STUDY, ANALYSIS AND DESIGN OF ADAPTIVE SOFTWARE ALGORITHMS FOR APPLIED AUTOMATED SYSTEM DEVELOPMENT AND TESTING PATTERNS IN THE AREA OF HIGH PERFORMANCE RUN TIME DYNAMIC STORAGE IN NETWORK CLOUD ENVIRONMENT

Submitted for the Award of the Ph. D. degree of
PACIFIC ACADEMY OF HIGHER EDUCATION
AND RESEARCH UNIVERSITY

By
Vikas Kumar Choudhary
(Reg./PAHERU/Ph.D./FOCS/14-1057)

Under the Supervision of

Dr. Sanjay Chaudhary
Professor, Dept. of CSE,
Madhav University, Sirohi (Raj).

FACULTY OF COMPUTER SCIENCE

PACIFIC ACADEMY OF HIGHER EDUCATION AND RESEARCH UNIVERSITY
UDAIPUR

D E C L A R A T I O N

I Mr. Vikas Kumar Choudhary

S/o Dharmendra Kumar Choudhary

Resident of 1/80, R.H.B. Colony, Goverdhan Vilas, Udaipur, Rajasthan, Pin: 313001

Hereby declare that the research work incorporated in the present thesis entitled

"STUDY, ANALYSIS AND DESIGN OF ADAPTIVE SOFTWARE ALGORITHMS FOR APPLIED AUTOMATED SYSTEM DEVELOPMENT AND TESTING PATTERNS IN THE AREA OF HIGH PERFORMANCE RUN TIME DYNAMICSTORAGE IN NETWORK CLOUD ENVIRONMENT” is our original work.

This work (in part or in full) has not been submitted to any University for the award or a Degree or a Diploma.

I have properly acknowledged the material collected from secondary sources wherever required.

We solely own the responsibility for the originality of the entire content.

Signature of the Candidate

Date:

Signature of the Supervisor/s

CERTIFICATE

Its gives me immense pleasure in certifying that the thesis entitled **STUDY, ANALYSIS AND DESIGN OF ADAPTIVE SOFTWARE ALGORITHMS FOR APPLIED AUTOMATED SYSTEM DEVELOPMENT AND TESTING PATTERNS IN THE AREA OF HIGH PERFORMANCE RUN TIME DYNAMICSTORAGE IN NETWORK CLOUD ENVIRONMENT** and submitted

By **Vikas Kumar Choudhary**, is based on the work research carried out under my guidance.

He/ She has completed the following requirements as per Ph.D. regulations of the University;

I. Course work as per university rules.
II. Regularly presented Half Yearly Progress Report as by prescribed the University.
III. Published/ accepted minimum of two research paper in a refereed research journal.

I recommend the submission of thesis as prescribed/ notified by the University.

Date: **Name and Designation of Supervisor/s**

COPYRIGHT

I, Vikas Kumar Choudhary, hereby declare that the Pacific Academy of Higher Education and Research University Udaipur, Rajasthan shall have the rights to preserve, use and disseminate this dissertation/thesis entitled

"STUDY, ANALYSIS AND DESIGN OF ADAPTIVE SOFTWARE ALGORITHMS FOR APPLIED AUTOMATED SYSTEM DEVELOPMENT AND TESTING PATTERNS IN THE AREA OF HIGH PERFORMANCE RUN TIME DYNAMIC STORAGE IN NETWORK CLOUD ENVIRONMENT"

In print or electronic format for academic / research purpose.

Date:

Signature of the Candidate

Place: Udaipur

ACKNOWLEDGEMENT

“Success is the final destination and every failure is intermittent blockades” a well said quote by our former president Dr. A.P.J. Abdul Kalam.

Though it is hard to walk on difficult roads, it leads me to the most beautiful Destinations. The past few years were challenging and have taught me many lessons, but the environment and people around me always nurtured me with positive energy persistently. From the God’s Grace a new member (Daughter Nivita) added to my family. Thus, it is an honour for me to express my thanks to all those who were the part of my memorable journey. Words wouldn’t adequate to express my gratitude for my parents without whom I would have accomplished nothing.

My father, **Shri Dharmendra Kumar Choudhary** is the first person who dreamt for my Ph.D. right from the time of my schools.

My mother, **Smt. Saroj Devi**, has raised me in all the difficulties and who has stayed by my side with the kind of love one could never imagine of.

With due respect, I thank my supervisor, **Dr. Sanjay Chaudhary** for his privileged guidance, suggestions and help in all possible aspects to make me achieve my goals. He always encouraged me to opt for the new ideas and believed in me. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth.

With deep sense of respect and gratitude I thanks to **Prof. Hemant Kothari**. His immense experience and immeasurable knowledge showed me the way to make the work informative and exhaustive. His constant encouragement and appreciation kept me motivated throughout and helped me immensely in completing this work.

Special thank my Family Mrs Anita (Wife), Neeta (Sister), Rashmi (Sister in Law), Lalit (Brother), Mr. Harsh Solanki (Brother in Law). I thank all of them for their support and encouragement all the time.

Vikas Kumar Choudhary

Abstract

Adaptive automation refers to process in which both the user and the system can initiate changes in the level of automation. The first adaptive automation systems were implemented in associate systems based on models of user's behaviour and workload. Recently, however, systems have been developed that follow the neural networks approach and use Artificial Intelligence measures to trigger changes in the state of automation. Studies suggested that this approach can facilitate user performance. Further, evidence is beginning to show that people not only think of adaptive systems as co-workers, they may even *expect* them to behave like humans.

Adaptive automation creates new challenges for both users and designers that go beyond traditional ideas of human-computer interaction and system design in run time dynamic network storage cloud.

In this research work implementation of adaptive automation, I will be discussed first with Adaptive Strategies with example of adaptive automation systems having different workload.

There is a discussion about Human computer Etiquette, where human and computer have different behaviour strategies, hence human should be careful while dealing with computers while doing automation.

Adaptive automation may be of different types in different kind of environments. This paper will discuss about managing adaptive automation in dynamic storage cloud. If there are multiple device involved then Emulated Software can be used for Cross Device Testing Automation.

To work adaptive automation in storage network, need to know about Fundamental Adaptive Storage Networks, about Adaptive network vision which includes Programmable Infrastructure, Analytics and Intelligence and third, Software control and automation.

Then discussion about data storage networks solves present challenges, uses of adaptive storage networks, solutions, open networking and strategic partnership.

Next will see adaptive interface to scalable cloud storage

To know and proceed further with adaptive automation, need to know about adaptive life cycle management, which have mainly six phases comprises of first, Decision of Automation involves expectation, QA, CI/CD and management support, then second decision to acquire automation tools, thereafter automation tools acquisition. Then, third automation introduction process about process analysis for fitting tools at right place. Fourth phase talks about Automation Planning, Automation Design, and Automation Development in Technical Environment, Phase five details about execution and management of automation and last phase talks about Review and Assessment of automation implementation.

Then will describing about automation artefacts includes code base, dependencies, configuration, Backup, Software Build, Run and Release, different processes, handling concurrency and administration and Log monitoring.

As the environment for considering adaptive automation is Storage Cloud Networks so need to understand about cloud computing and Storage fundamentals, Will see types and definition of storage devices (magnetic, flash, optical, online cloud and paper) their need, storage location, capacity and common problems encountered with storage devices.

Further storage networks are of two types, Direct Attached Storage and Network Attached Storage, will highlight about their uses cases, categories, security issues, future outlook and trends and their convergence.

To understand the storage network, distinction between storage protocols and storage network protocol need to be understood. Storage Protocols includes SCSI, FC, CIFS, NFS and HTTP. These protocols defines formation and storing data. Storage Networking Protocols includes iSCSI and FCoE, defines the way of transferring data over the IP or Ethernet based networks.

Storage virtualization plays an important role to achieve dynamic storage and further adaptive automation, there are two basic methods of Storage Virtualization, file based and Block Based. Virtualization methods are of three types, Host Based, Array Based and Network Based.

Now will check implementation of Storage Protocols, Storage Networks Protocols and Storage Virtualization with respect to Cloud Storage. With respecting to cloud computing will check definition of Software as a Service, Platform as a Service, Infrastructure as a Service and Database as a Service. Thereafter access methods (data transfer) of cloud storage are discussed using Web Service API, File Based Protocols (NFS, CIFS, FTP). Cloud storage will have some advantage over conventional storage systems.

There are some misconceptions about development cloud (Private Cloud) within organizations about its usages and relations of other environments and promotion to higher environments as per the internal policies of organization, concerns regarding to security and access and support and role of administrative group.

There are different ways of hosting shared services in Development cloud. Cloud services are delivered on-premise and off-premise.

Cloud service models and cloud deployment models (public, private, community and hybrid) have some security risks with respect to data, regulatory compliance, and Law, Vulnerability, Access, Identity and Infrastructure management. Also other expect of cloud computing, benefits of cloud computing in organization.

Cloud computing evaluation whether it can be dedicated, virtualized Internal, External and Hybrid Cloud. Implementation strategy and Initiatives also mentioned.

Can be used for adaptive development because it’s iterative, time boxed and change tolerant.

Adaptive S/W development management has polices namely Passive and Active Management and Leadership-Collaboration Management. Adaptive

Automation Testing have some patterns named as Data Patterns, Technical Patterns, Proxy Patterns, Business Patterns, Page Object Patterns, Façade Patterns, Factory Patterns, and Singleton Patterns.

There are some tools and technologies available to achieve results on Run Time Dynamic Adaptive Automation Testing (RTDDA). These tools are mainly divided in two categories Free/Open Source Tools and Commercial Tools. There are some tools used for cross platform testing, Hence in the list of many tools there should be strategy to choose right tools on the basis of Market research, Experts View and Personal Experience. Sometime if required tools may be developed in-house.

TABLE OF CONTENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The adaptive automation focuses on current research and developments in the neuroscience of information processing called as Neural Networks and Artificial Intelligence and how that knowledge can be used to improve performance in real-world environments with computing systems or information processing with cloud Storage. Here we need to understand that how the brain processes perceptual and cognitive information can lead to better designs for equipment, systems, and tasks by enabling a tighter match between task demands and the underlying brain processes similar way by using techniques of AI and neural networks Computing systems can process information to store and retrieve data at run time dynamic networked cloud and fetch as and when required.. Ultimately, research in this area can lead to safer and more efficient working solutions.

Ironically, interest in neural networks and Artificial Intelligence evolved from research surrounding how operators interact with a form of technology designed to make work and our lives easier – automation. In general, automation is a computing system machine agent capable of doing tasks normally performed by a human being. For example, In a Software Testing environment, either Software testing is done manually or it’s done using some tools and technique using automation designed to reduce task demands and workload. And it allow individuals to increase their scope of operation or control, perform functions those are beyond normal abilities, maintain performance for longer periods of time, and perform fewer mundane activities.

Automation also helps to reduce human error and increase safety.

Research on human interaction with automation concludes that it does not always make the job easier. Instead, it *changes* the nature of work.

“Automation changes the way activities are carried out and can introduce new and different types of problems. Automation can lead to different types of errors because user system goals may be inconsistent with the goals of systems and subsystems where subcomponents of system are tightly coupled, some weird problems may propagate more quickly and be more difficult to isolate. In addition, highly automated systems leave very less activities for user systems or humans to perform. Consequently, the user becomes a more passive monitor instead of an active participant. It’s shown that this shift from performing tasks to monitoring automated systems can actually inhibit one’s ability to detect critical signals or warning conditions. Further, a user (Technical Person) manual skills can begin to deteriorate in the presence of long periods of automation “

1.2 ADAPTIVE AUTOMATION

Given the problems associated with automation noted above, researchers and developers have started to find and develop alternative methods for implementing automated systems. Hence idea of *Adaptive automation* developed which has been proposed to address shortcomings of traditional automation. In adaptive automation, the automation level and/or the number of systems operating with automation can be modified dynamically in real (run) time.

“Additionally changes in the state of automation can be initiated by *either* the human (User) or the system (Computing Devices). Consequently, adaptive automation enables the level or modes of automation to be tied more closely to user needs at any given moment”

Adaptive automation systems can be identified as either *adaptable* or *adaptive. “I we talked about* taxonomy of adaptive technology. One side of this taxonomy concerns the underlying source of flexibility in the system, i.e., whether the information displayed or the functions themselves are flexible. Another side addresses how the changes are invoked.”

In **adaptable** systems, user initiates the changes in the allocation of functions. However, in **adaptive** systems both the user and the system can initiate changes in the state of the system.

The distinction between adaptable and adaptive terminology can also be described with respect to authority and autonomy.

"There are several levels of automation that range from completely manual, to semiautomatic, to fully automatic."

With the increase of level of automation, systems adapts more authority and autonomy and at the lower levels of automation, systems by pass autonomy to the user. The user either reject or accept the suggestions and then implement the action.

At moderate levels, the system have the autonomy to process with the suggested actions accepted by the user. And in the end at higher levels, the system may decide on a course of action, implement the decision, and merely inform the user.

Hence higher level of autonomy is expected with system called as run time dynamic adaptive systems.

"With respect to Scerbo's (2001) taxonomy, adaptable systems are those in which the user maintains authority over invoking changes in the state of the automation. On the other hand in adaptive systems, authority of invocation is shared. Both the user and the computing system can initiate changes in state of the automation.

There has been discussions having control among modes of operation. Some argue that users should always have authority over the system because they are ultimately responsible for the behaviour and performance of the system.

In addition, it is possible that users may be more efficient at managing resources when they can control changes in the state of automation.

Many of these arguments are based on work with critical systems related to security of data in which safe and secure operation is of utmost concern. There may be times when the user is not the best judge of when automation is needed. i.e. changes in automation may be needed at the precise moment the user is too busy to make those changes.

“It can be shown mathematically that the best decisions concerning whether to abort a data transfer are not those where either the human or the Data Centre maintain full control? Instead, the best decisions are made when the user and the automation share control.

Some critical situations where the user system is vulnerable, it would be extremely important for the system to have authority to invoke automation. If situations are at stake or the system is in jeopardy, allowing the system to intervene and circumvent the threat or minimize the potential damage would be paramount. For example, it is not uncommon for many of today's data centre/cloud to sustain high bandwidth enough to fulfil high demands. These Conditions make a strong case for system-initiated invocation of automation.

1.3 ADAPTIVE STRATEGIES IN RUN TIME DYNAMIC SYSTEMS

Let’s discuss about strategies by which adaptive automation can be implemented.

One set of strategies addresses system functionality in which entire process may be allocated to either the system or the user (It may be another system), or a specific task can be divided so that the system and user (client or server) both share responsibility for unique portions of the task. Alternatively, a task could be transformed to a different format for the user system to perform.

A second set of strategies is related to the triggering mechanism for shifting among modes of automation. This is a goal-based strategy. In this, levels of automation are triggered by a set of criteria depends on external events/forces. Thus, the **run time dynamic** system invoke the automatic mode only during specific tasks or in an emergency situation. Another approach would be to use real-time measures of user systems performance to invoke the changes in automation.

A third approach uses models of user system's (client or server) performance or workload to drive the adaptive logic in run time dynamic systems.

For example, a system could estimate current and future states of a user's activities, intentions, resources, and performance. Information about the user, system, and the outside world could then be interpreted with respect to the user's goals and current actions to determine the need for adaptive aiding. Finally, measures that reflect system workload can also be used to trigger changes among modes.

1.3.1 EXAMPLES OF ADAPTIVE AUTOMATION SYSTEMS

Adaptive automation has its beginnings in artificial intelligence. In the 1970s, efforts were made toward developing adaptive technologies to help allocating and organizing tasks between humans and computing systems.

By the 1980s, researchers began developing adaptive interfaces. For instance, Wilensky, Arens, and Chin (1984) developed the UNIX Consultant (UC) to provide general information about UNIX, information about executing/running UNIX commands, as well as debugging/analysing information. The UC could analyse user queries/questions, confirms the user goals, monitor the user's interaction history with system, and present the system's response for better performance. These methodologies are used now a days in business like online shopping to check user interests.

1.3.2 WORKLOAD AND SITUATION AWARENESS

Workload One of the arguments for developing adaptive automation is that this approach can moderate user's system workload. Most of the research till date has assessed workload through primary task performance. To perform a simulated task where the object was to send data to targets without colliding with one another. During manual control, the user were required to assess the situation on the display, make decisions about which targets to eliminate, and implement those decisions. During a shared condition, the user and the computer could each perform the situation assessment task. The computer scheduled and implemented the actions, but the controller had the ability to override the computer's plans. The users were also asked to perform a secondary task requiring them to monitor the movements of a pointer and correct any deviations outside of an ideal range. The secondary process used to invoke the automation on the primary process. For half of the participants, the computer suggested changes between automation or manual operation of the primary task and for the remaining participants, those changes were mandated.

In data storage cloud network shared control resulted in better performance than manual control on the primary task. The results showed that compulsory use of automation also improved performance during periods of manual operation. Regarding the secondary task, when use of automation was mandated, data processing workload was lower during periods of automation; however, under periods of manual control, workload levels actually increased and similar to those seen when its use was suggested. It comes out that authority over invoking changes between modes had differential effects on workload during periods of manual and automated operation. Specifically, it's found that the requirement to "consider" computer suggestions to invoke automation led to higher levels of processing workload during periods of shared/automated control than when those decisions were dictated by the computer.

Situation awareness as per the study of the effects of adaptive automation on situation awareness. Situation awareness is the ability to perceive elements in the environment, understand their meaning, and to make projections about their status in the near future. One might assume that efforts to moderate workload through adaptive automation would lead to enhanced situation awareness; however, that relationship has yet to be demonstrated empirically. In fact, within an adaptive paradigm periods of high automation could lead to poor situation awareness and make returning to manual operations more difficult. .

Effects of the adaptive conditions were moderated by complacency potential. Specifically, for individuals in the control conditions, those who were high as compared to low in complacency potential had much lower levels of situation awareness. On the other hand, there was no difference in situation awareness scores for high and low complacency individuals in the adaptive conditions. More important, the situation awareness scores for both high and low complacency individuals were significantly higher than those of the low complacency participants in the yoke control condition. The authors argued that a brain-based adaptive automation system could ameliorate the effects of complacency by increasing available attentional capacity and in turn, improving Situation Awareness.

1.4 HUMAN-COMPUTER ETIQUETTE

There has been interest in the merits of an etiquette for human-computer interaction for the requirements of adaptive automation. Etiquette is a set of prescribed and proscribed behaviours that permit meaning and intent to be ascribed to actions. Etiquette makes social interactions more cooperative and polite. Importantly, rules of etiquette allow one of form expectations regarding the behaviours of others. People generally adopt many of the social terminologies used in human-human interactions when they interact with computers.” Moreover, they also expect computers to adhere to those same conventions when computers interact with users.

When humans interact with computing systems that incorporate intelligent agents/subsystems they may expect those agents to conform to accepted rules of etiquette. However, the norms shall be contextually dependent: what is acceptable for one application may violate expectations in another. Hence, there may be a need to understand the procedures under which automated computing systems should behave in expected polite way.

There might be expectations regarding human etiquette to their interactions with adaptive automation. It's been observed that much of the dialog between team members in a data centre environments was focused on communicating plans and intentions. Any automated component would need to communicate in a similar manner to be accepted as a "team" player. Consequently, the described earlier was designed to allow users and the system to communicate in a conventionally accepted manner.

The benefits of adopting a human-computer etiquette are described in a study of human-automation interactions in adaptive data storage environment. In particular, they focused on interruptions. Participants were asked to perform the tracking and resource management tasks from the storage devices. Another task required participants to interact with an automated system that monitored system parameters, detected potential failures, and offered advice on how to diagnose faults. The automation support was implemented in two ways. Under the "normal" condition, the automated system would withhold advice if the user was in the act of diagnosing the system or provide a warning, wait few seconds, and then offer advice if it determined the user was not interacting with the system. Under the "extreme" condition the automated system offered/popped its advice without warning while the user was performing the diagnosis. The normal and extreme automation as examples of good and poor etiquette, respectively. Additionally, they examined two levels of system reliability. Under low and high reliability, the advice was correct 70 and 30 percent of the time, respectively.

As expected, performance was better under high as opposed to low reliability. Further, it's been found that when the automated system functioned under the good etiquette condition, operators were better able to diagnose system performance regardless of reliability level. Additionally overall levels of trust in the automated system were much higher under good etiquette within the same reliability conditions. Thus, “rude” behaviour made the system seem less trustworthy irrespective of reliability level. Several participants commented that they disliked being interrupted. The Systems designed to conform to rules of etiquette may enhance performance called as system reliability.

Findings gathered with a high criticality simulated system; however, the rules of etiquette (or interruptions) may be equally important for business or home applications. In a recent study, they examined the effects of different levels of communication on task performance with a simulated adaptive interface. Specifically, stakeholders worked with a computer system “partner” to solve problems (e.g., determining the shortest mileage between two cites or estimating gasoline consumption for a trip) using a commercial travel planning software package. In their study, the computer operator was actually in another room who followed a strict set of rules regarding how and when to intervene to help complete a task for the participant. In addition, they followed four different modes of communication that differed in the level of restriction ranging from context sensitive natural language to no communication at all. The results showed that as restrictions on communication increased, participants were less able to complete their tasks, which in turn, caused the computer intervene more often to complete the tasks. This increase in interventions led the participants to rate their interactions with the computer operator more negatively. Hence, these findings suggest that even for less critical systems, poor etiquette makes a poor impression. Apparently, everyone like real performance and ground work even if it is the computer or any automated system incorporating artificial intelligence.

1.5 MANAGING DYNAMIC (RUN TIME) ADAPTIVE AUTOMATION OF STORAGE CLOUD

Adaptive automation is finding its way into commercial and more common technologies of day to day activities. Few examples include adaptive control found on several high-end automobiles and "smart homes" that control electrical and heating systems to conform to user preferences. Also in Managing Data Centre and Cloud storage systems and accessing data as and when required. And not it reached the in the hands of every individual being in the form of smart phones.

There are experiences of an adaptive Data Centre or Cloud Storage. The Data centre is designed to regulate Information processing in a fast and secure way. The automation monitors the data transfer activities and makes inferences about the data patterns behaviour, predicts future needs, and adjusts the bandwidth or speed accordingly. Here when the automation fails to meet the user's expectations, the user can set the controls manually.

The heart of the adaptive Data Storage (or Cloud) is the adaptive control of data centre environment and functions to balance two goals: user desires and provide data as and when required instantly without any delay. Because these goals may conflict with one another, the system uses a reinforcement learning algorithm to establish an optimal control policy. The ACCE (Adaptive Control of Cloud Environment) encompasses a learning controller that selects settings based on current states of demand. The controller receives information about triggering of event that is supported by a cost evaluator. A state estimator generates high-level information about inhabitant patterns and integrates it with output from an occupancy model as well as information regarding levels of data available to make decisions about changes in the control settings. The state estimator also receives input from an anticipator module that uses neural nets to predict which zones are likely to be inhabited within the next seconds. Thus, if the data is moving within the centre, the ACCE can anticipate the route and adjust the data before it arrives at its destination. Hence there may be some observations about experiences with adaptive Data centre and cloud. First, there will be a hypothetical model of the ACCE's. There is a conscious effort to be more consistent in data transfer activities developed a meta-awareness of patterns and recognized behaviour more regular, it facilitated the operation of the ACHE, which in turn, helped it to save energy and maximize service of data access.

Here the value of communication should be understood. Whenever a bug is noticed in the hardware then system should be modified to broadcast a warning message throughout the Data Centre to reset the system. After the hardware problem had been addressed, the warning message should be retained because it provided useful information about what was happened. There may be situations where the user could benefit from being told about consequences of manual overrides.

1.6 CROSS DEVICE TESTING AUTOMATION

There are automation tools that allows you to run application on multiple platforms and saves a lot of time. As these automation tools are Time and effort saving but still manual intervention is required because these tools cannot test the usability and accessibility of applications.

- Cross Browser/OS testing independent of a user's machine
- Testing against older versions of browsers
- Real mobile device testing for mobile web and native apps, both iOS & Android
- Visual Validation for softer "look and feel" testing
- Ability to define test scripts in plain English and translate to any Programing Language code
- Test result reports includes screenshots of failed test cases
- Execution from Windows & OSX or Linux desktops
- Execution from CI platforms such as Jenkins & Bamboo

Emulator Software is used to perform cross platform and environment testing, emulator need to be virtualization. Virtual machine need to be created with different environmental combinations as well as emulator are used to check behaviour of applications.

1.7 FUNDAMENTAL OF ADAPTIVE STORAGE NETWORK

The Adaptive Storage Network is an approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence.

Since the introduction of the first Public Switched Telephone Network, networks have continually evolved and now a days it reached to the level of Software Defined Networking.

There are various stages of development from fixed endpoints to today’s broadband networks that connect mobile users to massive data centers and bandwidth behemoths i.e. Amazon, and Facebook have grown/scales to accommodate continuous demands.

Static infrastructure is going through a profound transformation than ever before. The latest product is autonomous networking with Data Center, which is a trend that has been building from few years. Hence autonomous network runs without much human intervention, it can configure, monitor and maintain itself independently.

Even though it’s a significant advance, autonomous adaptive networking is still needs much attention and development with Artificial Intelligence. Hence further approach is defined to the evolution of networking with the Adaptive Network that’s toward providing a network that can scale with an organizations as their business needs and markets change.

1.8 ADAPTIVE NETWORK VISION

The Adaptive Network is reinvention the network into a dynamic, programmable infrastructure built on analytics and dynamic automation.

The Run Time Dynamic Adaptive Network allows organizations to evolve their current infrastructures into more of a communications loop that relays information from network elements, instrumentation, users, and applications to a software layer for action, rather than blaming the network itself.

1.8.1 THE ADAPTIVE NETWORK INCLUDES THREE IMPORTANT LAYERS

- **Programmable infrastructure:** This includes the physical and virtual elements of network, as well as the related components gathered from them. The adaptive infrastructure layer should be highly intelligent and interprets data so the network can make decisions, whether that means routing traffic around a network that's down or investigating and correcting an issue with delay on a targeted site. Adaptive infrastructure needs a flexible grid; a reconfigurable photonic layer to give the ability to reroute channels of variable spectral occupancy across any path, and across any spectrum in the network from the IP layer correlated with forwarding data. In addition, an adaptive infrastructure needs tuneable coherent transponders to efficiently map a flexible number of client signals to the variable line capacity. In turn, that requires a centralized purpose-built architecture.
- **Analytics and intelligence:** The programmable infrastructure produces massive amounts of data. Big data indicate trends that the network learns and adjusts for over time and approach towards to automated adaptive network. Big data can trigger the network on how to coordinate in the long term, which traffic patterns to look out for, and which parts of the network could be

vulnerable. Even with small data things are happening at a fairly fast pace. It could be an urgent request from a customer. Such important events needs a speedy response from the Data Center network, so those moves can be made by the analytics. Hence once the decisions are made, a human operator or pre-defined policies could step in and approve or change things as necessary. In an actual autonomous network, there would be no external influence but a full adaptive approach.

