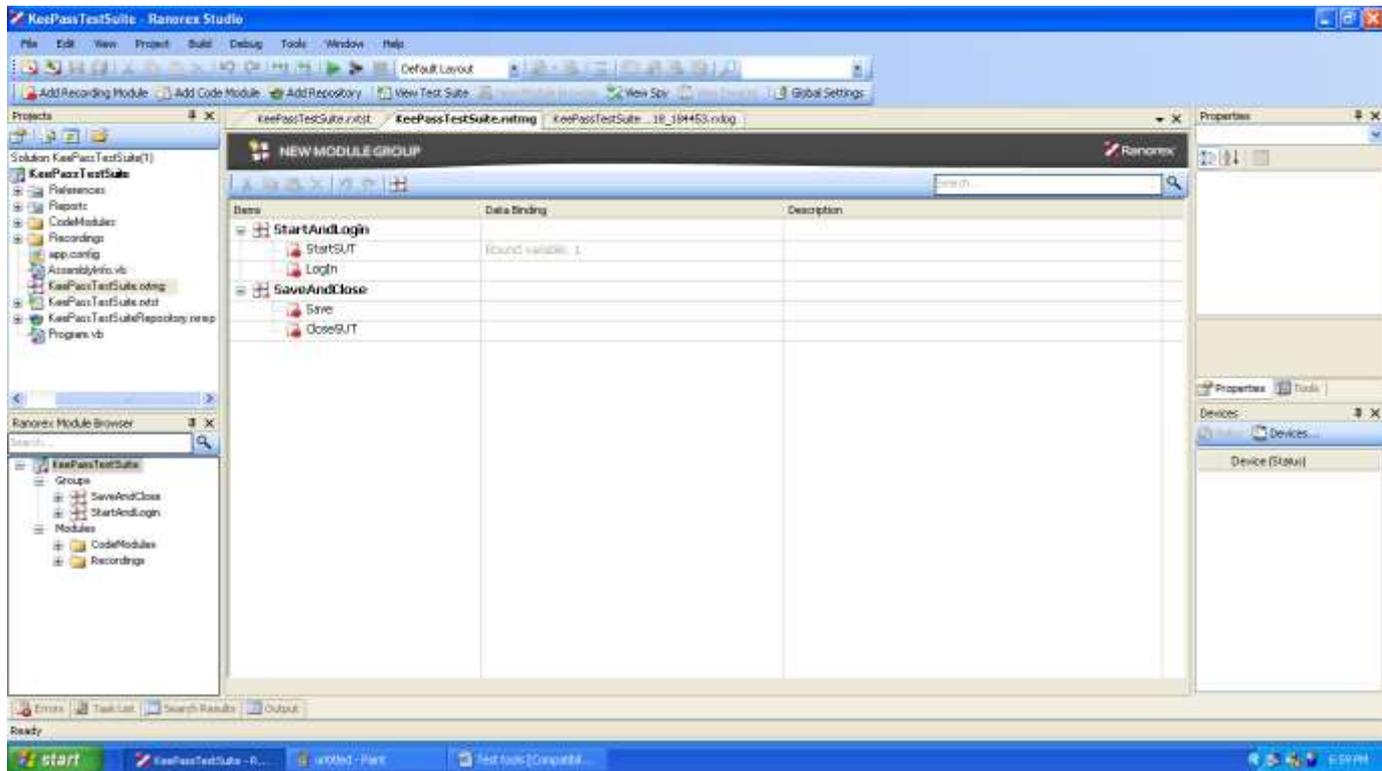
This demonstration uses the open source tool KeePass to demonstrate Ranorex functionality for automating an application in order to build up a testing framework. KeePass is a very powerful and easy to use tool for managing passwords and other private information.