- **Software control and automation:** It's been found that one cause of network outages is human error. Effective automation of network tasks, such as loading access controllers and provisioning routers, or automated calculation and configuration of tunnels to optimize data traffic and relieve congestion, can eliminate those errors and keep the adaptive network running at peak performance. The ability for adaptive automation to work across multiple sites is critical. Few technologies are good at working with one set of devices from a single vendor, but some networks are built on a single vendor's system. But Networks have to interoperate, using defined APIs, to function efficiently and move data efficiently and swiftly from point to point.

The development of the Adaptive Network is a significant movement for the Data Storage Networking world. Adaptive automation is a cohesive evolution that supports all aspects of intelligent automation, such as intent-based, analytics, and programmable domain control with Artificial Intelligence. It's a micro services-based architecture that delivers extensibility and scale, and it takes an Operations integration approach to provide operational and service agility.

The Adaptive Network is an approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence on network storage cloud.

1.9 AN ADAPTIVE DATA STORAGE NETWORK CAN SOLVE TODAY'S CHALLENGES

The key aspects of the Adaptive Network, as traffic increases and becomes less predictable, effective partnerships will become more critical. As network management conditions are difficult for carriers, with legacy network limitations; and intense competition from new market entrants and evolving business models. An agile and adaptive network can help operators overcome these challenges; and exploit the emerging opportunities like IoT and 5G users.

There is a push-pull from rapid business and technology change affecting users today. On one hand, dramatic growth in subscriber demands are driving front haul and backhaul traffic and putting networks under intense pressure. There's a wide race to develop and commercialize revenue-generating services, such as IoT use cases and 4G-5G mobile services, and to implement the network technologies and architectures needed to support and deliver them. Further new market entrants, including internet companies, are deploying massive-scale network connections that support low-cost data transport between key locations and data centers with unrivalled economies of scale.

The challenges for users are; to take continuously traffic growth in stride; then to prepare the network for the next-generation of Inter of Things and next generation use cases; and then to remain competitive on price with large connectivity providers in the market.

1.10 USES OF ADAPTIVE STORAGE NETWORK:

1) To Increase network agility and efficiency because the Adaptive Network turns the simple network into an automated, dynamic, programmable infrastructure built on analytics and automation, it helps meet growing bandwidth needs with on-demand scalability. As well as helping users to handle incremental traffic growth and unpredictable demand peaks, an Adaptive Network supports real-time scaling and resource-allocation to support differentiated Quality of Services for different applications and use cases, way for commercial next Generation services.

The agility of an Adaptive automated Storage cloud Network also helps user to maximize efficiency by automating a wide range of manual networking processes, from routine service provisioning and turn-up, to discovery and traffic routing over the best available components and paths. So it helps users compete effectively with even the largest connectivity providers.

2) Future-proofing the network with industry leading Data Storage solutions few organizations created the industry's most scalable portfolio of programmable, Data Storage network infrastructure to help operators meet massively growing bandwidth demand up to the edge. Data Storage solutions cover the metro edge, between data centers, the backbone core, and submarine. This Data Storage market leadership, enabled by consistently high R&D investment, is based on our deployed 100G, 200G, and 400G capable in-house modem technology, supported by a unique combination of software intelligence to get the best of optical innovation to the market.

Leading technology portfolio supports continuous convergence of voice and data traffic for mobile operators. This allows users to integrate 4G and 5G traffic in the future, and delivers it extremely cost effectively across a unified infrastructure.

By ensuring that the network can keep pace with exponential increases in bandwidth demands and new services requirements (low latency, high availability) of Data Storage portfolio protects clients' business for the long-term.

3) Helps to avoid vendor lock-in with open networking Many network providers design their portfolios to work together, however, infrastructure is becoming increasingly complex, needing to integrate components and processes in multi-vendor environments. This approach requires large-scale Dynamic infrastructure upgrades which are costly and disruptive, as well as reducing Return on Investment on existing data storage network equipment.

To maximize cost efficiency and value for clients, the portfolio of hardware and software are designed on the principle of openness. This allows users to tie an entire network infrastructure together into a single environment that delivers value for the business and end-customers long-term.

As well as integrating all equipment both legacy and new, can help monitor and manage multi-vendor networks with a centralized, integrated set of tools. This capability is delivered to Manage, Control and Plan (MCP) software, which gives full visibility of resources and services across multi-vendor domains, with tools to troubleshoot and manage diverse infrastructure components remotely.

4) Driving Data Storage network innovation in strategic partnership finally, but equally importantly, there should be strategic partnership among global operators. Based on financial and operational stability, they are able to commit to continual innovation of Data Storage Network portfolio, ensuring that clients can embrace emerging opportunities and take future network challenges in stride.

One example of how investing for the future is recent acquisitions of different technical setup includes of Packet Design, with network performance management software ,focused on Layer 3 Data network optimization, topology and route analytics. By integrating Packet Design into Blue Planet, It is able to extend intelligent orchestration and automation capabilities from layer 0, 1 and 2 into the IP layer. As a result, clients will be able to further optimize service delivery and maximize resource utilization – taking the Adaptive Network to the next level.

1.11 ADAPTIVE INTERFACE TO SCALABLE CLOUD STORAGE

Many of today's applications are delivered as scalable, multi-tier services deployed in large data centres. These services frequently leverage shared, scale-out, key-value storage layers that can deliver low latency under light workloads, but may exhibit significant queuing delay and even dropped requests under high load.

Scalable Cloud Storage is a system that helps these applications adapt to variation in storage-layer performance by treating scalable key-value storage as a shared resource requiring congestion control. Under light workloads, applications using **Scalable Cloud Storage** send requests to the store immediately, minimizing delay. Under heavy workloads, Cloud Storage automatically batches the application's requests together before sending them to the store, resulting in higher throughput and preventing queuing delay. We show experimentally that **Scalable Cloud Storage** adaptation algorithm converges to an appropriate batch size for workloads that require the batch size to vary by over two orders of magnitude. Compared to a non-adaptive strategy optimized for throughput, **Scalable Cloud Storage** delivers lower latency under light workloads; compared to a non-adaptive strategy optimized for latency, **Scalable Cloud Storage** can scale too many requests.

1.12 CONCLUSION

The development of adaptive automation introduces a qualitative leap in the evolution of Storage Networking and Cloud technology. Adaptive automation users will face systems that differ significantly from the automated technology of today. Adaptive Automated Systems in Run time Dynamic Storage Cloud will be much more complex from both the users' and designers' perspective. Adaptive, run time dynamic automated storage systems will need time to learn about users and users will need time to understand the automation. User and his data centre needed some time to adjust to each another. Further, users may find that adaptive systems are less predictable due to the variability and inconsistencies of their *own* behaviour. Therefore, users are less likely to get experience of these systems as tools, process, or even traditional computer programs. Any adaptive Data Storage Centre would respond to demand. **Hence interacting with adaptive systems is more like interacting with a teammate or co-worker**

The challenges for designers of dynamic adaptive systems are significant. Current methods in system analysis, design, and evaluation fall short of what is needed to create systems that have the authority and autonomy to swap tasks and information with their users. Adaptive automated systems require developers to be knowledgeable about task sharing, methods for communicating goals and intentions, and even assessment of user's states of mind. Researchers and designers of adaptive technology need to understand the organizational, and behavioural patterns that impact communication and teamwork among humans to create more effective adaptive systems. In this regard, ideas regarding human-computer interactions may be a mile stone to the development of successful adaptive systems.

So far, most of the dynamic adaptive automated storage cloud systems that have been developed address life critical activities where the key concerns surround the need of the user, the adaptive system itself, and recipients of the dynamic system's services. However, the technology has also been applied in other contexts where the consequences of human error are less severe (e.g., Adaptive Data Storage Cloud). And in plus, adaptive automation could be particularly useful when incorporated in systems aimed at training and skill development as well as entertainment and other knowledge based industries.

To date, most of the adaptive automation systems that have been developed were designed to maximize the user-system performance of a single user. Hence, they are user independent (i.e., designed to improve the performance of any user system). However, overall user-system performance is likely to be improved further if the system is capable of learning and adjusting to the behavioural patterns of its user. Although building dynamic and intelligent storage systems capable of becoming more user-specific might seem like a logical next step, that approach would introduce a new and significant challenge for designers of adaptive automation, addressing the unique needs of multiple users. The ability of Data Storage Cloud to successfully adapt to demand patterns is due in large part to his being the only inhabitant. There are challenges faced by an adaptive system trying to accommodate the wishes of two people who want the temperature set at different levels.

The problem of scaling services with multiple users is not unique to adaptive automation. This challenge arises from a fundamental aspect of humanity. Being social creatures people work in teams, groups, and organizations. Moreover, they are co-located or distributed around the world and networked together. Developers, Architects, Designers of collaborative meeting and engineering software realize that one cannot optimize the individual human-computer interface at the expense of interfaces that support team and collaborative activities. Hence, even systems designed to work more efficiently based on knowledge of brain/intelligent functions must ultimately take into consideration groups of people. Thus, the next great challenge for the neural networks and artificial Intelligence approach may lie with an understanding of how activity of multiple operators in social situations can improve the organizational work environment.

CHAPTER 2

RESEARCH METHODOLOGY

CHAPTER 2

RESEARCH METHODOLOGY

Keywords: Automation Tools, Analysing Tools, Cloud Computing, Heuristic Search, A*, Fuzzy Logic

2.1 QUESTIONNAIRES

i. Work and Responsibilities in Organization in Software Development, Software Testing, Quality Engineering and Software Integration and Deployment
ii. Describe the methods of development and Testing used in organization
iii. Describe the methodologies and status of manual testing
iv. Problems associated with manual testing
v. Problems used with manual testing tools
vi. Repositories used for Test Cases
vii. Uses of Cloud storage in Development and Testing Environment
viii. Methodologies of using Dynamic Storage in Software Environment
ix. Opinion of using automated testing as a solution coming with manual testing.
x. Factors needed to be considered before implementation of automation.
xi. What should be the process of implementing automation?
xii. As per the current situation is there any automated process carrying out doing software build, Testing and Delivery?
xiii. Opinion and suggestions implementing Adaptive Automation with Storage
xiv. Challenges of implementing adaptive automation?
xv. Algorithms used for adaptive software testing tools development in dynamic environment with large cloud storage.

xvi. Overcoming the challenges of implementing adaptive automation using tools and techniques

xvii. Level of satisfaction after implementing adaptive automation in System development, System Testing, Thereafter Integration and Delivery.

2.2 AUTOMATION TOOLS FOR SOFTWARE SYSTEMS TESTING

As per the response and data collected from questionnaires, following tools are listed for study and practical implementation to check the behaviour of adaptive automation. Following tools are chosen for deep study and concluding the Research.

2.2.1 SELENIUM WEBDRIVER

Selenium is an open source library with bindings in multiple languages (Java, C#, Python, etc.) that allows an engineer to write code that is then translated into human-like interactions with various browsers and mobile devices. At its core Selenium spins up a lightweight server on a machine that sends commands in the JSON format to a browser or device. These JSON commands typically include information such as the action to be performed (click, enter text, submit form, etc.) as well as information about how to identify the element for the action to be performed on.

These pieces of identifying information are based on the Document Object Model (DOM) or a web page or app and, for all intents and purposes, can be thought of as the HTML of a page.

2.2.2 SELENIUM

It is test automation framework for Web applications. Selenium is an automation framework of choice for Web automation engineers, particularly for those who possess advanced programming and scripting skills. Selenium become a core framework for other open-source automation tools such as Katalon Studio, Watir, Protractor, and Robot Framework. Selenium supports multiple operating systems (Windows, Mac, and Linux) and multiple browsers (Chrome, Firefox, IE, and Headless browsers). And it can be programmed with scripts can be written in various programming languages such as Java, Groovy, Python, C#, PHP, Ruby, and Perl. Engineers have flexibility with Selenium, can write complex and advanced test scripts to meet various levels of complexity, it requires advanced programming skills and effort to build automation frameworks and libraries for specific testing needs.

It is the most popular automation testing tool for web applications. Selenium can be run in multiple browsers and operating systems**.** It is compatible with other automation testing frameworks.

With selenium, a browser centred automation test scripts can be created which are scalable across different environments. Scripts can be created using Selenium that is of great help for prompt reproduction of bugs, regression testing, and exploratory testing.

2.2.3 ROBOT FRAMEWORK

Robot Framework implements the keyword-driven approach for acceptance testing & acceptance test-driven development (ATDD). Test capability can be extended by implementing additional test libraries using Python and Java. Selenium WebDriver is an external library in Robot Framework. Test engineers can leverage Robot Framework as an automation framework for web testing as well as for Android and iOS test automation. Robot Framework can be easy to learn for engineers who are familiar with keyword-driven testing.

2.2.4 JBEHAVE

JBehave is an open source BDD (Behaviour Driven Development) library that allows users to write their test cases in plain English and have them automatically translated into chunks of Java code to be executed.

JBehave allows someone like a product owner or scrum master to write test cases, hand them off to automation engineers and have those engineers write the automation scripts.

JBehave also creates easily digestible and human readable reports after execution, including information such as what test cases were run, how many test cases passed/failed and provides screenshots for any failed test cases.

Everything in JBehave is customizable and flexible, giving each team the power to define their own test runs and even create custom reports.

2.2.5 RESTASSURED

RestAssured is an open source framework that allows for easy and flexible testing of API based applications.

2.2.6 DOCKER

Docker is an open source tool that allows users to "containerize" applications and environments.

Using a simple Docker image and 1-2 commands a user can instantly deploy an environment on their local machine with a set of predefined conditions such as installed browsers with specific versions, specific applications installed or preconfigured network settings.

2.2.7 TESTNG

TestNG is a lightweight testing framework in between JUnit and JBehave/Cucumber. TestNG is ideal for teams that don't want to deal with the overhead of configuring BDD frameworks or are writing tests (such as API level) that do not lend themselves to BDD concepts such as stories or features.

2.2.8 CUCUMBER

Cucumber is another commonly used BDD library. It is very similar to JBehave but will be more familiar to those coming from a non-Java coding background.

It is designed over the concept of BDD (Behaviour-driven development). It performs the automated acceptance testing by running the stories that best describe the behaviour of the application. It gets a single up-to-date living document that is having both specification and test documentation.

Cucumber is scripted in Ruby. It also supports Java and.NET. It also has cross-platform OS support.

2.2.9 APACHE JMETER

It is Java desktop application designed for load testing. It mainly focuses on web applications. This tool can be used for unit testing and limited functional testing.

Its architecture is centred on plugins with the help of which JMeter provides a lot of out of box features. It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, Java objects, and database. It also includes Test IDE, dynamic reporting, command line mode, portability, multithreading, caching of test results and highly extensible core. It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, Java objects, and database.

2.2.10 APPIUM

This Test automation framework is intended for mobile applications, and automation of native, hybrid and mobile web applications built for iOS and Android. This tool uses vendor-provided automation frameworks and is based on client/server architecture. Appium is easy to install and use.

2.2.11 ROBOTIUM

It is an open-source test automation framework primarily meant for Android UI testing? It supports both native and hybrid applications. Using Robotium, time-saving, readable and easy to use automated grey box UI tests intended for android apps can be written. System testing, functional testing, and user acceptance testing over Android-based apps with the help of Robotium can be performed.

2.2.12 UFT

Unified Functional Testing (UFT) is a commercial testing tool for functional testing. It provides a feature set for API, web services, and GUI testing of desktop, web, and mobile applications across platforms. This tool has advanced image-based object recognition feature, reusable test components, and automated documentation. UFT uses Visual Basic Scripting for testing processes and object control. UFT can be integrated with Mercury Business Process Testing and Mercury Quality Center. This supports CI with Jenkins. It was previously known as QuickTest Professional (QTP).

It brings developers & engineers coming together under one umbrella and provides high-quality automation testing solutions. It makes functional testing less complex and cost-friendly. Its features include **Cross browser & multi-platform compatibility,** Optimized distributed testing, multiple testing solutions, image-based object recognition and canvas, visual test flows.

2.2.13 IBM RATIONAL FUNCTIONAL TESTER

IBM RFT is a data-driven testing platform. It supports applications such as .Net, Java, SAP, Flex, and Ajax. RFT uses VB, .Net and Java as programming languages. RFT has a feature called as Storyboard testing in which users' actions on AUT are recorded and visualized in a storyboard format through application screenshots. It can be integrated with IBM Jazz application lifecycle management systems such as IBM Rational Team Concert and Rational Quality Manager.

This tool is intended for **automated functional testing & regression testing**. It allows to perform data-driven and GUI testing. The automated testing is based upon script assure technology which highly improves the efficiency of testing and provides easy script maintenance. This tool does automate performance testing over web and server-based apps. It has RCA capabilities to remove performance bottleneck. It provides real-time reporting and test data customizations. It also offers load and scalability testing.

2.2.14 TESTCOMPLETE

Environment Supported are web, mobile, and desktop testing. Programming/scripting languages support: JavaScript, VBScript, Python, and C++Script. Testing performed: keyword-driven and data-driven testing with Test Complete offers easy-to-use record and playback feature. Like UTF, TestComplete's GUI object recognition capability can automatically change with UI objects which helps reduce the effort to maintain test scripts when the AUT is changed. It can be integrated with Jenkins in a CI process. This tool works for desktop, mobile and web applications. With Tet Complete, Functional UI tests can be built and run via robust record & replay capabilities or by scripting in your favorite languages, including Python, JavaScript, VBScript with support for applications, such as .Net, and native and hybrid iOS and Android apps, along with regression, parallel, and cross-browser testing capabilities.

2.2.15 HP QUALITY CENTER

HP Quality Centre software standardizes testing. It is an integrated IT quality management software application. Automated testing is one of its key features which constantly allows to test earlier and quicker. Sharing with reusability allows HP QC to have bug-free and reliable applications.

2.3 APPROACH AND STEPS OF STUDY/ ANALYSING RIGHT TOOL

1. Market Research: I have done Search for tool available whether Free Open Source, Community versions or Paid one fit in the requirements. To study open source automated tools and based design algorithm and methodologies and simulation of code base libraries of adaptive nature intelligent enough to understand their execution environment.

2. Experts View: Get feedback from the users and experts or from Forum of experts to get experiences on the features of tools. To take input from the users for their feedback to make the algorithm in a way so that it simulates itself as per the nature of protocol and requirements

3. Personal Experience: As per my personal work experience working with tools I have done research shortlisted best tools that best suits requirements and affordability of adaptive tools.

4. I Prepared comparison chart and done SWOT Analysis to select the best tool works on the principle of Run Time Dynamic Adaptive Automation. Studied the level of satisfaction/experience of users before using automated (adaptive design and test pattern algorithm) Build and Test of software applications. Then Study the level of satisfaction of users after using automated (adaptive) software applications.

5. As per the comparison and study of tools and methodologies find problems in application of Adaptive Software in Technical Organizations and designed an engineering algorithm in terms of model and methods for assembling software systems that can

dynamically adapt to context and can ‘account for them-selves’ specific to run time dynamic environment.

6. Then done minimizing tools for the requirements fits exactly we have to get through above mentioned guidelines.

2.4 STUDY ON CLOUD COMPUTING TERMS (TYPES AND DEFINITION, SERVICES)

There are few concepts those need to be addressed if cloud computing been used for Dynamic Run Time Adaptive Automation and Data Access/Storage.

Definition of "cloud computing"

As per definition provided by NIST (National Institute of Standards and Technology) which is summarized as follows: Cloud computing model is for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be provisioned/released with minimal effort or service provider interaction.

The full definition and explanation can be found in this public NIST document. Additionally, NIST has published a final version of cloud computing technology roadmap.

Different types of clouds and distinction between them.

There are four kinds of cloud deployment models: private clouds, community clouds, public clouds, and hybrid clouds.

A private cloud infrastructure is provisioned for exclusive use of organizations. It’s owned, managed, and operated by organizations, by third parties, or combination of them. It may exist on or off organizations premises. A community cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It can be owned, managed, and controlled by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises. A public cloud infrastructure is open to use by the public. A public cloud may be owned, managed, and operated by a business, academic organization, or government organization (or some combination of the three). The infrastructure for this type of cloud exists on the premises of the cloud provider.

A hybrid cloud is a combination of two or more separate cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology which enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Definition of "SaaS" or "PaaS". And a "Service" with respect to cloud.

A service is something that provides "value" to the organization's lines of business. There are mainly three basic service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Another service model (that is not officially defined by the National Institute of Standard and Technology) is Database as a Service (DaaS).

Different types of service models.

Software as a Service, or SaaS, is the delivery of applications that are developed and hosted by the SaaS vendor. The end user accesses the applications over the Internet.

Unlike traditional packaged applications the SaaS vendor owns the software and runs it on computers in its data centre. The customer does not own the software but can rents it, usually for a scheduled time fee.

Platform as a Service, or PaaS, is a model in which the user creates an application or service using tools and/or libraries from a provider. The provider provides the networks, servers, storage, operating systems, and other services that are required to host the consumer's application or service.

Although less established and not as widely available as compared to SaaS and IaaS offerings, ready-to-use application platforms offer great promise for organizations that aren't compelled to own and manage the underlying infrastructure. A well-known platform that uses the PaaS model is Google App Engine.

Infrastructure as a Service, or IaaS, is the virtualized processing, storage, and networking services along with automation and management capabilities in this area offer the most flexible level of services in the cloud computing model.

A well-known platform that uses the IaaS model is Amazon Elastic Compute Cloud.

Database as a Service, or DaaS, is a cousin of SaaS. It is a physical data management strategy for managing an on-site private cloud made up of several different database architectures.

These architectures can provide varying degrees of database service to an application based on business requirement's related to availability, scalability and performance.

The main objective of the DaaS strategy is to provide a high-quality database service while maximizing ROI on database software/hardware and supporting agile development efforts through faster service delivery.

2.5 PATTERNS/ALGORITHMS FOR ADAPTIVE SYSTEM DESIGN, DEVELOPMENT AND TEST

As per the data collected from questionnaires with regards to adaptive algorithms, a study is done and resulted with a simulated adaptive algorithm based on Search Algorithms, Informed Heuristic Search strategies and Fuzzy Logic. These approaches mentioned in algorithms can be used to develop adaptive software and tools for analysing and doing run time decision based on adaptive simulation and learning from environment and feedback mechanism.

2.5.1 INFORMED (HEURISTIC) SEARCH STRATEGIES

To solve big problems with number of possible states, problem-specific knowledge function must be added to increase the performance of search strategy algorithms.

2.5.1.1 HEURISTIC EVALUATION FUNCTIONS

Calculates the total cost of optimal path between two states of nodes/set. For example, a heuristic function for sliding-tiles games is processed by calculating number of moves that every tile makes from its goal state and then summing up all.

2.5.1.2 PURE HEURISTIC SEARCH

It counts heuristic values for every node then expands them in the order. In this process two lists are created, a closed list for the expanded nodes and an open list for the defined but unexpanded nodes.

Then In each iteration, a node having minimum heuristic value is expanded, and all its child nodes are created/placed in the closed list. Then, the heuristic function is applied to the child nodes to place them in the open list according to their heuristic value. And in the End the shorter paths are saved, and the longer ones are disposed.

2.5.1.3 A * (BEST FIRST) SEARCH

It is form of Best First search because it avoids expanding paths that are already expensive but expands most promising/optimized paths first.

The function for A* Search is defined as follows

f(n) = g(n) + h(n), where

- g(n) the cost (so far) to traverse the node
- h(n) expected cost to get from the node to the result
- f(n) estimated total cost of path through n to result. priority queue is used to implement same by increasing f(n).

2.5.1.4 GREEDY BEST FIRST SEARCH

It expands/executes the nodes that is estimated to be closest to goal. It expands nodes based on f(n) = h(n). It is implemented using priority queue and only estimated cost to get from the node to the goal is calculated.

Disadvantage − It can get stuck in loops. It is not optimal.

2.5.2 FUZZY LOGIC

Fuzzy Logic Systems (FLS) gives acceptable but definite output even if incomplete, ambiguous, distorted, or inaccurate (fuzzy) input is given.

2.5.2.1 WHAT IS FUZZY LOGIC?

Fuzzy Logic (FL) is a way of reasoning that resembles human understanding. The approach of FL concludes the way of decision making in humans that involves all interim possibilities between digital values YES and NO.

The conventional logic block takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human’s YES or NO.

The inventor of fuzzy logic, Lotfi Zadeh, observed that, the human decision making is different from computers which includes a range of possibilities between YES and NO, such as –

CERTAINLY YES
POSSIBLY YES
CANNOT SAY
POSSIBLY NO
CERTAINLY NO

Table 2.1: Decision Possibilities

Hence the fuzzy logic works on the different levels of possibilities of input data to get the definite output.

2.5.2.2 IMPLEMENTATION

- It's suitable for systems with various sizes and capabilities from small microprocessors to large, networked, workstation-based control systems also include networked storage cloud.
- It can be implemented in hardware, software, or in embedded systems.

2.5.2.3 NEED OF FUZZY LOGIC?

It is useful for many commercial and practical purposes.

- It can control machines and consumer products.
- It may not produce accurate reasoning, but acceptable reasoning to get the work done.
- Fuzzy logic helps to deal with the uncertainty in engineering design and execution by working on the set of acceptable defined data.

2.5.2.4 FUZZY LOGIC SYSTEMS ARCHITECTURE

It has four main parts as shown −

- **Fuzzification Module** − It converts the inputs, which are defined numbers, into fuzzy sets and splits the input signal x into five steps may be defined as

LP	x is Large Positive construct
MP	x is Medium Positive construct

S	x is Small construct
MN	x is Medium Negative construct
LN	x is Large Negative construct

Table 2.2: Possibilities Construct

- **Knowledge Base −** It creates IF-THEN rules provided by system designers.
- **Inference Engine −** It defines and simulates the human understanding process by making fuzzy inference on the inputs and IF-THEN rules, combining.
- **Defuzzification Module** – It processes the fuzzy set obtained by the inference engine and transformed into an exact value.

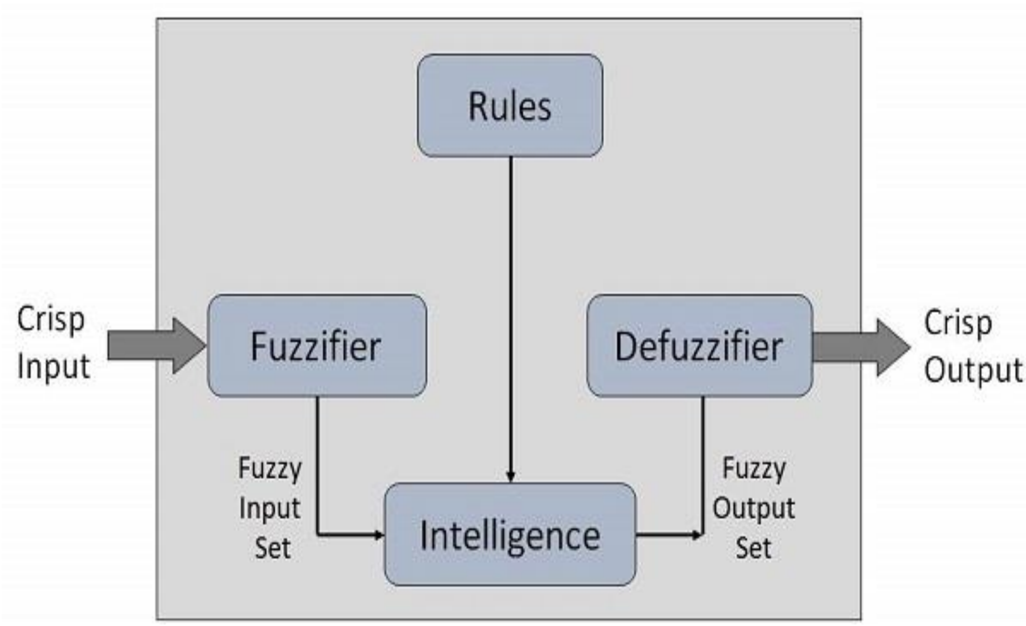


Figure 2.1: Fuzzy Logic System Architecture

The **membership functions work on** fuzzy sets of variables and Input.

2.5.2.5 MEMBERSHIP FUNCTION

Membership functions quantify linguistic term and presents a fuzzy set diagrammatically. A **membership function** for a fuzzy *set A* on the universe of X is defined as μ_A: X → [0, 1].

Each element of fuzzy set is mapped to a data between 0 and 1. This is called **degree of membership**. Membership Function quantifies the degree of membership of the element in *X* to the fuzzy set *A*.

- X axis represents the universe of discourse.
- Y axis shows the degrees of membership in the interval.

Multiple membership functions can be defined applicable to fuzzify a numerical value. Hence simple membership functions are used because use of complex functions does not add more precision in the output.

All membership functions are shown as below in graph diagram.

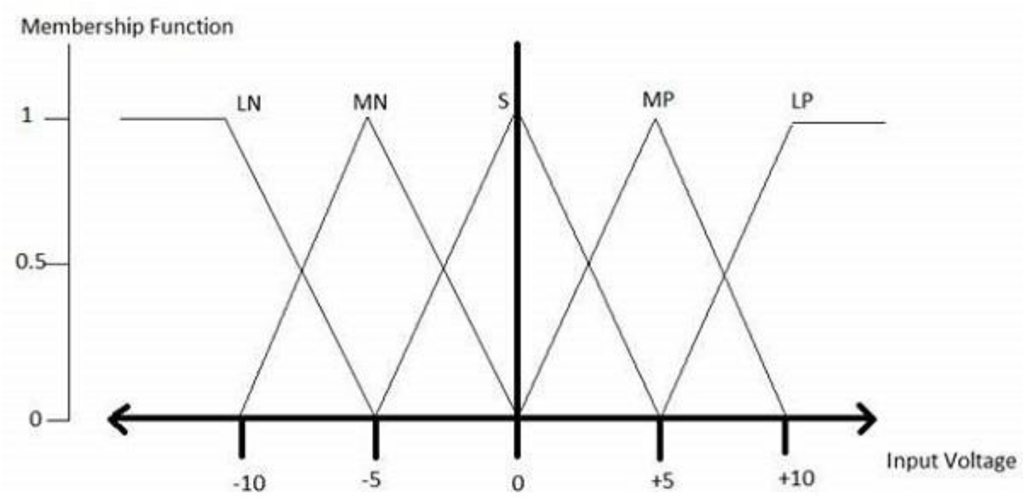


Figure 2.2: Membership function

The triangular membership function shapes are more frequent among various other membership function shapes.

2.5.2.6 ALGORITHM

- Define linguistic variables and terms.
- Construct membership functions for them.
- Construct knowledge base of rules.
- Convert scattered data into fuzzy data sets with functions. (fuzzification)
- Evaluate rules in the rule base. (Inference Engine)

- Combine results from each rule. (Inference Engine)
- Convert output data into non-fuzzy values. (de-fuzzification)

2.5.2.7 LOGIC DEVELOPMENT

Step 1: Define linguistic variables and terms

Linguistic variables are input/output variables may be in the form of known words or sentences to form simple constructs. For example, room temperature, may be categorized as, cold, warm, hot, etc.

Temperature (t) = {very-cold, cold, warm, very-warm, hot}

Every component of this defined set is a linguistic term which can cover some part of overall temperature values.

Step 2: Construct membership functions for them

The membership functions of temperature are represented as below.

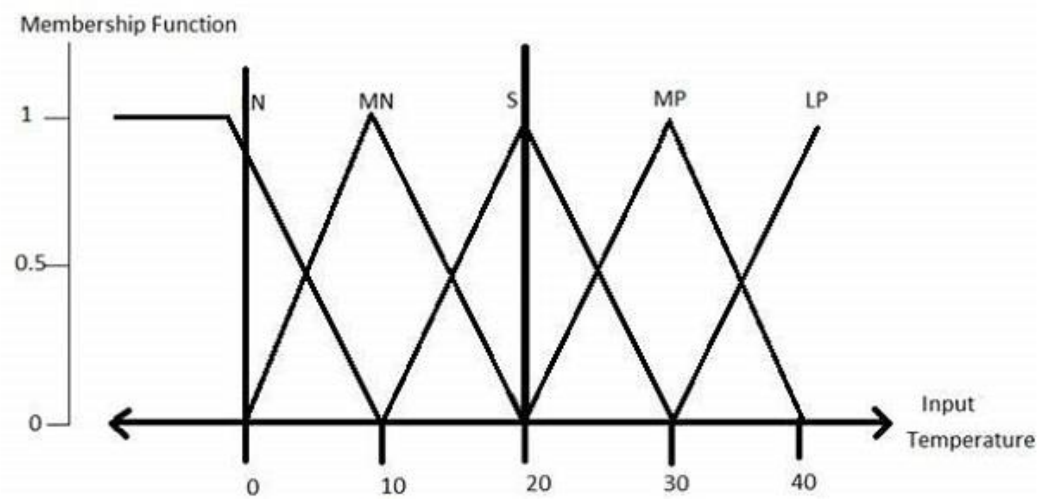


Figure 2.3: Membership Functions Constructs

Step3: Construct knowledge base rules

To construct knowledge-based rules a matrix of room temperature values versus target temperature values is created that an air conditioning system is expected to have.

RoomTemp. /Target	Very_Cold	Cold	Warm	Hot	Very_Hot
Very_Cold	No_Change	Heat	Heat	Heat	Heat
Cold	Cool	No_Change	Heat	Heat	Heat
Warm	Cool	Cool	No_Change	Heat	Heat
Hot	Cool	Cool	Cool	No_Change	Heat
Very_Hot	Cool	Cool	Cool	Cool	No_Change

Table 2.3: Target Room Temperature Decision Table

Build a set of procedure or regulations into the knowledge defined construct in the form of conditional IF-THEN-ELSE structures.

Sr. No.	Condition	Action

1	IF temp= (Cold OR Too_Cold) AND target=Warm THEN	Heat
2	IF temp= (Hot OR Too_Hot) AND target=Warm THEN	Cool
3	IF (temp=Warm) AND (target=Warm) THEN	No_Change

Table 2.4: Condition and Action Table

Step 4: Obtain fuzzy value

Fuzzy set operations perform evaluation of rules. The operations used for OR/AND are Maximum and Minimum. Combining all results of outcome to get a result. This result is a fuzzy value.

Step 5: Perform defuzzification

Then Defuzzification is performed according to membership function with output variable.

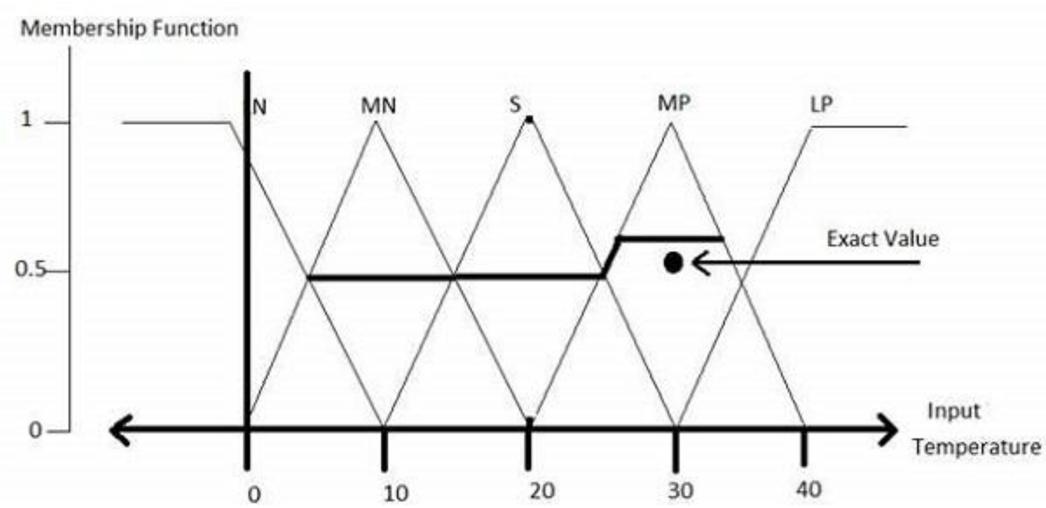


Figure 2.4: De-fuzzification Membership Function

2.5.2.8 ADVANTAGES OF FLSS

- Within fuzzy reasoning Mathematical concepts are simple to understand.

- FLS can be modified by altering rules due to flexibility of fuzzy logic.
- Fuzzy logic Systems can also consider imprecise, distorted, noisy input information.
- FLSs are easy to construct and understand.
- Fuzzy logic is a solution to complex problems in all fields of life, specifically run time dynamic systems as it works on the concepts of human reasoning and decision making.

2.5.2.9 DISADVANTAGES OF FLSS

- Lacking systematic approach to fuzzy system designing.
- Are understandable only when defined in easy terminologies.
- Suitable for the systems which do not need high accuracy as output.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3

LITERATURE REVIEW

KEYWORDS

Cloud Computing, Cloud Services, Cloud Network, Cloud Database, Cloud Management, Storage Devices, Data Storage, Adaptive Automation, Adaptive Automation, Real Time Systems, Run Time, Dynamic,

3.1 INTRODUCTION

Before Cloud, There was a boom on Database applications. But Cloud not only grows rapidly but also developed with variety of applications and services like Platform, Services and Products.

In this chapter, a review of literature is discussed in order to provide a theoretical background and to develop an understanding of the significance and role of Adaptive Automation of Software Applications in cloud,

The focus should be confidentiality integrity and availability of data in cloud. This study aims at presenting a broad introduction to cloud computing, Applications Development and Automated Testing challenges and opportunities in cloud.

It also tracks the background of cloud computing by surveying the main technological spreads that significantly contributed to the advent of this evolving technology, with the objective of clarifying the misperception over the “innovative or evolutionary” cloud computing technology. In addition, this chapter presents review of the research work.

Cloud services deployment models and related topics are dominating the IT landscape.

Many organizations actively addressing these deployment models and has developed an approach to enable to leverage them in a consistent manner to meet business needs.

Before delving into the cloud services deployment models and their security considerations a

Distinction must be made between cloud services offered by Cloud service providers and those offered by third parties.

Web-based network management revolving about database for network information processing and has the characteristics of broader distribution, full interactivity, real-time dynamic, run time and so on in the application; and it is beneficial for network performance and rapid fault recovery.

For this purpose cloud management system plays important role. Cloud Computing is a general term used to describe a network based computing over the Internet. Cloud Computing is basically a step up from Utility Computing and much more includes collection/group of integrated and networked hardware, software and Internet infrastructure (called a platform), Using the Internet for transport which provides hardware, software and networking services to defined clients. Here platforms hide the complexity and details of the underlying infrastructure from users and applications by providing graphical interface (GUI) or Applications Programming Interface (API). The cloud is used as a storage location can be accessed and computed from anywhere. Web application makes the use of distributed storage solution in order to scale up and expand.

There are some important literature and views on Cloud computing from different authors and researchers:

National Institute of Standards and Technology (NIST) gives definition of the Cloud:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Most of the organization uses the definition provided by NIST (National Institute of Standards and Technology) which can be found here: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Elbaum et.al reported that in regression testing feedback may play positive or negative role in test case prioritization. Do et.al presented an empirical study on assessing the ability of prioritization techniques to improve the rate of fault detection of test case prioritization techniques, measured relative to mutation faults in. The major difference between this work and other test case prioritization techniques is the prioritization is carried out on-the-fly as testing proceeds, which means testing history information is collected and used for future decision making. Other related studies include defect removal and its impact on software testing. Okamura proposed a new reliability estimation method that considers defect removal. This study presents a rigorous model for the defect removal process and its impact to the software under test, and developed the according methodology for testing and parameter estimation.

3.2 IMPORTANCE OF PROPOSED INVESTIGATION

The problems in building systems that must exhibit robustness to a changing environment; embedding significant COTS/Community Sourced independently evolving components; problems building systems that involve user scripting and ‘plug-ability’.

In proposed design the researcher will study to evaluate and estimation of the run time dynamic network characteristics and behavioural changes occurred due to applications fine tuning its network resources on server platform due to various reasons.

Here the importance is make and work the system using one interface application adaptive to environment so that the estimation matrix database from testing environment can be exported to the automated testing tool work on adaptive methodology and adaptive approach and same can be applied to change test strategies to finalize inputs of the network component under test (network node, NIC, HBA, etc..) With respect to the run time characteristics obtained thorough feedback mechanism.

There are various plugs and play hardware devices i.e. USB devices. Same way there should be plug and play software systems and it is possible only if software is designed on adaptive patterns. The current proposed investigation will try to find out how the adaptive patterns be implemented in automated software systems. In particular problems building the sort of systems are called on to construct all the time. I need to develop engineering models and methods for assembling software systems that can dynamically adapt to context and can ‘account for themselves’. Here the Software development is no longer garage ‘design and make’. Most of the times software products and services are embedded in a network of complex inter-product and inter-supplier dependencies. As software is the product of the operation of a 'supply chain' that must be designed and forms part of an 'ecosystem' that must be accommodated. Now rethinking software production requires a new discipline of business model and software system co-design.

Therefore Importance of adaptive system varies on the basis of application and environment and it will come with design algorithm and methodologies and code base libraries of adaptive nature intelligent enough to understand their execution environment and input from the user so that program simulate itself as per the nature of (storage networking) protocol, High Performance Networking Systems and can give adequate result/throughput for high performance storage and control systems.

3.21 FOLLOWING PROBLEM SOURCES PUT FORCE ON IMPORTANCE OF PROPOSED INVESTIGATION

- Improper Analysis: happens mostly due to miss-communication between end-users and system analysts.
- Inconsistent System Requirements Specification of automated applications.
- Incomplete or Obsolete or Ambiguous Design of automated Systems.
- Incomplete, Inconsistent Automated Test Plan.
- Improper Feature, Load Tests.

- Continuously changing run time dynamic networking environment.

3.3 SCOPE OF THE PROPOSED STUDY

Due to complexities in modern high performance computing and high performance storage networking systems because of using multi-layered protocol for data transfer on high speed (e.g. Fiber Optics, Ethernet) medium, becoming more complex over the years. As a result there has been as need for automation of functions to cope with this increased complexity. Although several recommendations have been made to develop some sort of automation frameworks and thereby using of various programming techniques for developing automated programs those can be used across platforms.

These techniques has started in use with some new human factors issues and concerns, for example, the ability to intervene effectively when an automated subsystem fails is one of the key issues in automated control systems. Other difficulties that users of automated systems may face include loss of system awareness and manual skills degradation. These kinds of problems may be characteristic of complex systems in which automation is implemented in a fixed or "static" manner.

Alternatively, systems in which automated process are implemented dynamically (run time), in response to changing task demands on the user, may be less vulnerable to such problems. It has been proposed that systems with adaptive automation are superior to conventional automation because they provide for regulation of user workload and vigilance, maintenance of skill levels, and task involvement.

Adaptive automation has been proposed as a means for further increasing the number and flexibility of levels of automation in the high performance networked systems. One outcome of this concept is that the adaptive system could select its own level of automation, depending upon the operating environment and system performance. The feasibility of this approach and its impact on user and system performance is poorly understood. At present, the only consensus in the design of adaptive automation systems seems to be that the philosophy of the user's role in the system will be critical.

Typically, it is argued that the user must interact with the automation as a control system, and as such provides consent to the level of automation assumed by the system. For adaptive (Learning) automation to work, the platform/system must utilize an accurate user-state classifier for the real-time assessment. User state classifiers, means as discriminate analysis and artificial neural networks or learning algorithms.

How an accuracy of 70% to 85% in real-time? To properly implement adaptive automation is a way to figuring out how big a workload needs to be to require intervention. There has been various effect/essence of Adaptive Automated Testing Tools and Test Strategies in run time dynamic network testing environment. This proposed investigation does a critical Study and performance/usability statistics of various Implementations of automated (open source) software testing tool used in high performance different/networking environment/technologies. Further study tries to understand the applications and design of open source automation frameworks and their selection criteria in Technical Organizations to design automated software testing and Adaptive Automated Test Strategies

In proposed design and study, there will be an evaluation and estimation of the network characteristics and behavioural changes occurred due to applications fine tuning its network resources on server platform. To make the system adaptive to environment the estimation matrix database will be exported to the testing tool and adaptive approach will be applied to change test strategies to finalize matrix database of the network component under test (network node, NIC, HBA, etc..) With respect to the run time characteristics obtained thorough feedback mechanism.

The peripheral situations affect the performance of systems; therefore, though one-shot human-cantered automation (HCA) designs might provide better outcomes than the systems designed, based on the "automate it as possible" philosophy.

3.3.1 FOLLOWING POINTS ARE KEPT IN MIND WHILE STUDYING

- Scalability: Devise a software and hardware architecture that scales up by some factor. That is, an application's storage and processing capacity can automatically grow by a factor of a million, doing jobs faster just by adding more resources.

- The Turing Test: Build a computer system processing task in less time.

- Trouble-Free Systems: Build system used by people each day and yet administered and managed by a single part-time person.

- Secure System: Assure that adaptive automated algorithm automate the system in a way that only services authorized users, service cannot be denied by unauthorized users, and information cannot be stolen

- Always Up: Assure that the system is always available.
- Automatic Programmer: which devise a specification language or user interface that
- Easy implementation for people to express system designs.
- Computers can compile/Interpret, and can describe all applications

- The system should learn about application, should asking questions and configure accordingly about exception cases and incomplete specification. And it should not be onerous to use.

- Formalizing Common Sense for minimum input consideration
- Machine Readable Specification and design
- Automatic Code Verification: i.e. Profiling Tools
- Automatic Feature and Load Testing
- Automatic Action Tracking: i.e. Switching Throughput
- Automatic Problem discovery and Recovery

3.4 REVIEW OF WORK (LITERATURE) ALREADY DONE ON THE SUBJECT

In our research work, it is proposed to discuss how an adaptive algorithm can simulate itself according to system environment and its parameters. Now As per review of literature, researcher needs to see how much work already done in this area.

Covering Following areas for Literature review:

- Neuro-ergonomics, computerized adaptive test and Level of Automation
- Dynamic (Run Time) Storage Allocation
- Automated System Testing
- High Performance Cloud and Storage Network

3.4.1 NEURO-ERGONOMICS, COMPUTERIZED ADAPTIVE TEST AND LEVEL OF AUTOMATION

3.4.1.1 EVALUATING KNOWLEDGE STRUCTURE-BASED ADAPTIVE TESTING ALGORITHMS

As **Huey-Min Wu, Bor-Chen Kuo and Jinn-Min Yang** developed a theoretical mathematical model called computerized adaptive test (CAT) for the students. Hence adaptive algorithm based on knowledge structure and called it, knowledge-structure-based adaptive testing (KSAT) algorithms. As per their findings the following are several major interfaces of system. The user management interface in is multi-functional. It allows new users to have access to creating new user accounts, creating multiple new user accounts, importing accounts from other sources such as Excel, and giving access to the database. The test administration interface displays the items and allows the examinees to answer the items presented. Since the KSAT system is an adaptive test, only one item per screen is presented.

The group profile interface in displays the group result of the exam. For example, in some concept of the interface, x students passed and y students failed test. Instructors can then take this information and understand the distribution of students' knowledge states and identify the strengths and weaknesses within a group. This information can be utilized for remedial instruction. Upon completion of the test, the student receives a personalized profile including name, scores, percentile, utilization of test items, date taken, and so forth.

3.4.1.2 STRATEGIES TO IMPLEMENT ADAPTIVE AUTOMATION

There are some strategies used to implement adaptive automation mentioned by **(Morrison & Gluckman, 1994; Rouse & Rouse, 1983).**

One set of strategies addresses system functionality. Where, full tasks can be allocated to either the system or the user, or a specific task can be partitioned so that the system and user each share responsibility for unique portions of the task. Otherwise, a task could be changed into a different format/shape to make it easier (or more challenging) for the user/user to perform/use.

A second set of strategies propagate the trigger for shifting among modes or levels of automation (Parasuraman et al., 1992; Scerbo, Freeman, & Mikulka, 2003). One approach relies on goal-based strategies. Particularly, changes among modes or levels of automation are triggered by a set of criteria or external events. Thus, the system might invoke the automatic mode only during specific tasks or when if it detects an emergency situation. Another approach may be to use run-time measures of user performance to invoke the acceleration (modification) in automation.

A third approach uses models of user performance or stress to drive the adaptive logic in system (Hancock & Chignell, 1987; Rouse, Geddes & Curry, 1987, 1988). **i.e.,** a system could estimate current and future states of a user's activities, ideas, his resources, and overall performance. Information about the user, the system being operated, and the outside world could then be interpreted with respect to the user's goals and current actions to determine the need for adaptive aiding. Finally, psycho-physiological measures that reflect user workload can also be used to trigger changes among modes.

3.4.1.3 ATTENTION ON AUTOMATION AT WORK

Billings (1991) focuses attention on automation at work, about how automation may correctly perform some activities or parts of activities, or how automation can interact with humans or support them in their tasks.
Billing (ibidem) defines Level of Automation in functional terms; "a level of automation corresponds to the set of function that a user can autonomously control in a standard situation united to system ability at providing solutions, and acting properly according to the proposed solution, and to check the results of its actions." As par with Billings's definition Rouse's makes observations (1988) that the adaptive automation provides variable (changing) levels of support to human control activities in complex systems, according to the situation. Moreover, the situation is defined by the task features and by the psychophysical status of human user. As a result, the human machine interaction should check on what has to be automated, and on how and when.

3.4.1.4 PERFORMANCE EFFECTS OF DYNAMIC FUNCTION ALLOCATION

There are studies reviewing empirical researches about **AA (Parasuraman, 1993), (Hilburn et al., 1993), (Scallen et al., 1995), (Parasuraman et al., 1996), (Kaber, 1997), (Kaber & Riley, 1999)** that focused on the performance effects of Dynamic Function Allocation in more complex systems, particularly monitoring and psychomotor functions. These studies makes evidence that AA significantly improves monitoring and tracking task performance in multiple task scenarios, as compared to static(controlled) automation and strictly manual control situations.

Another development for Adaptive Automation is the Neuroergonomics approach, which uses psychophysiological techniques to propagate changes in the state of automation. Studies resulted that this approach can helps user performance **(Scerbo, 1996).** Less work has been done to establish the impact of Adaptive Automation on cognitive function performance (i.e. Decision-making) or to make comparisons of human-machine performance when Adaptive Automation is applied to various information processing units (Kaber et al., 2002). **Scerbo (1996)** mentioned that the Adaptive Automation can start different types of automation strategies, in relation with the context (system and user).

An integration to this conclusion is provided by Kaber and Riley (1999), which defined adaptive automation as a programming of the control assignment between human/user and system, in order to improve the user performance. Human performance is a crucial aspect of the functioning of complex systems. Thereby, the human user should be involved in the controlled task, in order to avoid the out-of-the-loop performance of system.

As stated by Norman (1989), without necessary feedback people are indeed out-of-the-loop sometime; they may not know if their requests have been received by system, if the tasks are being performed properly, or if problems are repeating. Sharing the functions control is quantitative task to accomplish, also it involves the responsibility of the whole operation execution.

The dynamic function allocation (DFA) is a different aspect of Adaptive Automation (Kaber et al, 2001). It basically consists of assigning the authority on specific functions to either the human user or the automated system, depending on the overall context (i.e. user's state and outer conditions) and on a defined set of criteria. DFA should be designed by taking into consideration both the user (human) and the system (machine) status, and the means for allowing context recognition.

Focusing on the involvement and the autonomy that humans and machines have in each task to be performed is subject matter of debate. Few researches face the crucial issue of the authority that each part should have in controlling the system. Historically, humans played Human-Computer Interaction, New Developments the role of the supervisory control i.e. the machine decides about the tasks and the humans evaluate these decisions; depending on this checks, control on the actions is either regained by human users or provided (Sheridan, 1992).In this effort a crucial role is played by the human skills and abilities and by the systems natural limits (Parasuramanet al., 2000).

3.4.1.5 DIFFERENCE BETWEEN THE ADAPTIVE AUTOMATION APPROACH AND THE LEVEL OF AUTOMATION

There is a difference between the Adaptive Automation approach (The way it is carried out) and the Level of Automation (Kaber & Endsley, ibidem) with the traditional view of automation that is a fixed and regulated process designed to eliminate human interaction with System, or Adaptive Automation is designed to expect and anticipate changes under control of an engineer while maintaining precise control of all background variables may not of interest at present (Kay, 2006).