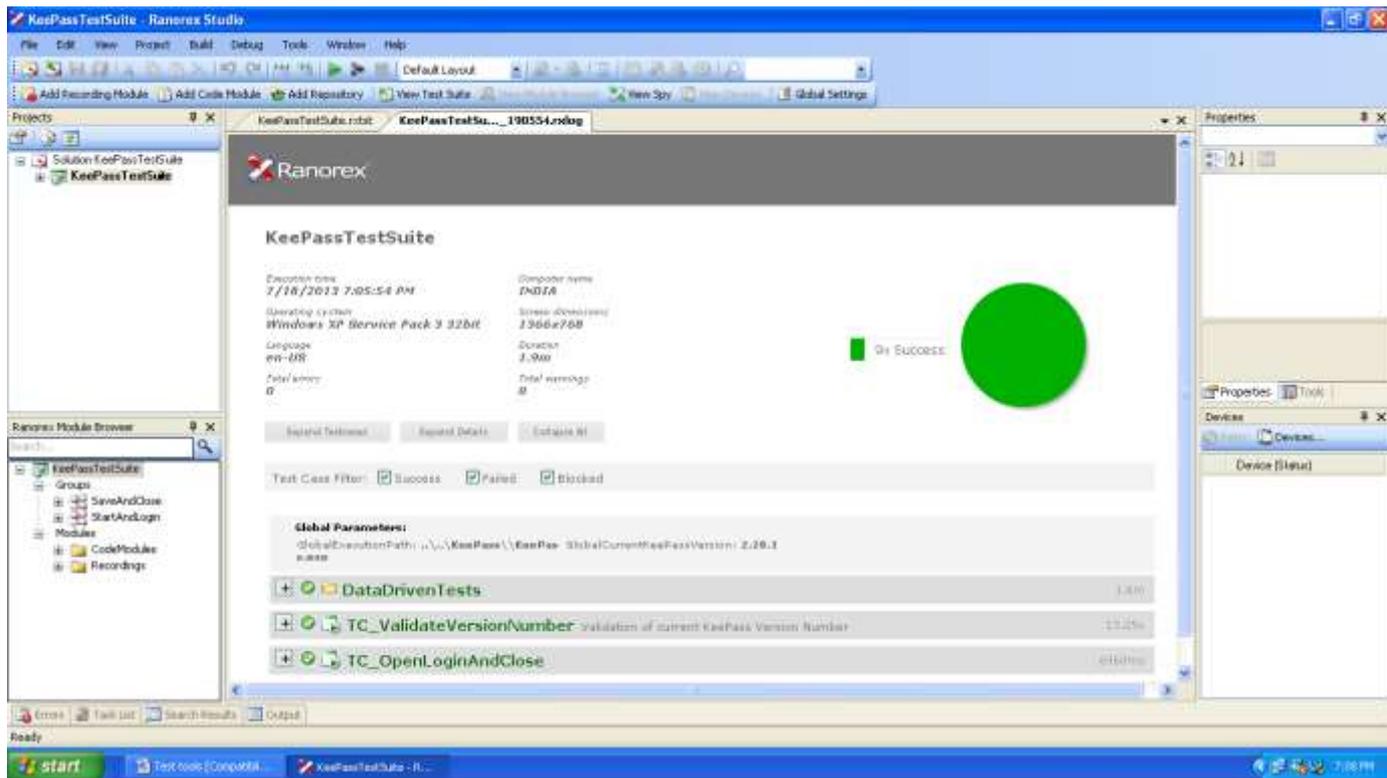


Snapshot of KeePassTestSuite



Snapshot of KeePassTestSuite Project Module





**Snapshot of KeePassTest Report**

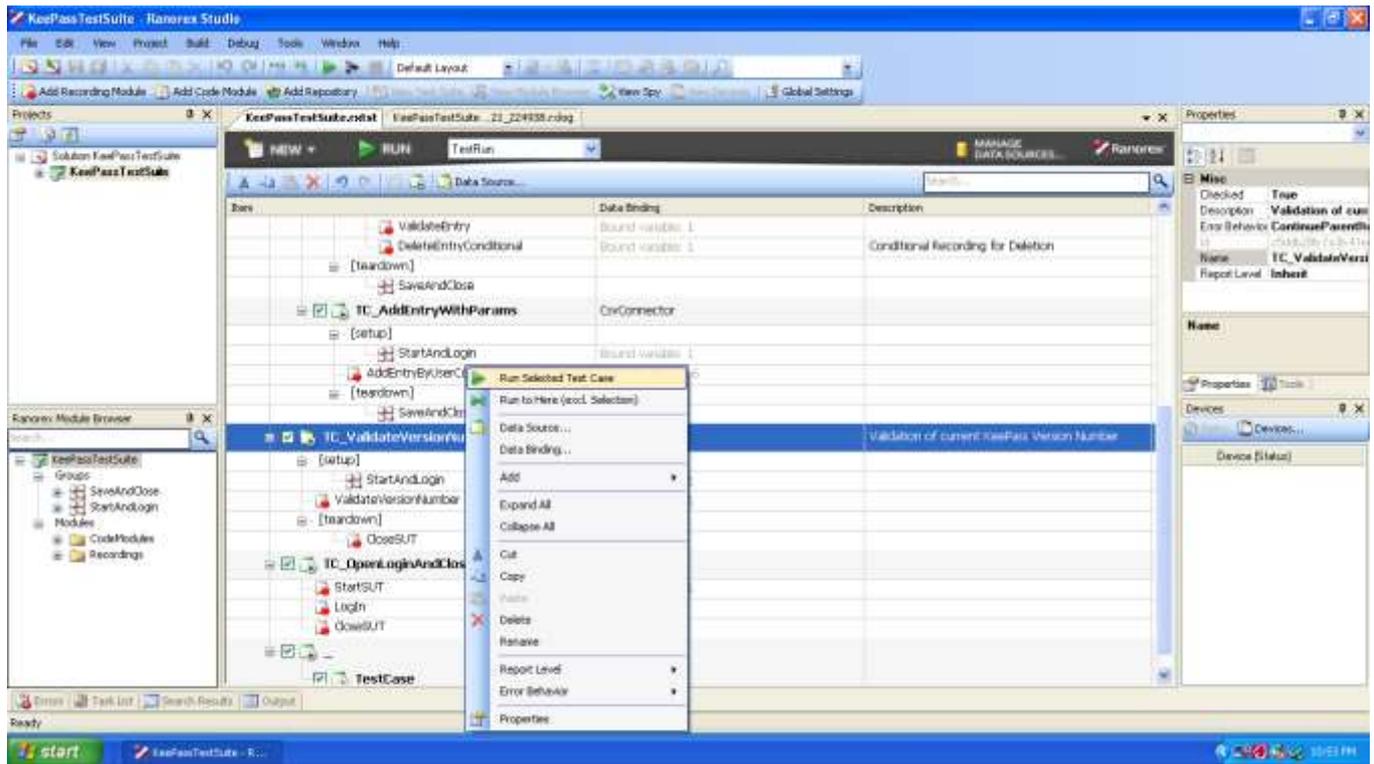
**Project Name: KeePass Password Safe**

**Test Title: KeePassTest Suit**

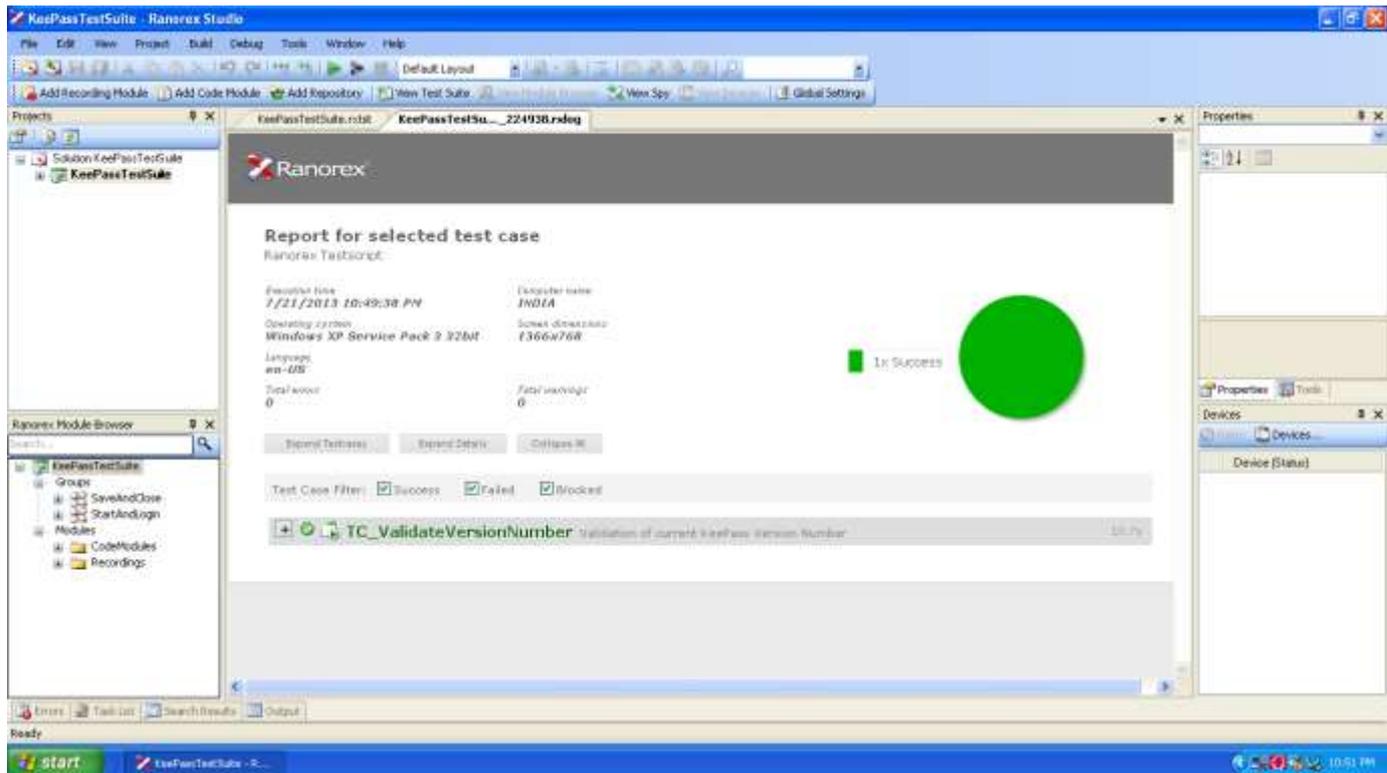
**Test Execution Date: 7/18/2013 7:05:54 PM**

**Description: For managing password**

Test Data	Total Error	Warning	Status (Pass/Fail)
DataDriven Tests	0	0	Success
TC_ValidateVersionNumber	0	0	Success
TC_OpenLoginAndClose	0	0	Success



Snapshot of Run selected Test Case



Snapshot of report for selected test case

**Module Name: KeePass Password Safe**

**Test Title: TC\_ValidateVersionNumber**

**Test Execution Date: 7/21/2013 10:49:38 PM**

**Description: Validation of current KeyPass Version Number**

Test Data	Total Error	Warning	Status (Pass/Fail)
TC_ValidateVersionNumber	0	0	Success

---

## **Chapter 5**

### **Conclusion, Results and Future work**

---

We discussed the generic ways to improve and control quality of software as long as each project is managed and developed differently, it will be easy to determine what truly affects quality with the help of this paper. By providing common (Framework) Meta Methodology throughout all the projects, each project can be significantly defined, executed and evaluated leading to a better understanding of a software quality. We can create a successful and QA Testing Tool on the basis of this Meta –Methodology.