Adaptive Automation is based on the dynamic allocation of the control of the complete task or of some parts, crossing along with time, on manual and automated phases. The Levels of automation allow only a static function assignment, because the task level of automation is established in the design phase (Kaber & Endsley, ibidem).

Adaptive Automation allows users to experiment with variables seen as key parameters in a system while preventing undesired secondary effects that could unexpectedly arise from variations in parameters not under study, which in manual systems might not be perfectly controlled. The Adaptive Automation Design Consideration from particular point of view it is possible to state that Work systems perform functions or units of work. Roles, instead, are more difficult to define. It's a sense to consider as an activity that can be performed either by human or machine (Harrison, Johnson, Wright, 2001).

3.4.1.6 THEORETICAL METHODS TO DEFINE FUNCTIONS, RULES AND SCENARIOS

The York Method (comes out from the Department of Computer Science, at University of York), It provides theoretical methods to define functions, rules and scenarios, and then represents them by specific grids. The main aim is to decide the suitability of functions with corresponding rules, in the light of different scenarios **(Calefato, Montanari, and Tango 2007)**.

"A function may be separated from all roles, and technically feasible and cost effective to automate, in which case the function may be totally automated.

In other way it is possible that the function maps entirely to one of the roles, and is infeasible to automate, in which case the function is totally performed within that role. In most of the cases however functions fit into neither category. In this situation the function might be partially automated" **(Harrison, Johnson, Wright, 2001)**. Hence defined Functions and corresponding roles have to be set into one or more scenarios.

Taking into consideration the driving scenario, it has to be measured the driver's competences in tasks critical to performance and safety? These concept can be known by an example belonging to the automotive domain. We can have to design a preventive/safety system. In order to design/develop the application, the driving scenario and its corresponding manoeuvres have been broken down into functions and sub-functions in order to outline which functions have to be performed manually, automatically or both. Secondly, system and user's roles have been combined with functions in order to outline which functions corresponds to roles, basis on the given scenarios. The given scenarios are selected in order to measure the user workload and situation awareness. Consequentially the selected scenario shows the whole behaviour of the system, along the seven Level of Automation implemented **(Calefato, Montanari, and Tango 2007)**.

3.4.1.7 ADAPTIVE CONTROL OF HOME ENVIRONMENT

Mozer (2004) described experiences living in an adaptive home. The home designed to regulate air, water, temperature and lighting. The automation monitors the inhabitant's activities and makes inferences about the inhabitant's behaviour, predicts future needs, and adjusts the temperature or lighting accordingly. Whenever the automation doesn't meet the user's expectations, the user can set the controls manually whenever required.

The heart of the adaptive home is the **adaptive control of home environment (ACHE)** and functions to balance two goals:

1) User desires and

2) Energy conservation.

Because above mentioned two goals may conflict with one another, the system uses a reinforcement learning algorithm to establish an optimal control policy. For lighting, the ACHE controls multiple, independent light fixtures, each with multiple levels of intensity. The ACHE encompasses a learning controller that selects settings based on present state. The controller receives signals about an event change that is supported by a cost evaluator. A state estimator propagated high-level information about inhabitant patterns and integrates it with output from an occupancy model as well as information regarding levels of natural light available to make decisions about changes in the control settings. The state estimator receives input from an anticipator part that uses neural nets to predict which zones are likely to be inhabited within the next couple of seconds. Thus, if the inhabitant is moving within the home, the ACHE can guess the route and adjust the lights before he arrives at his destination. Mozer (2004) recorded the energy costs with costs of discomfort (i.e., incorrect predictions and control settings) for a month and found that both decreased/converged within about 24 days.

Mozer (2004) had some observations about his experiences living in the adaptive house. First, he found that he generated a designed model of the ACHE's of his activities. Thus, he knew that if he need to work late at the office, the "house" would be expecting him home at the usual time and he often felt compelled to return home! Further, he admitted that a conscious effort to be more consistent in his activities. He developed a meta-awareness of occupancy patterns and recognized that made his behaviour more regular, it facilitated the operation of the ACHE, which in turn, helped it to save energy and maximize his comfort. Mozer (2004) also discovered the value of communication. At one instance, he found a bug in the hardware and modified the system to broadcast a warning message throughout the house to reset the system. As soon as the problem related to hardware had been addressed, however, he retained the warning message because it provided useful information about how his time was being spent. He mentioned that there were other situations where the user could benefit from being told about consequences of manual overrides.

3.4.1.8 HORSE-RIDER PARADIGM

The "horse-rider paradigm "is introduced in 1990 by Connell and Viola, and further it was developed by Flemish et al (2003), that named it "H-metaphor" and faced also by Norman (2007).

The “Horse-Rider paradigm” defines the relation between human and automation like the relation that a rider establishes with his/her horse, the human receives information about the actual system status through an osmotic exchange with it. Human intention (actions) become the parameters the system uses to offer as correct solution or answer to the faced context. Hence it is possible to improve the human performance that represents the crucial hearth of the interaction in complex systems. Besides the user is maintained in loop during the system control, in order to avoid or reduce the out-of-the-loop performance.

3.4.1.9 COMPUTERIZED ADAPTIVE TESTING

Lim Tock Keng, Ho Wah Kam Computerized Adaptive Testing in Reading Comprehension, A Computerized Adaptive Testing (CAT) project in reading comprehension was established to develop multiple choice tests grade levels, Primary are 3 and 5, and Secondary are 1 and 3. CAT is interactive and it allows participants to select their own entry points to the test and gives feedback on their performance. To Build a CAT system it requires the development of an item bank, selection of item and items order to be presented in a test, and evaluating the test for difficulty. The creation of the item deposits involved the reading comprehension skills, writing items, field testing, item analysis and calibration. The software, MICROCAT, used to develop an item deposit, to select items and item order to be presented in a test, and to evaluate the test for difficulty.

3.4.2 DYNAMIC (RUN TIME) STORAGE AND DATA (CENTER) VIRTUALIZATION

3.4.2.1 SECURING INTERNET PROTOCOL (IP) STORAGE: A CASE STUDY

Siva Rama Krishnan Somayaji, Ch.A.S Murty, published research paper Securing Internet Protocol (IP) Storage: A Case Study, Storage networking technology has enjoyed strong growth in recent years, but security concerns and threats facing networked data have grown equally fast. There are different ideas that are aimed at storage networks, including data modification, destruction and theft, DoS attacks, malware, hardware theft and unauthorized access, among others. Hence for a Storage Area Network (SAN) to be secure enough, each of these threats must be addressed inline. Conclusion of this research is a comparative study by implementing different security methods in IP Storage network and an IP-Storage network using iSCSI protocol. They analysed the performance of the IP Storage network without any security implemented and also by implementing SSLv2 and IPsec. And presented a comparative analysis IP storage network performance for every case.

3.4.2.2 A JOURNEY FROM FLOPPY DISK TO CLOUD STORAGE

Bindu Trikha (2010) mentioned Data storage and backup needs have evolved over the years necessitating the need for evolution of data storage methods and devices. The research was about the needs for a higher storage capacity and versatility of storage devices and the need for technologically advanced storage devices became apparent. She talked about all storage devices like punch cards, CD/DVD, Blu-ray Disks, and networked storage. Conclusion is that with the advent of time there is always a need for better options in terms of back up storage and as on date the best option available for internet users or Cloud Storage as backup store

3.4.2.3 WORKLOAD-AWARE VM SCHEDULING ON MULTICORE SYSTEMS

Insoon Jo, Im Y. Jung, Heon Y. Yeoma (2006) analysed that since in a virtual environment where multiple virtual machines can be run on a single physical host and found that performance interference between virtual machines is a big challenge in the field of virtualization. Hence I learned from this research work there were a workload aware virtual machine scheduler were introduced on multi core systems which finds the mapping of virtual machines related with the physical host and how VM can be scheduled to share the load with Dynamic Storage Allocation and Management .

3.4.2.4 PERSPECTIVE ON THE BENEFITS OF DATA VIRTUALIZATION TECHNOLOGY

Ramona A, Razvan R. (2006) done which work was based on the integration of virtualization technology with the data integration technology. He found that Data integration and maintenance is a big and costive way which includes the feeds in the applications, reporting, analysis etc. which required time and consume more resources. I been have some advantage got information from this research work were done to provide the benefits of combining the virtualization with data integration and how resources can be managed with run time dynamic storage allocation.

3.4.2.5 EFFECTIVE SECURITY ARCHITECTURE FOR VIRTUALIZED DATA CENTER NETWORKS

Udeze Chidiebele. C, Okafor Kennedy .C (2008) mentioned as Virtualization is the new and key concept in the field of information technology but since it's a new technology so there are lots of assumptions are exist with the security of the virtualized data center networks. He found from study that there were lots of architecture presented like integration of Open Flow Software Defined Networking (OFSDN) with VLAN Virtual Server Security (VVSS) were addressed to know the security issues in the virtualized data center. Hence other researcher benefitted from this research work about the security issues on the virtualized networks.

3.4.2.6 VIRTUAL DISK DRIVE SYSTEM AND METHOD

Soran; Philip E. (Eden Prairie, MN), Guider; John P. (North Oaks, MN) (2009) mentioned that disk drive is the main subsystem of a computer system and it may include RAIS subsystem which included the pool of storages like the Metrix of storage blocks or a page pool of storage. He concluded that Storage has a great importance in the field of virtualization and this research work was based on the concept of the virtualization of the storage drives. I got information about the Dynamic management of Storage Allocation with virtualization.

3.4.2.7 UNIFIED VIRTUAL STORAGE

Patil S.V and Honwadkar K.N (2009) Studies Virtualization of Distributed Storage in a Network. This work based on the few techniques to efficiently utilize the free disk space on the connected networked devices. Their founding is that since in the usual way we do reserved some amount of space to a machine and the unused space sits ideal there, in same way lots of storage space reserved and sits ideal on the several machines, Hence I was able to know about a new way were proposed to utilize such ideal space by using on shared basis on networked machines.

3.4.2.8 VIRTUALIZATION AS THE NEW AND KEY CONCEPT IN THE FIELD OF INFORMATION TECHNOLOGY

Hiteshi Atif (2010) mentioned about virtualization**,** since it's a new technology so there are lots of assumptions exist in regards to the security of the virtualized data center networks. He found that that there were lots of architecture presented like integration of Open Flow Software Defined Networking (OFSDN) with VLAN Virtual Server Security (VVSS) were addressed to address the security issues in the virtualized data center. Hence this research work was based on the security issues on the virtualized networks. Hence learned about the security implementation in Virtualized Dynamic Storage.

3.4.2.9 AVAILABILITY MODELLING AND ANALYSIS ON VIRTUALIZED CLUSTERING WITH REJUVENATION

Sung-Do Chi, Jong Sou Park (2006) mentioned as recovery and downtime is the major concern particularly in the field of information technology and lots of systems and terms are already defined to access the high availability and to get the free from error and fast failover either in term of application, machines or severs. It's found in this study that Clustering is provides the way to provide the high availability by running same redundant services in parallel so conclusion is that in case of failure of one service other will take over immediately.

3.4.2.10 DATA DYNAMICS USED FOR STORAGE SPACE IN CLOUD COMPUTING

K.GEETHA, DR. ANANTHI SHESHASAYEE published a research paper, they mentioned that Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. This work research the problem of ensuring the reliability of data storage in Cloud Computing. In particular, considering the task of allowing a Third party assessor, on behalf of the cloud client, to verify the reliability of the dynamic data stored in the cloud. The commencement of Third party assessor eliminates the involvement of the client through the assessing of whether his data stored in the cloud is indeed together, which can be helpful in achieving economies of scale for Cloud Computing. Further the support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, As services in Cloud Computing are not just limited to archive or backup data only. While other works done to ensure isolated data reliability often lacks the support of either public review capability or dynamic data operations, this work achieves both. Conclusion is to identify the difficulties and possible protection problems of extensions with fully dynamic data updates from prior works and then show how to construct the seamless combination of these two most important features in procedure design.

3.4.2.11 DYNAMIC STORAGE ASSURANCE ON CLOUD COMPUTING

P. Dhanalakshmi, V. Ramesh published paper about, Cloud computing investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. Also a distributed storage integrity auditing mechanism, it utilizes the homomorphic token and distributed erasure-coded data. In this research design allows users to audit the cloud storage with lightweight communication and computation cost. The auditing result ensures strong cloud storage correctness guarantee, and simultaneously achieves fast data error localization, i.e., the identification of misbehaving server and recover the corrupted data. As cloud data are dynamic in nature, the proposed design again supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. The conclusion of this paper is to prevent the file from integrity violations and recovering the corrupted file with easy overhead.

3.4.2.12 DATA CENTER VIRTUALIZATION AND ITS ECONOMIC IMPLICATIONS FOR THE COMPANIES

Logica BANICA, Mariana JURIAN, Cristian STEFAN studied In the current situation of the economic crisis, as and when companies target budget cuttings in a context of an explosive data growth, the IT community must evaluate potential technology developments not only on their technical advantages, but on their economic effects as well. More than ever, the old cliché "doing more things with fewer resources" is true today. They analysed Many IT companies started building very large facilities, called data centres (DCs) or Internet DC (IDCs), which provide businesses a wide range of solutions for systems deployment and operation. IT departments moved from data center and infrastructure consolidation to virtualization. Data center virtualization is the process of arranging available resources with the actual needs of the offered services, moving from physical servers to virtual servers, sharing and provisioning servers, networks, storage, and applications. Further By taking advantage of three basic innovations virtualization, tiered storage architectures and dynamic provisioning software, an organization can achieve greater efficiencies in their current computing environment. Unified computing architecture does end-to-end virtualization; all structures are optimized for virtualized environments, from the CPU to the aggregation layer. They found in conclusion that in combination with embedded management, this approach increases responsiveness and reduces the opportunities for human error, improving consistency and reducing server and network deployment times.

3.4.2.13 HIGH AVAILABILITY USING VIRTUALIZATION

Federico Calzolari (2009) presented a research paper with regards to Data Center where High availability has always been one of the main problems. He found that a new approach to the problem can be offered by virtualization. By using virtualization, now it is possible to achieve a redundancy system for all the services/processes running on a data center. This enhanced approach to high availability allows to share the running virtual machines over the servers up and running, by exploiting the features of the virtualization layer: start, stop and move virtual machines between physical hosts. The system (3RC) is based on a finite state machine with hysteresis, providing the possibility to restart each virtual machine over any physical host, or reinstall it from scratch. Further a complete infrastructure has been developed to install OS and middleware in a couple of minutes. To completely virtualize the main servers of a data center, a procedure has been developed to migrate physical to virtual hosts. The mentioned data center SNS-PISA is running at the moment in virtual environment under the high availability system. As an extension of the 3RC architecture, several storage solutions been tested to store and centralize all the virtual disks, from NAS to SAN, to grant data safety and access from everywhere. Conclusion was that exploiting virtualization and ability to automatically reinstall a host, they provide a sort of host on-demand, where the action on a virtual machine is performed only when a disaster occurs

3.4.2.14 MAKING I/O VIRTUALIZATION EASY WITH DEVICE FILES

Ardalan Amiri Sani, Sreekumar Nair, Lin Zhong, Quinn Jacobson (2013) presented a research paper, As Personal computers have diverse and fast-evolving I/O devices, making their I/O virtualization different from that of servers and data centres. In this paper, they present recent endeavours in simplifying I/O virtualization for personal computers. Their key insight is that many operating systems, including Unix-like ones, abstract I/O devices as device files. There is a tiny and stable set of operations on device files, Hence, I/O virtualization at the device file boundary requires a one-time effort to support various I/O devices. They further present de-virtualization, their design of I/O virtualization at the device file boundary and its implementation for Linux/x86 systems. Finally they were are able to virtualize various GPUs, input devices, cameras, and audio devices with fewer than 4900 LoC, of which only about 300 are specific to I/O device classes. Conclusion was that measurements show that de-virtualized devices achieve interactive performance from native ones by human users, even when running 3D HD games.

3.4.2.15 LARGE SCALE ONLINE STORAGE MANAGEMENT

Maurice Askinazi, David Free, Bruce Gibbard, Thomas Throwe (2003) presented a research paper on, The HENP computing facility at Brookhaven National Laboratory supports both the Relativistic Heavy Ion Collider (RHIC) and involvement of US in the ATLAS LHC experiment. This facility includes 150 Tera Bytes of centralized online (disk) storage, which is served to a processor farm of 2000 CPU's. Multiple levels of virtualization systems are used/managed in the deployment of this storage. They found that this allows for a great deal of flexibility in the maintenance, performance tuning, and expansion of the resource in a manner which is transparent to its users. The tools and strategies employed and the additional functionality achieved, they studied about consolidation, documentation and High Availability. Discussed about HBA, Switches and Cabling with Veritas Software for Storage Management. Showed the connectivity between these

3.4.2.16 AVAILABILITY ANALYSIS AND IMPROVEMENT OF SOFTWARE REJUVENATION USING VIRTUALIZATION

Thandar THEIN, Sung-Do CHI, Jong Sou PARK presents teaching and curriculum design for Information Technology classes. Present Days, students demand practical activities for the latest and upcoming technologies. It is now possible to satisfy this appetite for exciting education by employing server virtualization technologies to teach advanced concepts with extensive hands-on assignments. By utilization of virtualized servers, students are able to deploy, secure and manage virtual machines and networks in a container. Various techniques, assessment tools and experiences will be analysed and presented by this manuscript. Previous educating cases for Information Systems or Information Technology classes are done using non-commercial products, such as free VMware Server or VMware Player. These products may have very limited functionality in terms of networking, storage and resource management. Several advanced data center functions, such as Distributed Power

Management (DPM), vMotion and others, are not available in desktop versions of that type of virtualization software. This paper concludes the utilization of commercial software, such as vSphere 4.1, with full data center functionality and operations for teaching Information Technology classes of various levels.

3.4.2.17 SEMI SYMMETRIC METHOD OF SAN STORAGE VIRTUALIZATION

Dhanamma Jagli, Ramesh Solanki, Rohini Temkar, Laxmi Veshapogu presented a paper on Virtualization, which is one of the biggest buzzwords of the technology industry right at this moment. This research continues to search the fast growth in storage capacity and processing power in enterprise installations coupled with the need for high availability, requires Storage Area Network (SAN) architecture. This paper describing about Storage Virtualization, and Data Virtualization and also Symmetric and Asymmetric Virtualization at subsystem level. The goal of virtualization is to centralize administrative processes while improving scalability and workloads. This paper, concludes about new proposed method for virtualization, which would be overcome limitations of existed methods for storage virtualization and Data virtualization solves the most intractable troubles facing in IT organizations. Data virtualization processes agile and configurable layer between back-end physical databases and databases are represented using data services. Hence this proposed method would be providing all the feature of existed storage virtualization methods.

3.4.2.18 I/O DEMANDS OF BOTH SCIENTIFIC AND INDUSTRIAL APPLICATIONS

Traeger et al. 2008 Reported HPC in petascale computing is rapidly increasing I/O demands of both scientific and industrial applications. petascale computing must have the ability to process terabytes if not, petabytes of data which are generated in bursts and also should handle very high I/O concurrency from parallel processes running on millions of cores. Different layers of I/O stack, such as runtime library and OS kernel in order to explore higher parallelism and better locality of data access for I/O performance improvement. I/O stack are often unable to achieve the full potential, since aggregated impact is on the whole layers in turn that determines the I/O performance of the systems. Comprehensive re-examination of the design and implementation of existing software stack, especially for parallel I/O, is necessary for solving performance bottleneck in HPC system

The enterprise storage arrays architecture like Storage area Network (SAN), Direct attached storage (DAS), Network area storage (NAS) do even perform poorly when it comes to large scale distributed data intensive computing claimed by Philip Chen and Zhang 2014; Hennessy and Patterson 2011; Min et al. 2005. Authors also studied, in today's' storage system it has been noted that it significantly lacks in sustaining the strong growing concurrency and per compute throughput which is essential requirements of the I/O intensive applications executed in distributed environment.

3.4.3 AUTOMATED (ADAPTIVE CONTROL) SYSTEM TESTING

3.4.3.1 BEST PRACTICES FOR TESTING WITH EXISTING IT ENVIRONMENTS

Lazic Ljubomir has examined that Organizations are constantly working to leverage today's best practices for testing within the context of their existing IT environments. As IT works to balance the business needs for a certain application and the testing limitations with regards to resources and schedules, making the best use of the testing environment becomes critical. Doing optimized testing is a great way for organizations to move their testing efforts forward to reflect changing business environments and resource constraints.

3.4.3.2 TEST CASES TO MAXIMIZE THE PROPORTION OF PROGRAM

Palanisamy V has analysed that select test cases to maximize the proportion of program elements of a given type (e.g. Branches, statements, conditions, and loop) that are covered (and executed). The technique based on coverage makes use of greedy algorithm in order to prioritization out the repeatedly executing test cases. So, as per prioritization with the number of executing test cases the overall quality of the testing process can be improved.

3.4.3.3 STUDY AND ANALYSIS OF AUTOMATION TESTING TECHNIQUES

Sachin Sharma, Mrs. VISHAWJYOTI argued that Testing is a very important activity in Software Development Process. Hence to examine & modify source code. Effective Testing produces high quality software. This research Paper deals with a significant and vital issue of Software Testing. Testing can be done manually and by means of Automation as well. These Techniques have their own advantages & disadvantages. The Objective of this research paper is to execute Automation Testing using Software Testing Tool “Selenium”. With this testing tool, test cases are automatically recorded/saved in background while tester is entering the data in a web application screen. Hence I learned about the automation testing techniques and their advantage and disadvantage.

3.4.3.4 AUTOMATED TESTING IN DEVELOPMENT PHASE

SUNIL L. BANGARE, SACHIN M. KAMBLE, PALLAVI S. BANGARE, ABHIJIT V. NAIK done study In software development the applications are tested in testing phase of software development process. They found that testing of application is not possible without complete development of module/application. It takes extra time in completion of software development. Hence as proposed in this paper the model for tool which provides the way to developer to test his code/application in development phase itself. They also mentioned about the tool and the model helps in java API (application programmable interface) testing. With this tool, developer can able to test his code/module automatically considering all the aspect of testing. Here they have given an approach predefined test cases are loaded for testing, and thousands of test cases are run at same time and application is tested by developer. So it helps in regression testing. Hence I found the hints of in reducing software development period. Finally it saves the people resources, as well as hardware/software resources.

3.4.3.5 CHALLENGES FOR SOFTWARE ENGINEERING IN AUTOMATION

Birgit Vogel-Heuser, Christian Diedrich, Alexander Fay, Sabine Jeschke, Stefan Kowalewski, Martin Wollschlaeger, Peter G done study, which gives an introduction to the essential challenges of software engineering and requirements that software has to fulfil in the domain of automation. They concluded that besides, the functional characteristics, specific constraints and circumstances are considered for deriving requirements concerning usability, the technical process, the automation functions, used platform and the well-established models, which are described in detail. On the other side, challenges results from the circumstances at different points in the single phases of the life cycle of the automated system. The requirements for life-cycle-management, tools and the changeability during runtime are described in detail in this research.

3.4.3.6 AN INTEGRATED SELF-TESTING FRAMEWORK FOR AUTONOMIC COMPUTING SYSTEMS

Tariq M. King, Alain E. Ramirez, Rodolfo Cruz, Peter done study, As the technologies of autonomic computing become more prevalent, it is essential to develop methodologies for testing their dynamic self-management operations. Self-management features in autonomic systems induce structural and behavioural changes to the system during its execution, which need to be validated to avoid costly system failures. The next level of automation in systems also means that human errors such as incorrect goal specification could yield potentially disastrous effects on the components being managed; further emphasizing the need for runtime testing. In this paper a self-testing framework for autonomic computing systems is proposed to dynamically validate change requests. This framework extends the current architecture of autonomic systems to include self-testing as an implicit characteristic, regardless of the self-management features being implemented. They concluded and validate a framework by creating a prototype of an autonomic system that incorporates the ability to self-test.

3.4.3.7 ADAPTIVE AUTOMATION: LEVERAGING MACHINE LEARNING

Rajesh Mathur, Scott Miles, Miao Du done research to Support Uninterrupted Automated Testing of Software Applications, They started Checking software application suitability using automated software tools become an important element for most organisations irrespective of whether they produce in-house software applications or simply customise off-the-shelf software applications for internal use. They found that software solutions become ever more complex, the industry becomes increasingly dependent on software automation tools, yet the brittle nature of the available software automation tools limits their effectiveness. It's been discovered that Companies invest significantly in obtaining and implementing automation software but most of the tools fail to deliver when the cost of maintaining an effective automation test suite overrides the cost and time that would have other way been spent on manual software testing. Therefore A failing in the current generation of software automation tools is they do not adapt to unexpected modifications and obstructions without frequent (and time expensive) manual interference. Such problems are commonly acknowledged and known amongst industry practitioners, yet none of the current generation of tools have leveraged the advances in machine learning and artificial intelligence to address these problems. Thereby pre4ent paper proposes a framework solution that utilises machine learning concepts, namely fuzzy matching and error recovery. The suggested solution applies adaptive techniques to recover from unexpected obstructions that would otherwise have prevented the script from proceeding. Details are presented to the user of application in a report which can be analysed to determine if the recovery procedure was acceptable and the framework will adapt future runs based on the decisions of the user. Using concepts of this framework, a software testing practitioner can run the automated suits without human intervention while minimising the risk of schedule delays. Learning of adaptive methodology by fuzzy logic is been demonstrated in Software Automation Testing by tools.

3.4.3.8 TRAINING PEOPLE TO USE AUTOMATION: STRATEGIES AND METHODS

John Barnett presented a paper, as automation is being introduced into the workplace more and more frequently, and more and more people are learning to use automated systems. However, many people tend to exhibit patterns of behaviour towards automation which influences how they use it, or if they use it at all. Often, these behaviour patterns can either negate the advantages of automation, or allow automation to lead people into precarious situations. This paper concluded some of these common behaviour patterns and how training may help people avoid their negative consequences. Learning comes out as a suggested automation training strategy to help training developers design training programs for automated systems that takes user attitudes towards automation into account.

3.4.3.9 K MODEL FOR DESIGNING DATA DRIVEN TEST AUTOMATION

Rohan R. Kachewar presented that Frameworks and its Design Architecture Snow Leopard, here an automated testing improves the efficiency of testing practice on various sites of projects in the organization. Unfortunately, Its comes out that we do not have a common architecture or common standards for designing frameworks across different test levels, projects and test tools which can assist developers, testers and business analysts. To address the above problem, in this paper, He has first proposed a unique reference model and then a design architecture using the proposed model for designing any Data Driven Automation Frameworks. The conclusion is that the reference model is K model which can be used for modelling any data-driven automation framework.

3.4.3.10 AUTOMATION OF SMARTPHONE TRAFFIC GENERATION IN A VIRTUALIZED ENVIRONMENT

Tanya Jha, Rashmi Shetty presented a paper on Scalable and comprehensive analysis of rapidly evolving mobile device application traffic is highly important but a challenging problem for the Deep Packet Inspection, engines to perform effective policy management. A test framework in which a test driver can automate/orchestrate traffic generation is presented by invoking appropriate method (intent) of real mobile applications (as opposed to traffic replay) in regression testing of mobile application and traffic analysis engines in a virtualized environment, without real hardware. They concluded the concept by automating a real-time Skype call through a DPI engine in a virtual test setup using Android VMs. Understanding is made how automation can be made in network traffic with virtualized storage environment.

3.4.3.11 REGRESSION TESTING IN DEVELOPER ENVIRONMENT FOR ABSENCE OF CODE COVERAGE

M. Thillaikarasi, K. Seetharaman, presented a paper on the techniques of test case prioritization schedule the execution order of test cases to attain respective target, such as enhanced level of forecasting the fault. The prioritization be viewed as the path for deriving an order of relation on a given set of test cases which results from regression testing. Changing of programs between the versions can cause more test cases which may respond differently to following versions of software. In this process, a fixed approach to prioritizing test cases avoids the preceding drawbacks. The Unit test case prioritization techniques in the absence of coverage information, differs from existing dynamic coverage-based test case prioritization. They concluded paper that, the prioritization test cases relying on coverage information were projected from fixed structures relatively other than gathered instrumentation and execution.

3.4.3.12 SOFTWARE TEST AUTOMATION IN PRACTICE: EMPIRICAL OBSERVATIONS

Jussi Kasurinen, Ossi Taipale, Kari Smolander presented a paper on the objective of this industry study is to shed light on the current situation and improvement needs in software test automation. To this end, industry specialists from different organizational units were interviewed. In parallel with the survey, a qualitative study was conducted in selected software development organizations. The results showed that the software testing processes usually follow systematic methods to a large degree, and have only little immediate or critical requirements for resources. Hence the testing processes have approximately three fourths of the resources they need, and have access to a limited, but usually sufficient, group of testing tools. Hence the test automation, the situation is not as straightforward as it looks, based on our study, the applicability of test automation is still limited and its adaptation to testing contains practical difficulties in usability. In this study, we analyse and discuss these limitations and difficulties.

3.4.3.13 RELIABLE SOFTWARE DEVELOPMENT WITH PROPOSED QUALITY ORIENTED SOFTWARE TESTING METRICS

Latika Kharb, Dr. Vijay Singh Rathore researched about, an effective test measurement, a software tester requires a testing metrics that could measure the quality with productivity of software development process and increasing reusability, correctness and maintainability. The understanding of measuring software quality is not yet appropriate and is still far away from being standardized and in order to assess the software quality, an exact set of software metrics needs to be identified that any express these quality attributes. Our research objective in this paper is to construct and define a set of easy-to measure software metrics for testing to be used as early indicators of external measures of quality. So, it’s been emphasized on the fact that reliable software development with respect to quality could be well achieved by using set of testing metrics, and for that given the practical results of evaluation

3.4.3.14 WHEN TO RELEASE A SOFTWARE PRODUCT FROM THE PERSPECTIVE OF SOFTWARE RELIABILITY MODELS

Richard Lai, Mohit Garg, Parmod Kumar Kapur, Shaoying liu A Study of, If a software product with a significant number of defects is released too early to users, the software manufacturer will incur post-release costs of fixing the faults. If a product is released too late, the additional development cost and the risk of missing a market window could be substantial. Software Reliability Growth Models (SRGMs) can capture the quantitative aspects of testing and are used to estimate software release time. From a cost-benefit viewpoint, SRGMs aid developers to decide the optimal release time of the software product by providing effective approaches to minimising the expected total software system cost. This paper helps answer the question of when to stop testing a software product by presenting the perspectives from a study of cost models. The study focuses on aspects of the relationship between development cost and schedule delivery of the software product and the total software cost including the risk costs, such as the penalty cost incurred due to late delivery of software product and the cost of fixing a fault during the warranty period. We also investigate various software release policies, for example, policies based on the dual constraints of cost and reliability.

3.4.3.15 TOOLS AND BEHAVIOUR ABSTRACTION: A FUTURE FOR SOFTWARE ENGINEERING

Wilson Solís, Enrique Buenaponte, Marina Aguilar, Software engineers rely on and use tools to analyse automatically and detailed the code and design specifications. Some tools used to find new defects in old code, is expected in the future have more application in software engineering and are available to developers at the time of editing their products. If it were possible build tools fast enough and easy to use, software engineers would apply it to improve design and product development. But to solve any problem, traditional engineering use programming languages, however, the level of abstraction of the most popular is not much larger than C programs several decades ago. Moreover, this level is the same in all the code and do not leaves room for abstraction of behaviour, in which the design is divided into phases and which gradually introduces more details. This paper presents a study of the need for a larger set of analysis tools to create languages and development environments, which provide good support to archive this abstraction.

3.4.3.16 IMPORTANCE OF TESTING AND QA IN SDLC MODELS

Maneela Tuteja, Gaurav Dubey, A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle Models, In recent years, software testing is becoming more popular and important in the software development industry. Indeed, software testing is encircling a variety of activities along the development cycle and beyond, aimed at different goals. Hence, research in software testing faces a collection of challenges. A strict roadmap of most relevant challenges is proposed. The paths from the achievements to the goals are paved by outstanding research challenges, which are discussed in the paper along with the ongoing work. Software testing is old in the history of digital computers. Software testing is means of assessing the software to determine its quality. As testing typically consumes 40~50% of software development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering Software testing is a broad area, which involves other technical, non-technical areas, such as specification, design and implementation, maintenance, process and management issues in software engineering. This study focuses on the state of the art in testing techniques, as well as the latest techniques which representing the future direction of this area. Today, testing is the most challenging and dominating activity used by industry, therefore, improvement in its effectiveness, both with respect to the time and resources, taken as a major factor by many researchers. The purpose of software testing can be QA, verification/validation and reliability estimation.

3.4.3.17 OPTIMIZATION IN SOFTWARE TESTING USING METAHEURISTICS

FREITAS, F. G.,MAIA, C. L. B.,CAMPOS, G. A. L.,SOUZA, J. T.(2010), mentioned that there are Software Test problems that may not be solved with traditional software engineering techniques. Nevertheless, such problems may be modelled mathematically in order to be solved with mathematical optimization, especially with the use of metaheuristics. In this perspective, a new research field called Search based Software Engineering (SBSE), which deals with solving software engineering problems by means of optimization techniques, has emerged. Significance of the Software Testing phase, a specific technique called Search Based Software Testing (SBST) has become increasingly important. Initially, it’s described the main metaheuristics techniques used in the area. We follow with the presentation of the state of the art of SBST through the description of the main problems that have already been modelled and the results achieved. From the results, the promise of this field can be realized.

3.4.3.18 SOFTWARE AS A SERVICE (SAAS) TESTING CHALLENGES- AN IN-DEPTH ANALYSIS

Prakash. V, Ravikumar Ramadoss, S.Gopalakrishnan (IJCS, 2012), studied about Organizations in this modern era are interested in deploying and making use of readymade business applications. The reasons are , short time to market and lack of capital budget which is required to develop new software and for on-premise deployment and of course the rapid emergence of the Cloud. In fact cloud has attractive Software as a Service jargon which drives the idea of making use of ready-made and on-demand business solutions. As due to increasing demand in SaaS usage there is more in for SaaS Testing. This paper talks on the challenges for engineers in Saas and also analyses the ways in which SaaS testing differs from testing conventional applications

3.4.3.19 SOFTWARE DEVELOPMENT METHODOLOGIES, TRENDS AND IMPLICATIONS: A TESTING CENTRIC VIEW

Xihui Zhang, Tao Hu, Hua Dai, Xiang Li (ITJ, 2012), mentioned The practice of software development has evolved steadily over the decades. Various methodologies and models (e.g., life cycle models and agile methods) have been proposed to enhance its efficiency and effectiveness. This paper provides a testing centric view of software development processes. Particularly, it reviews software development methodologies (i.e., methods and models), identifies the latest trends in the industry and discusses their implications. The review of testing methodologies, the identification of trends and the discussion of implications will be useful to software development educators, students, practitioners and researchers.

3.4.3.20 TOWARDS TEST CASES GENERATION FROM SOFTWARE SPECIFICATIONS

R. Jeevarathinam, Dr. Antony Selvadoss Thanamani (IJEST, 2010), mentioned about Verification and Validation of software systems often consumes up to 70% of the development resources. Testing is one of the most frequently used Verification and Validation techniques for verifying systems. Many companies that certify software systems for use require that the software be tested to certain specified levels of coverage.

Presently, developing test cases to meet these requirements takes a big portion of the resources. Automating task result in significant time and cost savings. This software testing research is aimed at the generation of such test cases. In the proposed approach a formal model of the required software behaviour (a formal specification) is used for test-case generation and as an oracle to determine if the implementation produced the correct output during testing. This is referred to as Specification Testing. Specification based software testing offers several advantages to old code based testing. The formal specification is used as the source artifact to generate functional tests for the final product and since the test cases are produced at an earlier stage in the software development life cycle, they are available before the implementation is completed. In this approach the use of model checkers as test case generation engines is a central theme. Model checking is a process for exploring the reachable state-space of a system model to verify properties. There are some research challenges that must be addressed to realize this test generation approach. Conclusion is that this work is continuing instrumentation of Java byte code and will extend this work to C and C++. Some other research group has done fundamental research in other areas, such as software model checking (model checking the application itself and not just the input domain) and static analysis. In general, the ultimate goal is to combine the different technologies into a single coherent framework

3.4.3.21 CLASSIFICATION OF AUTOMATIC SOFTWARE BUILD METHODS

Marcin Kawalerowicz (Computer Science, 2013), the process of creating working software from source code and other components (like libraries, database files, etc.) is called "software build". Here apart from execution, linking and compiling, it can include other steps like automated testing, static code analysis, documentation generation, deployment and other. All steps can be automated using a build description (e.g. script). This research classifies the automated software build processes beginning at build script and reaching the various types of continuous integration.

This paper shows the classification of build automation dividing the automated build on the levered and continual builds. It also introduces continuous integration division to transitional (where build is not executed after every change and strict (where build is triggered after every change in the central repository). This paper described also methods of triggering the builds using polling and hooking.

3.4.3.22 CLOUD PENETRATION TESTING

Ralph LaBarge, Thomas McGuire (Computer Science, 2013), this paper presents the results of a series of penetration tests performed on the OpenStack Essex Cloud Management Software. Here Different types of security and penetration tests performed including network protocol and command line fuzzing, session hijacking and credential theft. Using mentioned techniques exploitable and vulnerabilities were discovered that could enable an attacker to gain access to restricted information contained on the OpenStack server, or to gain full administrative privileges on the server. So key recommendations to notify these vulnerabilities are to use a secure protocol, such as HTTPS, for communications between a cloud user and the OpenStack Horizon Dashboard, to encrypt all files that keeps user or administrative login credentials, and to correct a software bug found in the OpenStack Cinder type delete command. Conclusion is that it is important to continue to perform penetration tests on the OpenStack Cloud Management Software. OpenStack is getting used by many large companies for their private, and public clouds. Improving the overall security posture of OpenStack through penetration testing is a worthy effort since many OpenStack users are moving more of their applications and data into the cloud.

3.4.3.23 SOFTWARE TESTING MODELS AGAINST INFORMATION SECURITY REQUIREMENTS

Alexey Markov, mentioned about an overview and classification of software testing models are done. Recommendations on the choice of models are proposed, this research has revealed a great number of mathematical models that can be used to assess the technical software security at different levels of its lifecycle, which is very important for information security cost budgeting. The classification of models are practical when making the right choice or complexing models on the basis of available statistics. One should keep in mind that because of rapid development, complexity, and diversity of modern software kits, the above models must not be expected ever to provide high accuracy, and quite often they specifically provide intuitive data for taking a decision in preparation of software testing on the entire array of input data. The results of mentioned models applications are highly convenient for use in both the justification of testing labour costs and reporting records, which may be helpful for the customer to view the obtained results as reliable.

3.4.3.24 FORMAL METHODS OF SOFTWARE TESTING AND TERMINOLOGY

Sunil Kumar Scholar, Dr. P.K Yadav, (IJTA, 2011), Software provides a complete set of application development tools for building stand-alone, client-server, and Internet-enabled applications. But building it easy to build applications can be a two-edged sword. Not only developers can build powerful, and sophisticated applications, but also they can also build applications that troubled users, waste computer resources, and damage the credibility of the developer. Formal testing helps prevent bad applications from being released. For those unfamiliar with the topic, this paper can serve as a primer or first step in learning about a more formal, rigorous approach to software testing, conclusion is that Software testing is a critical element in the software development life cycle and has the potential to save time and money by identifying problems early and to improve customer satisfaction is by delivering a product without any defect. Unfortunately, it is often less formal and rigorous than it should, and a primary reason for that is because the project staff is unfamiliar with software testing methodologies, approaches, and tools. Without proper testing, however, there is a greater risk that an application will inadequately deliver what was expected by the business users or that the final product will have problems such that engineers will eventually abandon it out of frustration.

3.4.3.25 HOW AUTOMATED TESTING TOOLS ARE SHOWING ITS IMPACT IN THE FIELD OF SOFTWARE TESTING

Deepti Gaur, Dr. Rajender Singh Chhillar (IJCSMS, 2012), as, we know that Software testing is a very vast field in Software development life cycle. This paper, it's described that how automated testing tools are very much convenient and easy to use which also makes testing faster and more effective in less time. Actually the eco system of technology revolves at fast pace today and among all Testing tools, automated testing tools makes Software testing more significant and effective.

3.4.3.26 A SURVEY ON SOFTWARE TESTING TECHNIQUES IN CLOUD COMPUTING

Priyadharshini. V, Malathi (Computer Science, 2014) mentioned about Cloud computing is the next stage of the internet evolution. It works on sharing of resources to achieve coherence on a network. It enhanced computing standard that impacts several research fields, including software testing. There are different software techniques used for testing application. It changes the way of obtaining computing resources and also changes the way of managing and delivering computing services, technologies and solutions, meanwhile it causes new issues, challenges and needs in software testing. Software testing with cloud can reduce the need for hardware/software resources and offer a flexible and efficient alternative to the traditional software testing process. This paper provides an overview regarding trends, opportunities, challenges, issues, and needs in cloud testing and cloud based application. Conclusion is that Functional testing acquires high usage of hardware and software to simulate user activity. And non-functional testing clear the way of the measurement and association of the testing of non-functional attributes. Only some advantages and testing challenges of cloud computing have been identified. Hence Testing is a cyclic activity and new requirements need to be set up for each project.

3.4.3.27 A BRIEF OVERVIEW OF SOFTWARE TESTING METRICS

Premal B. Nirpal, Dr. K. V. Kale (IJCSE, 2011), Metrics are gaining importance and acceptance in corporate sectors as organizations grow, mature and strive to improve enterprise qualities. Measurement of a software test process is a required condition for an effective software test manager for designing and evaluating a cost effective test strategy. Effective management of software testing process requires quantification, measurement and modelling. Software Metrics gives quantitative approach to the validation of the software process models. This also help organization to obtain the information it needs to continue to improve its productivity, reduce errors and improve acceptance of processes, products and services and achieve the desired Goal. This research paper, focusing on metrics lifecycle, various software testing metrics, need for having metrics, evaluation process and arriving at ideal conclusion have also been discussed in the present paper. Conclusion is that Metric is the cornerstone in assessment and foundation for any business improvement. Therefore a Measurement Based Technique which is applied to processes, products and services to supply engineering and management information and working on the information supplied to improve processes, products and services, if required. It indicates Customer satisfaction level, easy for executive management to digest number and drill down, and whenever required and act as monitor when the process is getting out-of-control.

3.4.3.28 SOFTWARE TESTING AND SOFTWARE DEVELOPMENT LIFECYCLES

Chitra Wasnik (IJCDS, 2013), Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. What is Software Testing? This is a process of validating and verifying about a software program expected to do. Software Testing is a thorough investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing provides an objective, independent view of the software to allow the business to understand the risks at implementation of the software eco system. Test techniques includes, the process of executing a program or application with the intent of finding software bugs. It can also be said as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as required and can be implemented with the same characteristics. Software Development Life Cycle (SDLC) is a methodology that is typically used to develop, maintain and replace information systems for improving the quality of the software design and development process.

3.4.4 CLOUD (DYNAMIC) AND STORAGE NETWORK (NETWORKED STORAGE)

3.4.4.1 DISTRIBUTED STORAGE CLUSTER DESIGN FOR REMOTE MIRRORING BASED ON STORAGE AREA NETWORK

Un Yao, Ji-Wu Shu, and Wei-Min Zheng, mentioned that with the explosion of information nowadays, applying data storage safety requirements has become a new challenge, particularly in high data available cluster environments. With the emergence, storage can be consolidated, network-based, and mass data movements via Fiber Channels (FCs) can be of very high speed. Based on mentioned features, this research paper introduces a dual-node storage cluster designed for remote mirroring as a concurrent data replication method to protect data during system failures. This design takes advantage of a SAN system's benefits, and it process through a synchronous protocol to guarantee a fully up-to-date data copy on the remote site. By developing a Linux Operating System kernel module to control the I/O flow and by using the technologies of software Logic Unit Number (LUN) masking, background online resynchronization and a self-management daemon, it’s achieved a reliable mirroring system with the characteristics of server-free data replication, fault tolerance, online disaster recovery and high performance. In this research study, they implemented the design in a remote mirror subsystem built on a software Fiber Channel Storage Area Network (FC-SAN) system.

3.4.4.2 AN APPROACH FOR INVESTIGATING PERSPECTIVE OF CLOUD SOFTWARE-AS-A-SERVICE (SAAS)

Bhardwaj Sushil, Jain Leena, Jain Sandeep (2009) mentioned As Cloud computing further extending the area of virtualization, SOA, Web center, Information technology management etc. and bringing new paths to extend dimension on utility computing. They found in their research that Cloud computing provide an web interface for your application so you don't have to worried about the Hardwar's and software's and you have to only use services as a service. With virtualization it combines lots of facilities and enhancements and could computing can be categories further as SaaS, IaaS, Paas and SaaS.

3.4.4.3 CONCEPT OF CLOUD SERVICE MODEL

Shamsolmoali Pourya, Alam M.Afshar (2010) done research work to introduced the concept of cloud service model overview and discussed on the way to design a platform for a virtual group of engineers to explain and display their achievements on the network. They Found that Cloud computing has been defined and designed by the independent third party body and now after the development of cloud computing and internet become a new network manufacturing mode.

3.4.4.4 TECHNIQUES TO EFFICIENTLY UTILIZE THE FREE DISK SPACE ON THE CONNECTED NETWORKED MACHINES

Inyiama H.C, Okezie C.C (2010) mentioned since in the usual way we do reserved some amount of space to a machine and the unused space sits ideal there, in same way lots of storage space reserved and sits ideal on the several machines, so here a new way were proposed to utilize such ideal space by using on shared basis on networked machines. It helped to understand the using networked storage in sufficient manner.

3.4.4.5 KEY MANAGEMENT FOR ENCRYPTED STORAGE IN STORAGE AREA NETWORK

Hai Xin LU checked that secure storage becomes more pervasive throughout the enterprise, the focus quickly moves from implementing encrypting storage devices to establishing effective and secure key management policies. He found that although a considerable amount of research has been dedicated to encryption algorithms in the past decades, key management becomes an issue due to the quantity of data. i.e. with millions of data needed with million set of keys. To manage of these keys, complexity and operational inefficiency becomes an issue in Storage Network System. Hence from this research paper it's been understood about the new challenges essential to effectively devise new key management policies and mechanisms for secure storage.

3.4.4.6 CLOUD COMPUTING: AN INTERNET BASED COMPUTING

Hardeep Singh done study on the overview of Cloud Computing. Further he analysed that Cloud Computing is an Internet-based computing; where resources, software and information are provided to computers on-demand, like a public utility; is emerging as a platform for managing resources i.e. infrastructure, software and various applications. And concluded on some of the prominent applications of Cloud Computing, and how they meet the requirements of reliability, availability of data, scalability of software and hardware systems and overall customer satisfaction.

3.4.4.7 CLOUD COMPUTING-SOFTWARE AS SERVICE

Gurudatt Kulkarni presented a research paper on, he does research on SaaS service model, He further stated that Cloud Computing, means "Internet Computing." The whole Internet is commonly seen as networks of clouds; hence the term "cloud computing" for computation done through the Internet. In Cloud Computing eco system users can access database resources via the Internet, as long as they need, without worrying about any maintenance or management of actual resources. Databases in cloud are very dynamic and scalable. The paper concluded that Cloud computing is unlike grid computing, utility computing, or autonomic computing. In particular, it is a independent platform of computing. The best Example of cloud computing is Google Apps where any application can be accessed using a browser and it can be deployed on thousands of computer through the Internet.

3.4.4.8 OVERVIEW OF SECURITY ISSUES IN CLOUD COMPUTING

Mr. Ajey Singh, Dr. Maneesh Shrivastava started with definition of cloud computing as management and provision of resources, software, applications, and information as services with the cloud (internet) on demand. They further studied that Cloud computing comes into focus only when you think about what IT always needs a way to increase capacity or add capabilities on the fly without investing in added infrastructure, training new personnel, or licensing new software. "Cloud computing continues to gain more acceptance as a critical way to deliver on-demand e-sources to customers," The cloud architecture is implemented in such a way that it gives the flexibility to share application as well as other network resources(hardware etc.). This leads to a need based flexible architecture where the resources are getting expand or contract with a little configuration changes. Cloud computing is sometimes provided "as a service" over the Internet, typically in the form of infrastructure as a service (IaaS), platform as a service (PaaS), or software as a service (SaaS). An end-users perspective, they don't need to care for the OS, the plug-ins, web security or the software platform. Everything should be in place without any worry. This paper concluded on technical security issues in cloud computing, cloud computing has various benefits in an enterprise but major concern is how security is implemented in cloud computing

3.4.4.9 DATA-PLACEMENT STRATEGY BASED ON GENETIC ALGORITHM IN CLOUD COMPUTING

Qiang Xu, Zhengquan Xu, Tao Wang presented started with the development of Computerized Business Application, the amount of data is increasing exponentially. Cloud computing gives high performance resources and mass storage resources for massive data processing. In distributed cloud computing systems, data intensive computing can lead to data scheduling between data centres. Reasonable data placement can reduce data scheduling between the data centres effectively, and improve the data acquisition efficiency of users. They proceed in this paper, about the mathematical model of data scheduling between data centres is built. Now Global optimization ability of the genetic algorithm, generational evolution produces better approximate solution, and gets the best approximation of the data placement at last. The experimental results concluded that genetic algorithm can effectively work out the approximate optimal data placement, and minimize data scheduling between data centres.

3.4.4.10 SECURE DATA STORAGE IN CLOUD COMPUTING

B. Shwetha Bindu, B. Yadaiah started researching about Cloud computing which has gained a lot of hype in the current world of I.T. Cloud computing is termed as the next big thing in the computer world after the internet. Cloud computing is utilizing internet for the tasks performed on the computer and it is visualized as the next- generation architecture of IT Enterprise. The ‘Cloud’ represents the internet. Further they found that Cloud computing is related to several technologies and the convergence of various technologies has emerged to be called cloud computing. In comparison to conventional ways Cloud Computing moves application software and databases to the large data centres, where the data and services will not be fully trustworthy. And they concluded on secure data storage in cloud; it is an important aspect of Quality of Service. To ensure the correctness of users’ data in the cloud, an effectual and adaptable scheme is proposed with salient qualities. This research achieves the data storage correctness, and allow the authenticated user to access the data and data error localization, i.e., the identification of misbehaving servers.

3.4.4.11 AN OVERVIEW OF CLOUD TESTING AS A SERVICE

Amandeep Kaur Parmar, Navjeet Singh, Dr. Gurdev Singh presented a research paper on how Cloud storage is used , They started with Testing which is an important process for software quality assurance. Here it’s been noticed that a cloud infrastructure creates significant new opportunities for software quality assurance and testing. Making the Software test infrastructure that directly connects the production cloud infrastructure is too expensive, in terms of hardware, software licenses and software professionals. It’s been concluded that to deal with this challenge the paper gives an overview of various cloud testing strategies and also introduces ‘Testing as a Service’, using Cloud computing.

3.4.4.12 CLOUD VERSUS ON-PREMISE COMPUTING

Cameron Fisher published a research paper on, this study talks about investigating new choices for enterprise solutions, decision-makers need to increasingly weigh the merits of Cloud offerings. System performance and security are key requirements along with vendor reputation and user community. The competition to acquire customers and expand market share is prompting vendors to offer attractive introductory pricing to capture Cloud tenants. Further knowing the business, technology and contractual drivers will inform the decisions on the future of Cloud at your organization. Hence decisions to embrace the Cloud will always require complete analysis of the options and business metrics. So after performing a full needs analysis and understanding the variables, a reliable and cost-effective result is better selected and managed. The conclusion is to understand the implications for deciding on Cloud versus On-Premise Computing going forward.

3.4.4.13 VERIFICATION OF DATA RELIABILITY AND SECURE SERVICE FOR DYNAMIC DATA IN CLOUD STORAGE

Nithiavathy.R, Suresh J, presented paper on, Cloud computing has been the genuine solution to the rising storage costs of IT Enterprises. The prise of data storage devices is too high rate at which data is being generated, where the enterprises or individual users to frequently update their hardware or software. And the data outsourced to the cloud would help in reducing the maintenance. The user's data are moved from cloud to large data centres, which are located remotely which does not have control over it. The design allows the user with communication and computation cost. To maintain reliable cloud storage correctness, and to locate them is behaving server in which the data are frequently changing in cloud. It is an efficient method for dynamic operation which include erase, append, and block modification and it very effective in fighting against server colluding attacks, by zantine failure, malicious data block modifications.

The enterprise storage arrays architecture like Storage area Network (SAN), Direct attached storage (DAS), Network area storage

(NAS) do even perform poorly when it comes to large scale distributed data intensive computing claimed by **Philip Chen and Zhang 2014**; Hennessy and Patterson 2011; Min et al. 2005. Authors also studied, in today's' storage system it has been noted that it significantly lacks in sustaining the strong growing concurrency and per compute throughput which is essential requirements of the I/O intensive applications executed in distributed environment.