SQA methodology is an important factor to every software development, designed to promise that quality & productivity requirement are fulfilled. We put this special issues together to help people understand the increasing importance of SQA methodology as an essential part of software projects, outline some new ideas (Integrated SQA) and proposes a measurement based methodology for specifying, monitoring & evaluating the software product quality .We are proposing a new Software Quality Oriented Methodology (SQOM), which is flexible to incorporate new changes in the software industry and provides detailed guidelines and templates for real world implementation and customization.

Future software development environment will contribute to increased software quality. They will enhance and support the existing SQA process, but they will not solve all the problems. To cope with the future challenges, SQA must be mechanized—It must become an integral part of any software development process.

There are several Meta-Methodology manuals have been developed. Most of those manuals are not being used. Perhaps we can increase the use of Meta SQA methodology by proper documenting a list of concerns that should be considered by team managers and project leaders. We need to add continue to add our list. Further research is suggested to recognize other quality problems in software projects not found in this research and to evaluate the impact of different practices on project quality.

Sometimes it is difficult for big projects like military projects and space projects to allocate plenty time and manpower for SQA behavior and SQA management.SQA methodologies are still in its early stages. The SQA activities that are industrial be supposed to maintain to get better. This great scale assignment management methodology and QA testing tool is to be developed next.

---

## References

- [1] Hower, Rick (2006). *Software QA and Testing Resource Center*. Updated April 2006. Web site: <http://www.softwareqatest.com/>
- [2] *Software Quality Assurance, NASA*.
- [3] Humphrey, W.S. 1988. Characterizing the Software Process. *IEEE Software* 5(2);73-79, March 1988.
- [4] A Management Methodology to reduce Risk and Improve Quality, Published by IEEE Computer Society December, 2007.
- [5] Jani, Hajar Mat “Applying Case-Based Reasoning to software requirements specifications quality analysis system” *Software Engineering and Data Mining (SEDM)*, 2010 2<sup>nd</sup> International Conference on 23-25 June 2010.
- [6] How to improve Software Quality Assurance in developing countries, *Advanced computing :An International Journal*, Vol.3, No.2, March 2012.
- [7] The Economics of Software Quality Assurance : A Simulation –Based Case Study, *IEEE*, September 1988.
- [8] *Quality Assurance in Future Development Environments*, IEEE 1987
- [9] *A Measurement –Based Approach for Implanting SQA & SCM Practices*, IEEE 2000
- [10] *Software Standards IEEE 730 : A guide to writing successful SQA plans*
- [11] *Tailoring SQA to fit your own life cycle*, IEEE March, 1988
- [12] *Enhancing Defect Tracking Systems to Facilitate Software Quality Improvement*, IEEE
- [13] *Towards Continuous Process Development: Designing SQA Function for Large Organisation Needs*, 1998, IEEE
- [14] *Software Quality Assurance –Concepts and Misconceptions*, 1998 IEEE
- [15] *Evaluation and Measurement of Software Process Improvement –A systematic Literature Review*, IEEE 2012
- [16] *Studying the Impact of Software Development Characteristics on Project Goals: A Casual Model*, the *Open Software Engineering Journal*, 2010
- [17] Glick, B,” *An SQA Quality Tracking Methodology*” *Software Maintenance*, 1990, *Proceedings. Conference on 26-29 Nov 1990*

---

[18] Hribar,L,Burilovic,A.,Huljenic,D.”Implementation of the Software Quality Ranks method in the legacy product development environment”Telecommunications,2009.10<sup>th</sup> International Conference on 8-10 June 2009

19] Quality practices in open source software development affecting quality dimensions, Jul 2011.

[20] Software quality analysis and measurement service activity in the company Tanaka, T.; Aizawa, M.; Ogasawara, H.; Yamada A.; Software Engineering, 1998.