3.4.4.14 STORAGE AREA NETWORK PROBLEM-SOLVING ISSUES

Priyanka Malviya studied about some issues of SAN Infrastructure, How to access SAN, advantage and disadvantage, connection issues, HBA configuration issues, SAN Boot issues, SAN connectivity issues, and then solving those issue.

However, there are still issues that can occur and take some time to resolve that problem come in the Storage area network applications. The number of Storage Protocols and Storage Interfaces rapidly increased in a Networking technology field, it avoids the Bottleneck of data centres. This research paper focuses on some guidelines that may help to understand some of the design issues involved in SAN. Problems that are abstract and cannot solve on SAN infrastructure and then application run on SAN can solve after understanding all the parameters. Fibre channels also, make some concern that is not solvable and creates issues.

3.4.4.15 SURVEY ON CLOUD STORAGE

Jiehui JU, Jiyi WU, Jianqing FU, Zhijie LIN (2011) published a research paper, They mentioned that As interest in the cloud increases, there has been a lot of discussions about the maturity and trustworthiness of cloud storage technologies. They done a study and survey about, Is it still hype or is it real? Many users and managers are getting very excited about the potential benefits of cloud storage, such as being able to store and manipulate data in the cloud and capitalizing on the promise of performance, more scalable, and cheaper storage. They concluded research paper with a typical Cloud Storage system architecture, a reference Cloud Storage model and Multi-Tenancy Cloud Storage model, survey the past and the state-of-the-art of Cloud Storage, and the

Advantage and challenges that must be addressed to implement Cloud Storage Network. In this research use cases in Cloud Storage offerings were also summarized.

3.4.4.16 SCALING DATA AND IO OPERATIONS

Tran et al. 2012 proposed Applications are being deployed to read more data thereby increasing the I/O operation. Clients make simultaneous accesses to trivial portions of gigantic multidimensional storage array. The clients vary its access operations with different patterns of read and write. This diversity limits the scalability of storage and data management, which becomes the critical issue. In order to encounter the needs of applications, the storage stack needs to be enhanced and specialized. In Exascale scenario, the chunking layout management becomes the bottleneck that is not addressed by this proposed model. There is a huge gap being created between the relational tables, file system and application model.

3.4.4.17 DYNAMIC ACTIVE STORAGE

Chen and Chen 2012 came up with dynamic active storage. The research work discussed the essential idea of active storage – moving code near data for execution. The proposed model is more suitable for those kinds of applications where the data does not have dependency among various data nodes or the applications should share the successive data dependence. The proposed model gathers the data dependence pattern and file distributing information. Based on the information the active storage client will calculate and predict the bandwidth consumption to execute a task. If the predicted bandwidth cost is less than the file size, the offloading will not happen else the request is accepted. This research has contributed by proposing the DAS system and highlighted the data dependency will degrade the performance of I/O intensive application executed in the distributed and parallel networked environments.

3.4.4.18 ACTIVE STORAGE FABRICS CONCEPT

Authors Fitch et al. 2009 reviewed and researched on active storage fabrics concept. The concept exhibits the computations being embedded with distributed data facilitating the execution of applications closer to the data in a parallel or serial fashion. The execution takes place by using the common data access methods and to alleviate the interoperations and executions. A slight modification needs to be made to the application-storage interface and the middleware to consider both the execution fashion. The authors have merged the fabric with middleware components using IBM GPFS, IB 2 and created prototypes with Blue Gene/L/P systems. The literature review reveals that the proposed framework is hugely dependent on the Memory (DRAM) of the parallel machine. Putting together the parallel in database memory components will form the active storage fabrics. It uses the Key/Value pair and is distributed along with partitioned data sets on different servers or parallel servers. The overhead here is maintaining the balance of the datasets, which are distributed. The Authors research has revealed directions to develop and modify utilities for data management with POSIX as well as RDBMS and how the active storage utilities and legacy components can inter work with each other.

3.4.4.19 PERFORMANCE AND AVAILABILITY OF THE STORAGE SYSTEM

Sivathanu and Bairavasundaram n.d. Proposed performance and availability of the storage system can be improved by expending the semantically smart disks. These disks are well aware of the file system structures and execute the file system operations. Explicitly, the research work enhanced the probability of using such disks for deploying database systems. In order to achieve the goal, authors have informed the required changes for configuration of database and the changes to be made in file system. This is vital to tap the potential of disks and explore its competencies to make database work on it. The limitations that are observed from this proposed model are lack of communication pattern. There is not enough intellectual regarding the pattern in which the applications communicate with the disks or vice-versa. Even though is some interfaces like SCI or IDE is used for communication purpose, in order to understand the semantics at both the ends there would be need of a third interface which is also not mentioned in the research study.

3.4.4.20 DISTRIBUTED STORAGE CLUSTER DESIGN FOR REMOTE MIRRORING BASED ON STORAGE AREA NETWORK

Jun Yao, Ji-Wu Shu, and Wei-Min Zheng presented a research paper, They proceed with the explosion of information nowadays, and applying data storage safety requirements has become a new challenge, especially in high data available cluster environments. With the enhancement of Storage Area Networks (SANs), storage can be network-based and consolidated, and mass data movements via Fiber Channels (FCs) can be of very high speed. Based on features, this research paper introduces a dual-node storage cluster designed for remote mirroring as a concurrent data replication method to protect data during system failures. This storage systems design takes full advantage of a SAN system's benefits, and it adopts a synchronous protocol to guarantee a fully up-to-date data copy on the remote site. In this study, they implemented the design in a remote mirror subsystem built on a software Fiber Channel Storage Area Network (FC-SAN) system.

3.5 RESEARCH GAPS IN THE PROPOSED FIELD OF INVESTIGATION

Researcher has seen that As Huey-Min Wu, Bor-Chen Kuo and Jinn-Min Yang submitted research paper on “Evaluating Knowledge Structure-based Adaptive Testing Algorithms”. In this automated adaptive tool have interface and as per the login to user that tool was easily distribute questionnaires to take the exams and even was able to display results, but the scope was limited to evaluation and it was not to adapt the nature of run time environment.

Morrison & Gluckman, 1994; Rouse & Rouse, 1983 discussed about different strategies, mainly interaction between user and machine, where machine could estimate different states of an user’s activities, intentions, resources, and performance. Therefore same Information about the user, the system, and the outside world could then be interpreted with respect to the goals and actions to determine the need for adaptive aiding, hence machine learning is proposed with regard to user activities not as per dynamic environment. Kaber et al, 2001, told that the dynamic function allocation (DFA) is a peculiar aspect of AA. It basically consists of assigning the authority on specific functions to either the human user or the automated system.

The York Method (developed at the Department of Computer Science, University of York) tell about the design consideration and desire of control to automate.

The “Horse-Rider paradigm” tells the relation between human (user) and automation like the relation that a rider establishes with his/her horse. As per all above findings , they were able to success in their design criteria but no algorithmic solution was provided to select criteria for automated tools as per the adaptive nature of environment and no study was done to design algorithm adaptive as environment changes and their implementation criteria in run time dynamic environment.

The researcher observed that there in all previous studies, no implementations and design done towards applications verifying networking cloud/environment and component under test (network node, NIC, HBA, etc...) With respect to the run time characteristics obtained thorough feedback mechanism.

k.geetha, dr. ananthi sheshasayee published a research paper data dynamics used for storage space in cloud computing, but lacking adaptive approach for using storage space.

Patil S.V and Honwadkar K.N (2009) done research on Unified Virtual Storage but this Unified virtual storage lacking the approach of dynamic management with adaptive automation.

Maurice Askinazi, David Free, Bruce Gibbard, Thomas Throwe (2003) presented a research paper on Large Scale Online Storage Management, but this also lacking the approach of adaptive automation in online storage management.

Dhanamma Jagli, Ramesh Solanki, Rohini Temkar, Laxmi Veshapogu presented a paper on Semi Symmetric Method of SAN Storage Virtualization, this have dynamic management but Semi Symmetric Method does not have adaptive approach of managing it.

Latika Kharb, Dr. Vijay Singh Rathore researched about Reliable Software Development with Proposed Quality Oriented Software Testing Metrics, but adaptive automated testing is missing in Software Testing Metrics. Quality of software would have been better if adaptive algorithms have and tools supported are being used.

Hiteshi Atif (2010) done research to found Virtualization as the new and key concept in the field of information technology, virtualization supports dynamic automation for supporting different applications but not fully acquired with adaptive automation.

It is clear there are different approaches to develop adaptive algorithms, especially in the area of avionics, Neuro-ergonomics, Educations, robotics and missile/space and communications technology but nobody have considered enough to implementation in testing of high performance storage networking and control systems and even no appropriate automated tool being developed on the basis of adaptive philosophical theory to work and test the technological and networking systems and high performance control systems. System for Automated Build, Analysis and Automated Testing also need same.

CHAPTER 4

INTRODUCTION
TO
AUTOMATON

CHAPTER 4

INTRODUCTION TO AUTOMATON

4.1 INTRODUCING THE ADAPTIVE (AUTOMATED) LIFE CYCLE MANAGEMENT (ALM)

Today's applications face the challenge of doing required tasks and perform within an ever-shrinking schedule and with minimal resources. Hence Project stake holders attempt to do more with less, because organizations want to test software as quickly and thoroughly as possible. Hence organizations are turning to automated testing for Applications, Systems and Other Hardware resources.

This is reality that many (most of the) tests cannot be executed manually all the time, for example simulating 1, 0000 virtual users for volume testing, automated testing is introduced for such projects. But stakeholders may not know what's involved in introducing an automated tool to a software project, and also unfamiliar with the breadth of application that automated tools have.

But by defining systematic approach within the ALM, test activities can be organized and executed in such a way as to maximize test coverage within the limits of testing resources. Structured test methodology involves a multi-stage process, supporting the detailed and interconnected activities that are required to introduce and utilize an automated test tools and Techniques:

1) Develop test design and test cases.
2) Develop and execute test cases.
3) Develop, manage and arrange test data and the test environment.
4) Document, track, and obtain closure on issue/trouble reports.

Automated testing represents a paradigm change not only in the software industry but also in manufacturing Industry. This change doesn't simply involve the application of test automation tools. Rather, it have the entire test lifecycle and the system development lifecycle. The ALM implementation takes place in parallel with the system (software or any product) development lifecycle. Software professionals (engineers) to make a successful leap to automated adaptive testing, they must embrace structured approaches to testing. The ALM is revolutionary as it promulgates a new structured, building-block approach to the entire test lifecycle, which enables software professionals and stakeholders to approach software testing and Automation in a methodical and repeatable fashion.

The growth of adaptive capability has stemmed in large part from the growing popularity of the iterative and incremental development lifecycle, a software development methodology that focuses on minimizing the development schedule while providing frequent, incremental software builds. The objective of incremental and iterative development is to engage the user early throughout the design and development of each build in order to refine the software, and ensuring that it more closely reflects the needs and preferences of the user and thus addressing the required aspects of development in early builds.

In the environment of continuous changes and updates to the software through each software build, software testing itself takes on an iterative nature. Next build is accompanied by a considerable number of additional tests as well as rework to existing test scripts, or in software modules. Given the continual changes and additions to software applications, especially Web applications, adaptive software testing becomes an important control mechanism to ensure accuracy and stability of the software through each build of application.

The ALM, invoked to support automation efforts involving automated test tools, incorporates a multi-stage process. This supports the detailed and interrelated activities that are required to decide about Tools and Methods required for project. The methodology includes the process of how to utilize an automated test tool, covers test development and test design, and addresses test execution and management. It also supports the development and management of test data and the test environment, and addresses test documentation to include problem reports.

It represents a structured approach to execute automated testing. It is necessary to help the team away from these common test (White Box, Black Box) program mistakes:

1) Starts the use of an automated test tool without a testing process in place, resulting in an ad hoc, non-repeatable, non-measurable test program goes away from adaptive approach

2) Starts Implementation of a test design without following any design standards, resulting in the creation of test scripts that are not repeatable and therefore not reusable for incremental software builds.

3) Doing Attempt to automate 100% of testing requirements, when tools or in-house–developed automated test harnesses do not support automation of all tests required.

4) Using the wrong tool or developing elaborated in-house test harness.

5) Including test tool implementation too late in the Application-Development Lifecycle, therefore not allowing sufficient time for tool setup and test tool introduction process (learning curve) which should be of Adaptive nature.

6) Initiating Automation Test engineer involvement too late in the application-development lifecycle, resulting in poor understanding of the application and system design, which results in incomplete automation testing.

The Adaptive (Automated) Lifecycle Methodology (ALM) comprises following components:

1. Take Decision to Automate in Quality Assurance, Quality Engineering and Build & Release.
2. Getting requited Tools Acquisition for Analysis.
3. Doing Automated Introduction Process.
4. Working with Planning, Design, and Development.
5. Defining and Execution and Management of Process.
6. Finalize Program Review and Assessment.

Following sections describe process, and subordinate processes contained within each primary process

4.2 DECISION TO AUTOMATION

The decision to automation represents the first phase of the ALM. This phase covers the entire process that goes into the automation decision. During this phase, it's important for the team to manage automation expectations and to outline the potential benefits of automation when implemented correctly. A tool proposal needs to be outlined, which will be helpful in acquiring management support.

4.2.1 OVERCOMING FALSE EXPECTATIONS FOR AUTOMATED TESTING

While it has been proven that automation is valuable and can produce a successful return on investment, there isn't always an immediate payback on investment. It's important to address some of the misconceptions that persist in the industry and to manage the automation utopia. There is a misconceptions.

Automation needs a significant short-term investment of time and energy to achieve a long-term return on investment (ROI) of faster and cheaper regression.

4.2.2 AUTOMATIC TEST PLAN GENERATION

Currently, there is no commercially available tool that can automatically create a comprehensive plan while also supporting design and execution.

Throughout a software career, the engineer can expect to witness tool demonstrations and review an abundant amount of tool literature. Often the engineer will be asked to stand before one or more senior managers to give a tool functionality overview. In this case, the audience may represent individuals with just enough technical knowledge to make them enthusiastic about automation, while unaware of the complexity involved with an automation effort. Specifically, the managers may have obtained secondhand information about automated tools, and may have reached the wrong interpretation of the actual capability of automated tools.

Management may expect that the tool being proposed automatically develops the plan, designs and creates the procedures, executes all the procedures, and analyzes the results automatically. Meanwhile, informing the group that automated tools should be viewed as *enhancements to manual process*, and that automated tools will not develop the plan, design and create the procedures, or execute the procedures.

Shortly into the presentation and after several management questions, it becomes very apparent just how much of a divide exists between the reality of the tool capabilities and the perceptions of the individuals in the audience. The *automated tool* seems to bring with it a great deal of wishful thinking that's not closely aligned with reality.

An automated tool will not replace the human factor necessary for testing a product. The proficiencies of engineers and other QA experts will still be needed to keep the test machinery running. An automation tool can be viewed as an additional part of the machinery that supports the release of a good product.

4.2.3 ONE TEST TOOL FITS ALL

A single tool cannot be used to support all operating system environments.

Generally, a single tool will not fulfill all the requirements for an organization/team. Consider the experience of one engineer encountering such a situation. The engineer is asked by a manager to find a tool that could be used to automate the all the department's applications. The department using various technologies including mainframe computers and Sun workstations; operating systems such as; programming languages such as C,C++, Java, Python other client/server technologies; and Web technologies such as DHTML, XML, ASP, and so on.

After conducting a tool evaluation, the engineer determined that the tool of choice was not compatible with the Visual C++ third-party add-ons (in this case, Stingray grids). Other tool to be brought in that was compatible with this specific application.

4.2.4 IMMEDIATE REDUCTION IN SCHEDULE

An automated tool will always not minimize the testing schedule.

Another automation misconception is the expectation that the use of an automated tool on a new project will immediately minimize the schedule. The schedule will not experience the anticipated decrease at first, and an allowance for schedule increase is required when initially introducing an automated tool. This is due to the fact that when rolling out an automated tool, the current process has to be augmented or an entirely new process has to be developed and implemented.

The entire team and possibly the development team, needs to become familiar with this new automated process (such as ALM) and needs to follow it. Once an automatic process has been established and effectively implemented, the project can expect to experience gains in productivity and turnaround time that have a positive effect on schedule and cost.

4.2.5 BENEFITS OF AUTOMATED TESTING, QA, AND CONTINUES INTEGRATION/DELIVERY.

The previous discussion points out and clarifies some of the false automated expectations that exist. The engineer will also need to be able to elaborate on the true benefits of automation, when automated testing is implemented correctly and a process is followed. The engineer must evaluate whether potential benefits fit required improvement criteria and whether the pursuit of automation on the project is still a logical fit, given the organizational needs. There are three significant automation benefits:

- Producing a reliable system.
- Improving the quality of the effort.
- Reducing effort and minimizing schedule.

Many return on investment case studies have been done with regard to the implementation of automation.

4.2.6 ACQUIRING MANAGEMENT SUPPORT

Whenever any organization tries to adopt a new technology, they encounter a significant effort when determining how to apply it to their needs. With some training, organizations wrestle with time-consuming false starts before they become capable with the new technology. For the team interested in implementing automated tools, the challenge is how to best present the case for a new automation technology and its implementation to the management team.

Engineers need to influence management's expectations for the use of automation on projects. Engineers can manage expectations of others in the organization by forwarding helpful information to the management staff. Bringing up tool issues during strategy and planning meetings can also help develop better understanding of tool capabilities for everyone involved on a project or within the organization. An engineer can develop training material on the subject of automation and can suggest to management that a seminar be scheduled to conduct the training.

The first step in moving toward a decision to automation on a project requires that the team should be able to show to management understanding of the appropriate application of automation for the specific need at hand. For example, the team needs to check early on whether management is cost-averse and would be unwilling to accept the estimated cost of automated tools for a particular effort. If so, executive need to convince management about the potential return on investment by conducting cost/benefit analysis.

If management is willing to invest in an automated tool, but is unable or unwilling to staff a team with individuals having the proper software skill level or to provide for adequate tool training, the team needs to point out the risks involved and/or may need to reconsider a recommendation to automation.

Management needs to be made aware of the additional cost involved when introducing a new tool, not only for the tool purchase, but for initial schedule/cost increase, additional training costs, and for enhancing an existing process or implementing a new process.

Automation represents flexible technology, which provides several ways to accomplish an objective. Use of this technology requires new ways of thinking, which only amplifies the problem of tool implementation. The issues that organizations face when adopting automated systems include those outlined below:

- Finding/hiring tool experts.
- Experimenting/Using the correct tool for the task at hand.

- Developing and implementing an automated process, which includes developing automated design and development standards.

- Analyzing applications to determine those that are best suited for automation.

- Analyzing the requirements to determine suitability for automation.

- Training the team on the automated process, automated design, development, and execution.

- Initial increase in schedule and cost.

4.3 AUTOMATION TOOL ACQUISITION

Tool acquisition represents the second phase of the ALM. This phase guides the engineer through the entire tool evaluation and selection process, starting with confirmation of management support. Since a tool should support most of the organizations' testing requirements, whenever feasible the engineer will need to review the system's engineering environment and other organizational needs and come up with a list of tool evaluation criteria enabling the reader to make an informed decision with regard to the types of tests to be performed on a particular project. The engineer then needs to define an evaluation domain to pilot the tool. Finally, after all those steps have been completed, engineer can make vendor contact to bring in the selected tool(s). Then tool is evaluated, with sample criteria provided.

4.4 AUTOMATION (ADAPTIVE) INTRODUCTION PROCESS

The process of introducing automation to a new project team constitutes the third phase of the ALM. This phase outlines the steps necessary to successfully introduce automation to a new project, which are summarized in the following sections.

4.4.1 PROCESS ANALYSIS

Process analysis ensures that an overall process and strategy are in place and are modified, if necessary, to allow automation to be introduced in a successful fashion. The engineers define and collect test process metrics in order to allow for process improvement. Here test goals/objectives and strategies need to be defined and test process needs to be documented and communicated to the team. In this phase, the kinds of process applicable for the technical environment will be defined, that can be supported by automated tools.

During the process analysis, techniques are defined. Plans for user involvement are assessed, and team personnel skills are analyzed against requirements and planned activities. Early team participation is emphasized, supporting refinement of requirement specifications into terms that can be adequately analyzed while also supporting team understanding of application requirements and design.

4.4.2 TOOL CONSIDERATION

The tool consideration process includes steps that investigate whether incorporation of automated tools that have been brought into the company without a specific project in mind now would be beneficial to a specific project, given the project requirements, available environment, personnel resources, user environment, platform, and product features of the application under execution. Schedule is reviewed to ensure sufficient time for tool setup and development of requirements hierarchy; potential tools and utilities are mapped to requirements, tool compatibility with the application and environment is checked, and workaround solutions are found for incompatibility problems.

4.5 AUTOMATION PLANNING, DESIGN, AND DEVELOPMENT

Planning, design, and development is the fourth phase of the ALM. These subjects are summarized in the following sections.

4.5.1 AUTOMATION PLANNING

The planning stage represents the need to review long–lead-time test planning activities. During this phase, the test team identifies procedure creation standards and guidelines; hardware, software, and network required to support environment; test data requirements; a preliminary schedule; performance measure requirements; a procedure to control configuration and environment; as well as defect-tracking procedure(s) and associated tracking tool(s).

The plan contains the results of each preliminary phase of the structured methodology (ALM). The plan will define roles and responsibilities, project schedule, test planning and design activities, environment preparation, risks and contingencies, and acceptable level of thoroughness (acceptance criteria). Plan appendices may include procedures, naming conventions, procedure format standards, and a procedure traceability matrix.

The environment setup is part of planning. It represents the need to plan, track, and manage environment setup activities, where material procurements may have long lead times. The team needs to schedule and track environment setup activities; install environment hardware, software, and network resources; integrate and install environment resources; obtain/refine databases; and develop environment setup scripts and test bed scripts.

4.5.2 AUTOMATION DESIGN

The design component addresses the need to define the number of criteria to be performed, the ways that automation will be approached (paths, functions), and the conditions that need to be exercised. Design standards need to be defined and followed.

An effective program, incorporating the automation of software testing, involves a mini-development lifecycle of its own, complete with strategy and goal planning, test requirement definition, analysis, design, and coding. Similar to software application development, requirements must be specified before design is constructed. Requirements need to be clearly defined and documented, so that all project personnel will understand the basis of the effort. Requirements are defined within requirement statements as an outcome of requirement analysis.

After requirements have been derived using the described techniques, procedure design can begin. Procedure design consists of the definition of logical groups of procedures and a naming convention for the suite of procedures. With a procedure definition in place, each procedure is then identified as either an automated or a manual test. During the planning phase, the team gets an understanding of the number of techniques being employed and an estimate for the number of procedures that will be required. The team also will have an estimate of the number of procedures that will need to be performed manually, as well as with an automated tool.

Much like a software development effort, the program must be mapped out and consciously designed to ensure that activities performed represent the most efficient and effective process for the system under execution. Program resources are limited, yet ways of executing the system are endless. A design is developed to portray the effort, in order to give project and test personnel a mental framework on the boundary and scope of the program.

Following analysis, the team develops the program design models. The first of these design models, the *program model*, consists of a graphical illustration that depicts the scope of the program. This model typically depicts the techniques required to support the dynamic effort and also outline static strategies.

Having defined a program model, the team constructs an *architecture*, which depicts the structure of the program and defines the way that procedures will be organized in support of the test effort.

The next step in the procedure design process (see Table 1) is to identify those procedures that stand out as being more sophisticated, and as a result are required to be defined further as part of detailed design. These procedures are flagged and a detailed design document is prepared in support of the more sophisticated procedures.

Following detailed design, data requirements are mapped against the defined procedures. To create a repeatable, reusable process for producing procedures, the team needs to create a document that outlines procedure design standards. Only when these standards are followed can the automated program achieve real efficiency and success, by being repeatable and maintainable.

Table 4.1: Automation Procedure Design Process

Step	**Description**
1	**Architecture Review**. The team reviews the architecture in order to identify the techniques that apply.
2	**Procedure Definition (Development Level)**. A procedure definition is constructed at the development level, identifying the procedure series that applies for the various design components and techniques.
3	**Procedure Definition (System Level)**. A procedure definition is constructed at the system level, identifying the procedure series that applies for the various techniques.
4	**Procedure Design Standards**. Design standards are adopted and a unique naming convention is adopted that distinguishes the procedures on the project from procedures developed in the past or on other projects.

5	**Manual Versus Automated Tests**. Procedures is depicted as being either performed manually or as an automated test.
6	**Test Procedures Flagged for Detailed Design**. Test procedures which stand out as more sophisticated are flagged. These test procedures then defined as part of detailed design.
7	**Detailed Design**. Those procedures flagged as part of step 7 are designed in further detail within a detailed design file or document. Detailed design consist of pseudo-code of algorithms, preliminary test step definition, or pseudo-code of test automation programs.
8	**Data Mapping**. Procedure matrix is changed to reflect test data requirements for each test procedure.

The exercise of developing the procedure definition not only aids in development, but helps to quantify or bound the effort. The development of the procedure definition involves the identification of the suite of procedures that need to be developed and executed in support of the effort. The design exercise involves the organization of procedures into logical groups and the definition of a naming convention for the suite of test procedures.

At the system level, it may be useful to develop a detailed design for sophisticated tests. This might involve procedures that perform complex algorithms, consist of both manual and automated steps, and test programming scripts that are modified for use in multiple test procedures. The first step is to review the procedure definition at the system test level. This review is conducted for the purpose of identifying those procedures that stand out as being more sophisticated and that, as a result, are required to be defined further as part of detailed test design.

Detailed design may take the form of program pseudo-code, when programming is required. The detailed design may be represented simply as a sequence of steps that need to be performed in support of an execution. When programming variables and multiple data values are involved, the detailed design may reflect the programming construct of a loop supporting an iterative series of execution involving different values, together with a list or table identifying the kinds of data or ranges of data required for the execution.

Following the performance of detailed design, data requirements need to be mapped against the defined procedures. Once data requirements are outlined, the team needs to plan the means for obtaining, generating, or developing the data.

The structure of the program (architecture) is commonly portrayed in two ways. One procedure organization method involves the logical grouping of procedures with the system application design components, and is referred to as a *design-based architecture*. Another method represents a technique perspective and associates procedures with the various kinds of techniques represented within the program model, and is referred to as a *technique-based architecture*.

An understanding of execution techniques is necessary when developing design and the program design models. Personnel performing need to be familiar with the techniques associated with the white box and black box test-approach methods.

White box test techniques are focused at exercising software program internals; *black box* techniques compare the application under test behavior against requirements that address testing via established public interfaces such as the UI or the published API.

4.5.3 AUTOMATION DEVELOPMENT

For automation be reusable, repeatable, and maintainable, development standards need to be defined and followed.

After performing analysis and design, the team is now ready to perform development.

Keep in mind that the design and development activities follow an iterative and incremental approach, in order to address the highest risk functionality up front. Table 2 correlates the development phases to the process phases. The automation processes and steps outlined in the table are strategically aligned with the development process, and the execution of these steps results in the refinement of procedures at the same time wen software developers are creating the software modules. Automated and/or manual test procedures are defined during the integration phase with the intention of reusing them during the system test phase.

Table 4.2: Development/Test Relationship

Phase	**Development Process**	**Test Process**
Module (Unit) Development	Design module from requirements.	Perform test planning and test environment setup.
	Code module.	Create test design and develop test data.
	Debug module.	Write scripts or record scenario using module.

	Unit test module.	Debug automated test script by running against module. Use tools that support unit testing.
	Correct defects.	Rerun automated script to regression as defects are corrected.
	Conduct performance testing.	Check that system is scalable and will meet performance requirements.
Integration	Build system by connecting modules. Integration-test connected modules. Review trouble reports.	Combine unit scripts and add new scripts that demonstrate module interconnectivity. Use tool to support automated integration testing.
	Correct defects and update defect status.	Rerun automated script as part of regression, as defects are corrected.
	Continued performance testing activities.	Check that system is scalable and will meet performance requirements.

System Test	Review trouble reports.	Integrate automated scripts into system-level procedures where possible, and develop additional system-level test procedures. Execute system and record results.
	Correct defects and update defect status.	Rerun automated script as part of regression test as defects are corrected.
Acceptance Test	Review incident reports.	Perform subset of system testing as part of demonstration of user acceptance test.
	Correct defects.	Rerun automated script as part of regression test as defects are corrected.

Many preparation activities need to take place before development can begin. A development architecture is developed (described in the next section), which provides the team with a clear picture of the development preparation activities or building blocks necessary for the efficient creation of test procedures.

The team will need to tailor the sample development architecture to reflect the priorities of their particular project. Part of these setup and preparation activities involves the need to track and manage environment set up activities, where material procurements may have long lead times. Prior to the commencement of development, the team also needs to perform analysis to identify the potential for reuse of existing procedures and scripts within the automation infrastructure (reuse library).

The team needs to develop procedures according to development/execution schedule. And schedule needs to allocate resources and reflect development due dates, among other factors. The team needs to monitor development progress and produce progress status reports. Prior to the creation of a complete suite of procedures, the team performs a modularity relationship analysis.

The results of this analysis help to incorporate data dependencies, plan for workflow dependencies between executions, and identify common scripts that can be applied repeatedly to the effort. As procedures are being developed, the team needs to ensure that configuration control is performed for the entire bed to include design, scripts, and data, as well as for each individual procedure. The bed needs to be baselined using a configuration management tool.

Test, QA, Integration, Deployment development involves the development of procedures that are maintainable, reusable, simple, and robust, which in itself can be as challenging as the development of the application under execution. Development standards should be in place supporting structured and consistent development of automated execution. Development standards can be based on the scripting language standards of a particular tool.

The adoption of existing system development standards is generally a better approach than creating a standard from scratch. If no development standards exist within the organization for the particular tool scripting language, it's important for the team to develop script development guidelines. Such guidelines can include directions on *context independence*, which addresses the particular place where a procedure should start and where it should end. By developing procedures based on development guidelines, the team creates the initial building blocks for an automation infrastructure.

The automation infrastructure will store a library of common, reusable scripts. Throughout the effort and in future releases, the engineer can make use of the automation infrastructure to support reuse of archived procedures, minimize duplication, and thus enhance the entire automation effort.

4.5.4 AUTOMATION DEVELOPMENT ARCHITECTURE

Team members responsible for product development need to be prepared with the proper materials. Team member need to follow an architecture that includes, for example, a listing of the procedures assigned to them and a listing of the outcome of automated versus manual analysis. Also, team personnel need to decide when to automate. At times a test team might want to avoid automating using a GUI tool before the interface, whether API, character UI, or GUI, is stabilized, to avoid having to reengineer the automation in response to non–bug-related changes. At other times, the team might find workaround solutions when automating an unstable GUI, such as focusing automation on the known stable parts only.

The engineer needs to adhere to the procedure development and execution schedule, test design information, automated tool user manuals, and procedure development guidelines. Occupied with the proper instructions, documentation, and guidelines, test engineers will have the foundation that allows them to develop a more cohesive and structured set of test procedures. Repeating a process and repeatedly demonstrating a strong program depends on the availability of documented processes and standard guidelines such as the test development architecture.

4.5.5 TECHNICAL ENVIRONMENT

This activity needs to be supported by a technical environment, which facilitates the development of procedures. As a result, the environment needs to be set up and ready to go. This environment includes the technical environment, which may include facility resources as well as the hardware and software necessary to support test, development and execution. It needs to be ensured that there are enough workstations to support the entire team.

Environment setup activities can also include the use of an environment setup script to load data or restore a drive image, and to calibrate the tool to the environment. When tool compatibility problems arise with the application under execution, workaround solutions have to be identified. When developing procedures, it's important that the schedule for developing procedures is consistent with the execution schedule. It's also important that the team follow procedure development guidelines.

The team must ensure that the proper test room or laboratory facilities are reserved and set up. Once the physical environment is established, the team ensures that all necessary equipment is installed and operational. The plan defined the required technical environment and addressed environment planning. Within the environment section of the plan, the team has already identified operational support required to install and check out the operational readiness of the technical environment. The team needs to ensure that operational support activities have been properly scheduled and must monitor progress of these tasks.

Specific tasks and potential issues outlined in the plan should now have been addressed and resolved. Automated tools that apply should have been scheduled for installation and checkout. These tools now should be configured to support the team and be operational within the test environment.

The environment setup activity includes the need to track and manage environment setup activities, where material procurements may have long lead times. The hardware supporting the environment must be sufficient to ensure complete functionality of the production application. Environment hardware needs to support performance analysis. In cases where the environment utilizes hardware resources that are also supporting other development or management activities, special arrangements may be necessary during actual performance testing.

During system execution, the software configuration loaded within the execution environment must be a complete, fully integrated release with no patches and no disabled sections. The hardware configuration supporting the environment needs to be designed to support processing, storage, and retrieval activities, which may be performed across a local or wide area network, reflecting the target environment.

The environment design also needs to consider stress testing requirements. Stress and load tests may require the use of multiple workstations that will run multiple procedures simultaneously; some automated test tools include a virtual user simulation functionality that eliminates or minimizes the need for multiple workstations.

Data will need to be obtained with enough lead time to support refinement and manipulation to support requirements. Data preparation activities include the identification of conversion requirements, the preprocessing of raw files, loading of temporary tables, possibly in a relational database management system format, and the performance of consistency checks. During planning, the team defined and scheduled the environment activities.

Now the team needs to track the environment setup activities. Resources need to be identified to install hardware, software, and network resources into the environment and integrate and installed environment resources. The environment materials and the application under system need to be baselined within a configuration management tool. Additionally, execution environment materials may include test data and test processes.

The team needs to obtain and modify databases necessary to exercise software applications, and develop environment setup scripts and test bed scripts. The team should perform product reviews and validation of all source materials. The location of the environment for each project or task should be defined within the plan for each project.

4.6 EXECUTION AND MANAGEMENT OF AUTOMATION

At this stage, the team has addressed test, design and development. Environment setup planning and implementation was addressed consistent with the requirements and guidelines provided within the test plan.

With the test plan in hand and the environment now operational, it's time to execute the automation defined for the test program. When executing procedures, the team must comply with a procedure execution schedule, as discussed earlier. The procedure execution schedule implements the strategy defined within the test plan. Plans for unit, static analysis, code coverage integration, system, and user acceptance testing are executed. Together, these execution phases make up the steps that are required to execute the system as a whole. The various steps involved during execution and management are outlined below.

- When executing automation procedures, the team needs to comply with a procedure execution schedule. Following execution, outcome evaluations are performed and execution result documentation is prepared.

- Plans for unit, integration, system, and user acceptance testing are executed, which together make up the steps that are required to executive the system as a whole. With the unit testing, code profiling of software can be performed. “Profiling is a tuning process that determines whether an algorithm is efficient enough”. Profiling also discover instances where there is improper scaling of algorithms, instantiations, and resource utilization.

- Integration testing focuses on the application internals. In this, units are incrementally integrated and tested together based on control flow. Since units may consist of other units, also called *module testing*, may take place during unit testing.

- During system test, the engineer is testing the integration of parts that comprise the entire system. A separate team usually performs system-level executions. The team implements the procedure execution schedule and the system test plan.

- The team also performs analysis to identify specific components or functionality that are experiencing a greater relative number of problem reports. As a result of this analysis, additional test procedures and effort may need to be assigned to the components. Results analysis confirm whether executed execution procedures are proving to be worthwhile in terms of identifying errors.

- Each team needs to perform problem-reporting operations in compliance with a defined process. The documentation and tracking of system problem reports is greatly facilitated by an automated defect-tracking tools and techniques.

- The team manager is responsible for ensuring the execution according to schedule, and Team member are allocated and redirected when necessary to handle problems that arise during the execution effort. To perform this oversight function effectively, the manager needs to perform program status tracking and management reporting.

- Metrics provide the manager with key indicators of the coverage, progress, and the quality of the effort. During white box testing, the engineer measures the *depth* of testing, by collecting data relative to path coverage and test coverage. During black box functional testing, metrics collection focuses on the *breadth* of testing, to include the amount of demonstrated functionality and the amount of execution that has been performed.

4.7 AUTOMATION PROGRAM REVIEW AND ASSESSMENT

Automation program review and assessment activities need to be conducted throughout the product lifecycle, to allow for continuous improvement activities. Throughout the lifecycle and following execution activities, metrics need to be evaluated and final review and assessment activities need to be conducted to allow for process improvement? The various steps necessary for program review and assessment are outlined below.

- Following execution, the team needs to review the performance of the program to determine where changes can be implemented to improve the program performance on the next project. This program review represents the final phase of the Automated Lifecycle Methodology (ALM).

- Throughout the program, the team collected various metrics. The focus of the program review includes an assessment of whether the application satisfies acceptance criteria and is ready to go into production.

- As part of its culture, the team needs to adopt an ongoing iterative process of *lessons learned* activities. Such a program encourages engineers to take the responsibility to raise corrective action proposals immediately, when such actions potentially have significant impact on program performance. Throughout the entire lifecycle, it's good practice to document and begin to evaluate lessons learned at each milestone. The metrics that are collected throughout the lifecycle and especially during the execution phase help pinpoint problems that need to be addressed.

- Lessons learned, metrics evaluations, and corresponding improvement activity or corrective action need to be documented throughout the entire process in a central repository that's easily accessible.

- After collecting lessons learned and other metrics, and defining corrective actions, engineers also need to assess the effectiveness of the program to include an evaluation of the *program return on investment*. Engineers capture measures of the benefits of automation realized throughout the lifecycle in order to support this assessment.
- Teams can perform surveys to inquire about the potential value of process and tool requirements. A survey form is used to solicit feedback on the potential use of requirement-management tools, design tools, and development tools.

4.8 FACTORS TO MAKE APP INFRASTRUCTURE AGNOSTIC

The 12 Factor App requirements make sure that the app can be made infrastructure agnostic, be it cloud or standalone. From an Engineering perspective, we have to make sure the below standards are maintained and the arrangements of the infrastructure/resources are as per the twelve factor requirements below:

4.8.1 CODEBASE

One codebase tracked in revision control, many deploys. Each app should have only one code base which is version tracked with proper version maintenance of the code. One App cannot have different code bases. There can be different deploys of the same app running. Deploys are running instances of the same app. Each deploy can have different version of code depending upon the code build pipeline status.

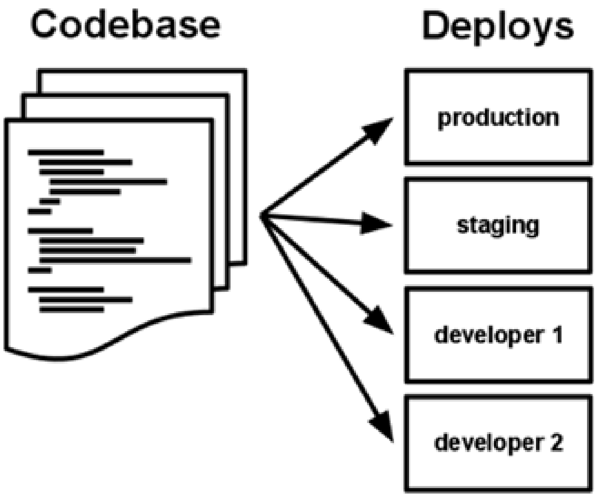


Figure 4.1: Codebase Deployments

4.8.2 DEPENDENCIES

Explicitly declare and isolate dependencies. The dependencies should not be tied implicitly inside app or classes and should be declared in one place for the whole app. Any new developer looking at the app should be able to just pull in dependencies from the dependency management tool and not worry much about pulling in implicit dependencies and should be able to start building and running the app within no time.

4.8.3 CONFIGURATION

Store config in the environment. Each environment (deploys) should have its own specific variables declared in an environment specific config files. The easy test of it is that the app should be able to be made open source anytime for its config to be kept separated.

4.8.4 BACKING SERVICES

Treat backing services as attached resources. All the resources whether local DBs or external services should be loosely coupled and can be replaced easily in future if we want to. The backing services should be easily configurable and can be easily replaced in case of failures.

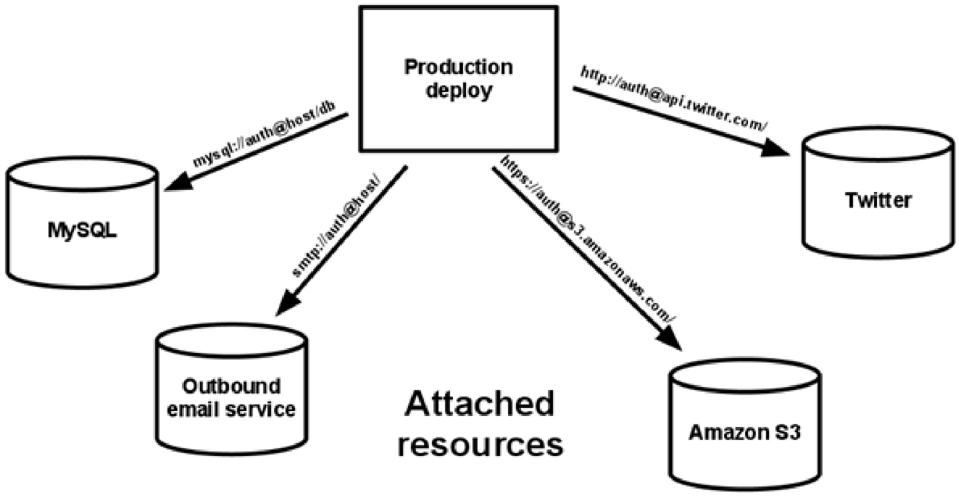


Figure 4.2: Backing Services

4.8.5 BUILD, RELEASE, RUN

Strictly separate build and run stages. The build stage for an app should be separate and be executed before the run(release) stage to make sure the build doesn't have any issue before running the app itself. This helps to isolate the problem and track it easily to fix any issues in lesser time.

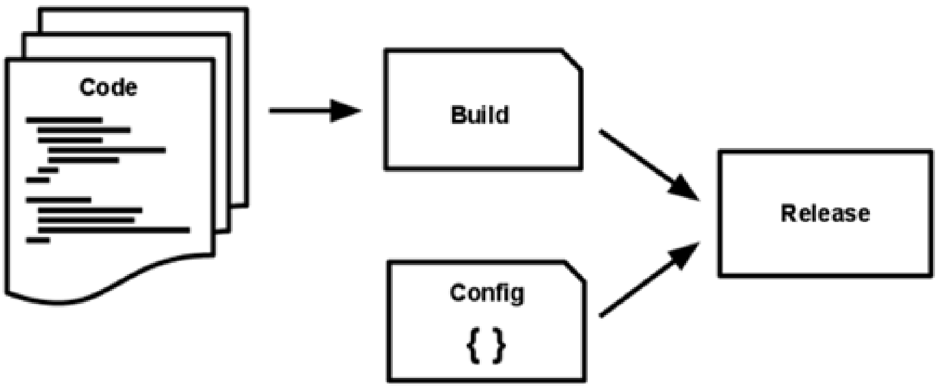


Figure 4.3: Build Release Run

4.8.6 PROCESSES

Run the application as stateless processes. Everything should be stored in a data store and any future processes should not depend on the cached data on filesystem or any place. Each process should be independent on its own and should not depend on data of any other process' state.

4.8.7 PORT BINDING

Export services via port binding. Expose apps via URL: port for easy access to other external or internal applications. Each app and its services should be easily accessible via URL endpoint whether it is used by an external or internal API or application.

4.8.8 CONCURRENCY

Scale out via the process model. Each app should have its own process types/groups defined like below to scale out the processes as per the requirements in that particular process type. Array of process types and number of processes of particular type is known as the *process formation*. App should not demonize or write PID files in system, instead it should let the OS handle the processes spun out from the APP which can be distributed over different systems as per the process loads.

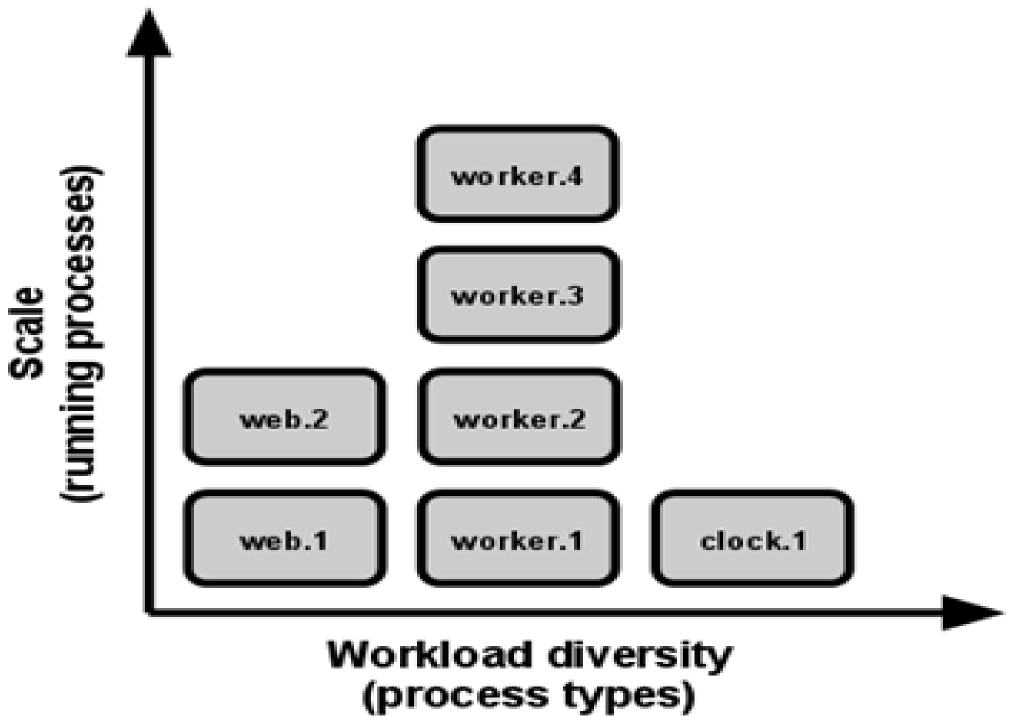


Figure 4.4: Workload Diversity

4.8.9 DISPOSABILITY

Maximize robustness with fast startup and graceful shutdown. Processes should be able to start up and shutdown in small amount of time and should be able to resume the program execution in minimal amount of time. The system should be robust enough to handle such failures and get back online in no matter of time.

4.8.10 DEV/PROD PARITY

Development, staging, and production should be as similar as possible. There should be minimal gaps in processes to take app from development to production. The different gaps which need to be minimized are as below:

1. *The time gap*: A developer may work on code that takes days, weeks, or even months to go into production.
2. *The personnel gap*: Developers write code, ops engineers deploy it.

3. *The tools gap*: Developers may be using a stack like Nginx, SQLite, and OS X, while the production deploy uses Apache, MySQL, and Linux.

4.8.11 LOGS

Treat logs as event streams. A 12 factor system should not worry about storing log files, it should just be writing all its logging events to stdout and let the deployment infrastructure take care of handling logs. The log are events of things happening in the app sorted in a timely manner which need to be routed to appropriate tools to handle logs for analysis and alerting.

4.8.12 ADMIN PROCESSES

Run admin/management tasks as one-off processes. All the admin related tasks whether it is in development or production should be executed as on-off processes which once triggered should handle all the admin related task in one go than executing multiple processes for the same.

Depending upon the organization structure and priority, the above factors should be incorporated in an application. Below are some the priorities currently in place for the Cloud Native Architecture requirements:

Table 4.3: Admin Process Priority and Description

High = Cloud ready, Medium = Cloud friendly, Low = Cloud Resilient

Factor	Priority	Description
1. Codebase	High	One code base tracked in revision control, many deploys
2. Dependencies	High	Explicitly declare and isolate dependencies
3. Configuration	Medium	Store config in the environment
4. Backing Services	Medium	Treat backing services as attached resources
5. Build, Release, Run	Low	Strictly separate build and run stages
6. Processes	High	Execute the app as one or more stateless processes
7. Port Binding	Low	Export services via port binding

8. Concurrency	High	Scale out via the process model
9. Disposability	Medium	Maximize robustness with fast startup and graceful shutdown
10. Dev/Prod Parity	Medium	Keep development, staging, and production as similar as possible
11. Logs	High	Treat logs as event streams
12. Admin Processes	Low	Run admin/management tasks as one-off processes

Chapter 5

ADAPTIVE AUTOMATION TESTING DESIGN PATTERNS

CHAPTER 5
ADAPTIVE AUTOMATION TESTING DESIGN PATTERNS

Keywords: Adaptive, Automation, Design, Development, Testing, Patterns

5.1 INTRODUCTION

In this Chapter we will discuss about the Adaptive Software Development, Testing and Automation Design Patterns Process.

Adaptive software development focuses on the rapid creation and evolution of software.
This method grew out of the rapid application development method. Adaptive Software Development replaces the traditional models with a repeating series of speculate, *collaborate*, and *learn* cycles.

Adaptive software Design (ASD) is a software development process that grew out of rapid application development (RAD). It includes the principle that continuous adaptation of the process to the work at hand is the normal state of affairs.

Adaptive software replaces the traditional models with a repeating series of *speculate*, *collaborate*, and *learn* cycles.

This provides for continuous learning and adaptation to the emergent state of the project. The characteristics of an ASD life cycle are following goal focused, clear picture of features, iterative in nature, time Bound, risk driven, and change tolerant. ASD is also a type of agile software development methodology.

ASD is made of following steps (*speculate*, *collaborate*, and *learn*). So let us see those.

The word ***speculate*** refers to the concept that all stakeholders may be comparably wrong for certain aspects of the project's when discussing the definition of same. During design, the project is initiated and adaptive cycle planning is conducted. Adaptive cycle planning uses project initiation information, mission statement, project constraints, delivery dates, and basic requirements, to define the set of release cycles (software increments)

Collaboration refers to the efforts for managing the work based on predictable parts of the environment (planning and guiding them) and adapting to the uncertain surrounding mix of changes caused by technology, requirements, stakeholders, software vendors.

The ***learning*** cycles, are based on the short iterations with design, build and testing. During these the knowledge is structured by making small mistakes based on false assumptions and correcting those mistakes, leading to more experience and eventually complete understanding problem domain.

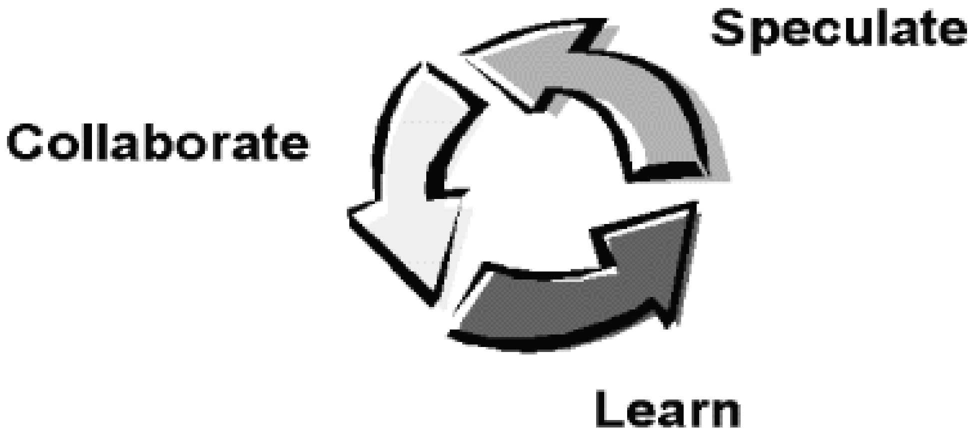


Figure 5.1: Learn Speculate Collaborate

An adaptive software development approach is based on observation" and that means both the design of product and the process to create the solution are continuously adjusted with each other based on observation throughout the project an adaptive approach encourages changes throughout the life cycle of project to optimize the design of the overall solution.

An adaptive process is best-suited for projects with uncertainty. It is somewhat ridiculous to try to develop a detailed plan for the effort upfront for uncertain problems. That’s an ideal situation for an adaptive process in this stakeholders need to start with general direction based on whatever is upfront and continue to refine the direction based on observations throughout the duration of the project.

Focuses is on the rapid creation and evolution of software systems. There is never a stoppage where the software is finished; there are just temporary stable periods between releases. Dynamic cycle in ASD provides for continuous learning and adaptation to the emergent state of the project.

The focus of adaptive software development is in the programming code. Instead of planning the software out well in advance, Programmers have a basic idea in their mind and those kicks off. When parts need changing or adapting to a new system, the programmers simply do it.

Overall, removing the pre-planning steps allows the developers/designers to make the software very quickly. Sometime it may result in software that doesn’t perform the precise functions required, but it’s not a problem. The developmental cycle in this process is made short such that a new version with added features can come out very quickly. Rapid prototyping is the related to both adaptive software development and rapid application development.

The other SDLC models are more oriented to the practices of stability, predictability and decreasing returns. The industry, moving to increase environments, those are unpredictable, nonlinear, and fast approaches.

Adaptive Software Development (ASD) evolved to address above mentioned issues. It focuses on the important factor from the management’s perspective, to enhance the ability to manage product development.

In Jim Highsmith's definition, "Adaptive Software Development framework is being matured on years of experience with traditional Software Development methodologies, consulting on, practicing, and writing about Rapid Application Development (RAD) techniques and working with high-technology software companies on managing their product development practices".

Waterfall model is characterized by linearity and predictability. It is viewed as a sequence of **Plan** → **Build** → **Implement**.

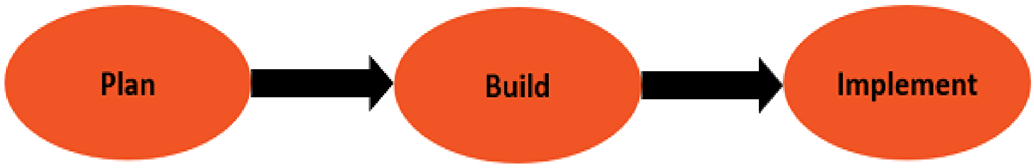


Figure 5.2: Plan Build Implement

The Evolutionary Lifecycle models like Spiral model moved the Deterministic approach to the Adaptive approach, with **Plan** → **build** → **Revise Cycles**.

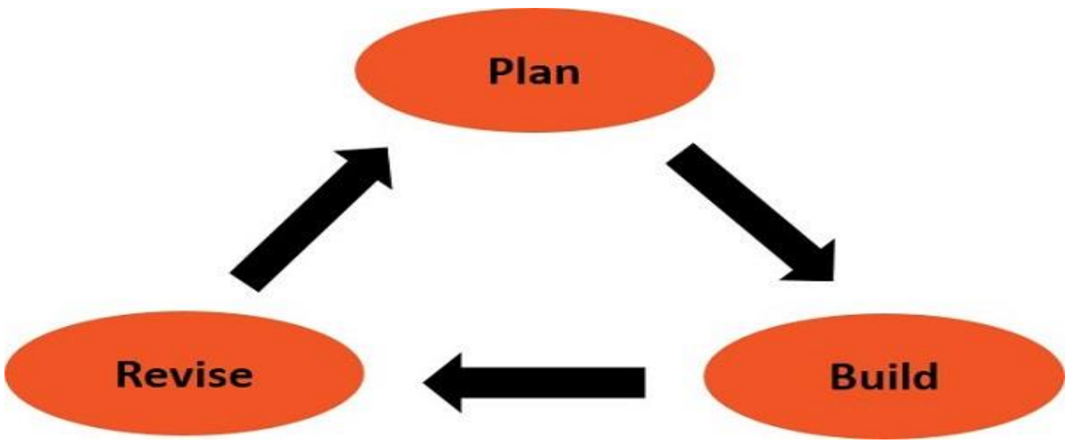


Figure 5.3: Plan Build Revise

However, the mind set remained Deterministic with long-term predictability turns into short-term predictability. Hence the practices of Evolutionary Lifecycle models are found to be less Deterministic.

5.2 THE ADAPTIVE LIFE CYCLE

The Adaptive model is built cyclical like the Evolutionary model, and the names of the phase reflect the unpredictable nature of increasingly complex systems.

Adaptive Development goes further as following −

- It explicitly replaces Determinism with Emergence.
- It traverse from change in life cycle then in management style.

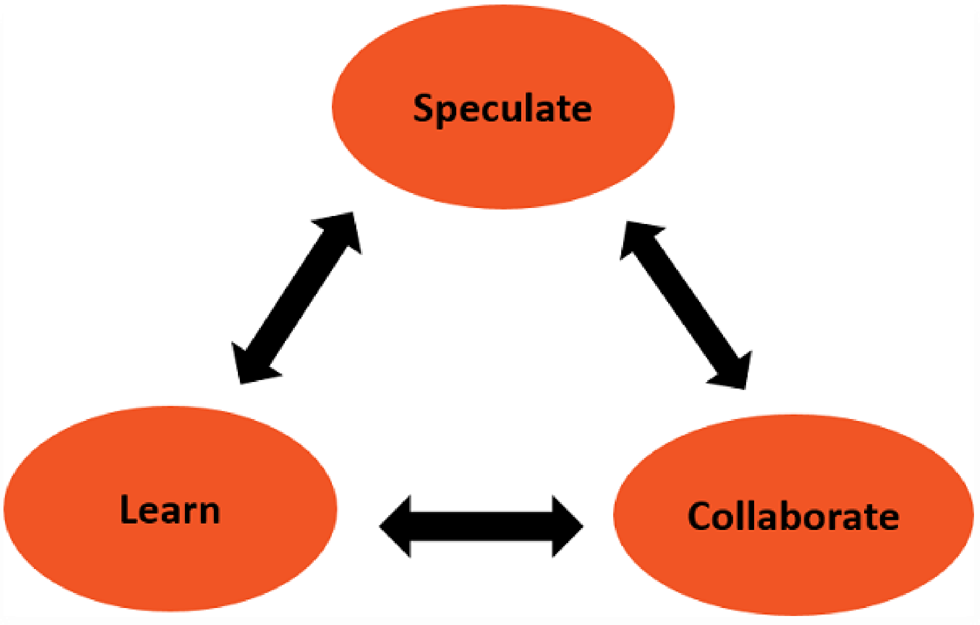


Figure 5.4: Speculate Learn Collaborate

The phases in ASD Lifecycle are as following −

- **Speculate** − the deterministic word planning is replaced by speculate, planning of product specifications or project management etc...
- **Collaborate** − Collaborate represents balance between
    - Managing the traditional project management style and
    - Starting and maintaining the collaborative environment for emergence.
    - Collaborative Activities, build products, keeping up the pace of changes in the development environment.

- **Learn** − Learn aims stakeholders, to use the results of each development cycle to learn the direction of the next iteration cycle.

Concepts of Adaptive Software Development as following.

5.3 COMPLEX ADAPTIVE SYSTEMS (CAS) THEORY

Brian Arthur with his team, at the Santa Fe institute, used the CAS theory to revolutionize the understanding of Sciences, Evolution, and Economic Systems.

HE culminated that the new world is one of increasing returns, instability, and inability to determine cause and effect.

So there is differences of behavior, style, and culture with Management Techniques, Strategies and Understanding.

5.4 COMPLEX SOFTWARE DEVELOPMENT

So as per above observations even the software development organizations are accruing similar challenges.

1) Second World is represented by the Deterministic development, derived from management practices that are rooted with the basics of stability and predictability (decreasing returns)
2) Second World is represented by the industries moving from decreasing to increasing return environments those are unpredictable, nonlinear and fast.

To address the issues of this second world, a framework is being designed called as Adaptive Software Development which is different from the Deterministic Software Development.

The Adaptive Software Development focuses on addressing the complex problems of engineering environment and Management of organizations.

- ASD for the development life cycle.

- Adaptive Management Techniques with different mind-set from then traditional project management practices and techniques.

There are two perspective of Adaptive Software Development (ASD): −

1) **Conceptual perspective** determined on the Complex Adaptive Systems (CAS) theory.
2) **Practical Perspective** requires following
    - Long Work experience with Deterministic software development methodologies.
    - Consulting, and practicing, about Rapid Application Development (RAD) working with Hi-tech software organizations on managing product development.

Now let's focus on conceptual perspective of Adaptive Software Development.

5.5 COMPLEX ADAPTIVE SYSTEMS (CAS) CONCEPTS

On the basis of CAS concepts Adaptive Software Development is based on following concepts:

- **Emergence**
- **Complexity**
- **Quality**

5.5.1 Emergence

In complex software product-development projects, outcomes are inherently unpredictable. However, successful products emerge from such environments most of the time.

This can happen by Emergence, as illustrated in CAS theory. It can be understood by a simple example, flocking behavior of flying birds.

When it's been observed, that

- Each bird tries to

   - Maintain a minimum distance from other objects in the environment, including other birds, Match velocities with birds in its neighborhood bird, Move towards the perceived center of mass of birds in its neighborhood.

- Hence it's been observed that there is no rules of behavior of whole group but observation is about the behavior of individual birds. However, there exists an emergent behavior, the flocking of birds. When diverted birds rush to manage path, the flock splits around obstacles and reforms on the other side.

Hence it can be understood that the most difficult mental model changes in Adaptive Development from ways of managing and organizing that individual freedom to creative new order emerges unpredictably from spontaneous self-organization.

Development, and emergence are the most important concept from the management perspective.

5.5.2 Complexity

In the Software Development context, Complexity is about the individuals of a team such as the developers, customers, vendors, competitors, and stockholders, their numbers and their speed, size and technological complexity.

5.5.3 Quality

In a complex environment, the practice of "Do things right from start" does not work as right things cannot be predicted at the beginning. Aim should be to produce the right value as a final result. However, in complex software development environment, the combinations and permutations of value components like scope (features, performance, defect levels), schedule, and resources is vast that there can never be an optimum value. Hence, the focus should be to shift to deliver the best value in the competitive environment.

5.6 RAD PRACTICES

RAD Practices generally contains following −

- Evolutionary Lifecycle, Focus Groups, Sessions, Reviews, Time-bound Project Management, Continuous Software Engineering, Dedicated Teams with war rooms

The RAD projects have an inherent adaptive, emergent flavor. Microsoft process follows the RAD practices are both examples of Adaptive Development in action. Giving a label and realizing about a growing body of scientific knowledge (i.e., CAS theory) explains why they work. This should provide a basis for extensive use of ASD practices.

Following are has six basic characteristics of Adaptive Software Development Lifecycle

5.6.1 MISSION-FOCUSED

For many projects, requirements may be uncertain at the beginning but the goal that guides the team is well articulated manner. Mission statements act as guide that encourage exploration in the beginning but have a linear focus over the course of a project. A mission provides boundaries in place of a fixed destination. Mission statements provide direction with criteria for making critical project trade-off decisions.

Without a focused and constant mission refinement practice, iterative lifecycles become oscillating lifecycles, with no progress in the development process.

5.6.2 FEATURE-BASED

The Adaptive Software Development Lifecycle should be based on application features and not on process. Features are developed side by side with an iteration based on the stakeholder’s priorities.

Feature evolves over iterations when the customers provide feedback. Primary Feature may consists of that provide direct results to the customer after implementation also a customer-oriented document (user manual). Secondary features are documents i.e. the data model, may be defined as deliverables.

5.6.3 ITERATIVE

The Adaptive Software Development Lifecycle is iterative which focuses on frequent releases to get feedback, learning and setting the proper direction for continuous development.

5.6.4 TIME-BOXED

In Adaptive Software Development Lifecycle, the iterations are time-bound. But time-boxing in Adaptive Software Development is not time deadlines. Team does not work for long hours and works in collaborative environment hence no compromise on the quality of the deliverables.

Time-boxing is considered as a direction for focus and forcing hard trade-off decisions whenever required. In an uncertain environment, where change rates are high, there should be a periodic forcing function to get the work finished.

5.6.5 RISK-DRIVEN

In Adaptive Software Development, the iterations are managed by identifying and evaluating the risks.

5.6.6 CHANGE-TOLERANT

Adaptive Software Development is tolerant to change, and look at this as the ability to incorporate better advantage, but should not be a problem for development.

5.7 ADAPTIVE SOFTWARE DEVELOPMENT - PRACTICES

The Adaptive Software Development driven by continuous adaptation, with the lifecycle equipped to accepting continuous change.

Hence the Adaptive Software Development Lifecycle explores Continuous learning, Change Orientation, Re-evaluation, and Peering into a changing future and does intense collaboration among stakeholders.

5.7.1 ADAPTIVE SDLC

Adaptive Software Development combines RAD with Best Practices, such as Project initiation, Planning, Concurrent Engineering, Quality review and Final QA and release.

Adaptive Software Development practices is illustrated as follows −

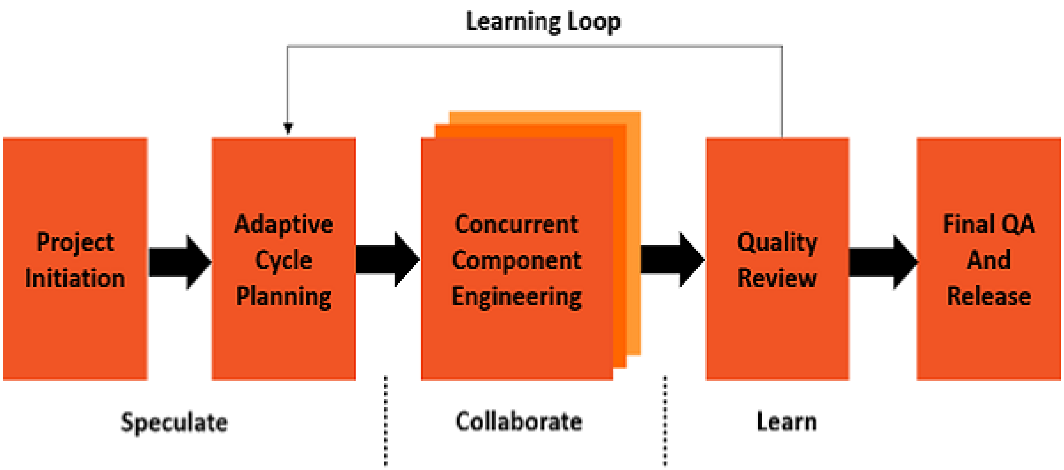


Figure 5.5: Adaptive Learning Loop

As mentioned above, Adaptive Software Development practices are spread across the three phases as follows −

- **Speculate** includes Initiation and planning, it does Project Initiation then Establishes time-box for the entire project, thereafter decide on the number of iterations and assign a fixed time to each one to develop a theme or objective for each of the iterations and then assign features to each iteration

- ○ **Collaborate** – Starts Concurrent feature development, Collaboration for distributed teams, for smaller projects, and for larger projects
- ○ **Learn −** Quality Review results quality from the stakeholders perspective and it result quality from a technical perspective, The functioning of the delivery team and the practices team members are utilizing The project status

Now let us explore the three points as following.

5.7.2 SPECULATE - INITIATION AND PLANNING

Speculate phase has two activities

1) Initiation
2) Planning

There are five practices need to be processed repetitively during the initiation and planning phase.

- Project initiation
- Establishing time-bound planning.
- Define the number of iterations and assign fix time for each iteration.
- Develop objective for each of the iterations
- Assign features to every ach iteration

5.7.2.1 PROJECT INITIATION:

Project Initiation involves, setting the project's mission and objectives then Understanding constraints, does establishing the project organization, Identifying and outlining requirements and Making initial size and scope estimates and finally Identifying key project risks

The project initiation data should be gathered in a preliminary sessions, considering speed as the major aspect. Initiation can be completed in around a week and effort for a small to medium sized projects, or two to three weeks effort for larger projects. Requirements are gathered in detail to identify features and establish an overview of the object, data, or other architectural model.

5.7.2.2 ESTABLISHING TIME-BOX FOR THE ENTIRE PROJECT:

The time-box for the whole project should be established, based on the scope, requirements, estimates, and resource availability that result from project initiation work. Speculating does not overrule estimating, but it means accepting that estimates may go wrong.

5.7.2.3 ITERATIONS AND TIME-BOX:

Project stakeholders should decide on the number of iterations and the individual iteration lengths based on the overall project scope and the degree of uncertainty.

For a small to medium sized application −

- Iterations vary from four to eight weeks.
- Some projects work best with around two-week iterations.
- Some projects might require around more than eight weeks.

Choose the time, based on what works. Once decision has been taken on the number of iterations and the lengths of each of the iterations, assign a schedule to each of the iterations.

5.7.2.4 DEVELOP A THEME OR OBJECTIVE:

Stakeholders should develop a theme/objective for every iteration and same should be similar to the Sprint Goal in Scrum. Each iteration should deliver a feature set that demonstrates the product functionality and making the product visible to the customer to enable review and feedback.

Inside iterations, the builds should deliver working features on a preferably daily basis enabling integration process and making the product visible to the development team. Testing should be an ongoing, and should be integral part of the feature development. Theme should not be delayed until the end of the project.

5.7.2.5 ASSIGN FEATURES:

Stakeholders should assign features to each iteration. The important criteria for this feature assignment is that every iteration should deliver a visible set of features with considerable functionality to the customer.

During the assignment of features to the iterations, Development team should come up with the feature estimates, risks, and dependencies and pass them to the customer. And Customers should decide on feature prioritization, with the information provided by the development team.

Iteration planning is feature-based and done as a team with all Stakeholders. Experience has shown that Iteration planning provides better understanding of the project than a task-based planning manager. Thereafter, feature-based planning reflects the uniqueness of each project.

5.7.3 COLLABORATE FEATURE DEVELOPMENT

During the Collaborate phase, focus should always be on the development. The Collaborate phase has two activities −

- The Development team should collaborate and deliver working software.
- The project managers should facilitate collaboration with concurrent development activities.

Collaboration is an act of shared creation that keeps together the development team, the customers and the managers. This is fostered by trust and respect.

Teams should collaborate on Technical problems, Business requirements and Rapid decision making

Following are the rules for the Collaborate phase in ASD.

5.7.3.1 COLLABORATION FOR DISTRIBUTED TEAMS

In the projects involving distributed teams, the following should be considered varying alliance partners with Broad-based knowledge, the way people interact, the way they manage interdependencies

5.7.3.2 COLLABORATION FOR SMALLER PROJECTS

In smaller projects, when team members works in physical proximity, Collaboration with informal and whiteboard scribbling should be encouraged, as this is found to be effective, All Stakeholders should be collaborated in small projects.

5.7.3.3 COLLABORATION FOR LARGER PROJECTS

Larger projects needs added practices, collaboration tools, and project manager interaction and should be arranged on the contextual basis.

5.7.4 LEARN - QUALITY REVIEW

ASD encourages the concept of 'Experiment and Learn'. Learning from the mistakes/experimentation requires that the team share partially completed artifacts early, in order to −

- Find mistakes
- Learn from them
- Find small problems before they become large.

At the end of each iteration, below are four general categories of things to learn

- Result quality from the customer's perspective
- Result quality from a technical perspective
- The functioning of the team

- The project status

5.7.4.1 RESULT QUALITY FROM THE CUSTOMER'S PERSPECTIVE

Getting feedback from the customers is the first priority in Adaptive Software Development projects. These are designed to explore a working model of the application and record customer change requests.

Customer focus group sessions are facilitated sessions, but rather than generating requirements or defining project plans, they are designed to review the application itself. The customers provide feedback on the working process resulting from an iteration.

5.7.4.2 RESULT QUALITY FROM A TECHNICAL PERSPECTIVE

In ASD projects, cyclic review of technical artifacts should be given much importance. Code Reviews should be done on regularly. Reviews of other technical artifacts, can be conducted weekly or at the end of an iteration as per the requirements.

In Adaptive Software Development projects, the team should keep an eye on its own performance periodically. It encourage the teams to learn about approach and their work, together as a group.

Iteration-end retrospectives checks periodic team performance self-review such as and it determine what is not working then what the Team needs to do more or less.

5.7.4.3 THE PROJECT STATUS

The Project review helps to plan future course of action. In the ASD projects, determining the project status is feature-based approach, the end of each iteration marked by completed features resulting in working software.

The Project Status review should include

- Where about of the project?
- Details of project and the plans?

- And where should the project be at this time?

Hence the project team and the customers need to continuously ask themselves, "What have been learned so far, and does it change perspective of goal?"

5.8 ADAPTIVE S/W DEVELOPMENT - MANAGEMENT

Traditional software management is characterized by the term command-control.

Organizations are steeped in a tradition of optimization, efficiency, predictability, control, and rigor and process improvement. But the emerging information age economy requires adaptability, speed, collaboration, improvisation, flexibility, innovation, and result.

Business review and management books has terms such as empowerment, participative management, learning organization, human-cantered management, etc., but these are not put into managing modern organizations.

In the context of Adaptive Software Development, gap looks wider and there is a necessity to consider the Adaptive management techniques that have been proven successful in other fields.

5.8.1 ADAPTIVE MANAGEMENT

Adaptive management is successful in the environments where the resource managers worked with stakeholders and scientists as a team, and has following goals −

1) Learn how to manage systems respond to human interventions.
2) Improve resource policies and practices for future planning.

The principle behind adaptive management is that management activities are experiments as their outcomes cannot be reliably predicted beforehand. These experiments are always used as learning opportunities for the improvements in the future planning.

Adaptive management is put on board to increase the ability to respond timely in the face of added information and in a setting of varied stakeholder objectives/ preferences. It encourages stakeholders to show disputes and discuss in an orderly fashion while the environmental uncertainties are being investigated and better understood.

Adaptive management helps the stakeholders, the decision makers recognize the limits of knowledge and the need to act on imperfect information.

Adaptive management helps to change the decisions made by making it clear that the decisions are provisional and a management's decision need not always be right hence Modifications are expected.

There are two types of Adaptive management Policies –

1) Passive Adaptive Management.
2) Active Adaptive Management.

5.8.2 PASSIVE ADAPTIVE MANAGEMENT

The aim of Adaptive management is to enhance the scientific knowledge and reducing uncertainties.

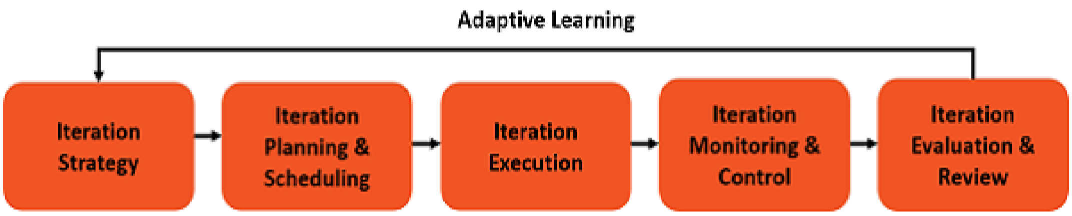


Figure 5.6: Adaptive Learning

From Passive Adaptive management, a preferred course of action, based on existing information and understanding, is selected then outcomes of management actions are monitored, and subsequent decisions are adjusted based on the outcomes.

It gives a perspective of learning and effective management. However, it is limited in its ability to enhance scientific/management capabilities for conditions that go beyond the course of action selected.

5.8.3 ACTIVE ADAPTIVE MANAGEMENT

An Active Adaptive management style reviews the information before management actions.

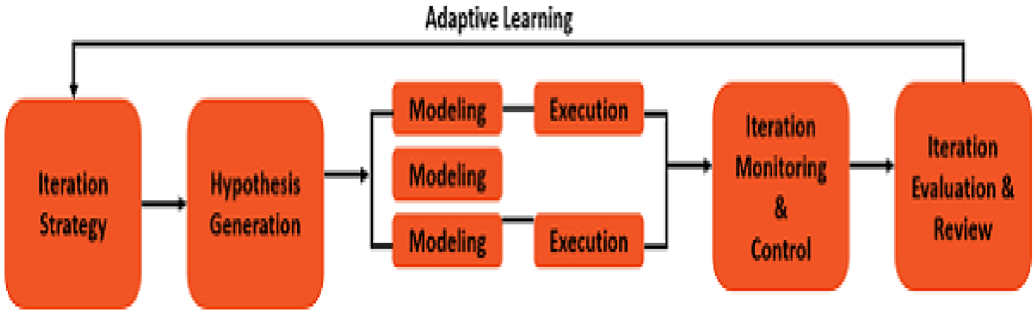


Figure 5.7: Adaptive Learning Iteration

A range of, alternative system and related responses, rather than a single model, is developed. Management options are chosen based on the evaluations of alternative models.

5.8.4 LEADERSHIP-COLLABORATION MANAGEMENT

Adaptive management is a management technique which is best suited for Adaptive Software Development. The approach requires resource/people managers, those can work with people, allow human-interventions, and create an amicable environment for goals to achieve.

In software development, the leaders/managers/directors often take up these responsibilities. Leaders are not merely commanders. Leaders are collaborators and work with the team. Collaborative-Leadership (Working with the team) is the most important factor in Adaptive development.

Leader should have following qualities −

- Understand and set the direction for team.
- Should Influence people involved and provide proper guidance.

- He should Collaborate, facilitate and macro-manage the team. Here we should understand that micromanagement creates negative impact on team and should be avoided.
- Managers Provide direction with creative environments where talented people can be innovative, creative, and make effective decisions.
- Commanding should not be a predominant style but it is occasionally allowed.

Adaptive Testing

5.8.4.1 SPECULATE

A degree of certainty is indicated about the expected results by the plan. The Team Leaders ability of steering the project in innovative directions is restricted by the implicit or explicit goals of the plan conformance.

The term speculate is important in Adaptive Software Development. The reality of the uncertainty in difficult problems is recognized without the planning being abandoned. Exploration and experimentation is encouraged by Speculate. Iterations with short cycles are encouraged.

5.8.4.2 COLLABORATE

Complex applications which are evolved, require a large volume of information for collecting, analyzing and applied to the problem. Automation environments have high degree of information flow and hence complex applications require huge volume of data for collecting, analyzing of the information. This diverse knowledge requirements that can only be handled by team collaboration.

In order to produce accurate results, share knowledge and to make decisions, it is required to work in collaborative way.

A balance between the traditional management techniques and creation and maintenance of the collaborative environment is explained by Collaboration.

5.8.4.3 LEARN

For the ultimate success of the project work, Learning is an important part in the Adaptive (Dynamic Run Time) Development lifecycle. By using defined practices, the team has to enhance the knowledge by:

- Technical Reviews
- Project Retrospectives
- Customer Focus Groups

After each of the iteration, reviews are done. Then assumptions are checked and results of each of the cycle for learning the next direction by the developers and customers. The team learns about the

- Changes in the product and the assumptions of how products are developed then about product changes

5.9 ADAPTIVE AUTOMATION TESTING

Adaptive (Run Time, Dynamic) testing is the counterpart of adaptive control in software testing. It suggests that software testing strategy should be modified on-line by using the testing data collected during software testing as our understanding of the software under test is improved.

Previous studies on adaptive testing involved a simplified Controlled Markov Chain (CMC) model for software testing which employs several unrealistic assumptions. In this paper we propose a new adaptive software testing approach in the context of an improved and namely, general CMC model which aims to eliminate such threats to validity.

A set of more realistic basic assumptions on the software testing process is proposed and several unrealistic assumptions are replaced by less unrealistic assumptions.

A new adaptive testing strategy based on the general CMC is developed and implemented. Mathematical simulations and experiments on real life software are conducted to demonstrate the effectiveness of the new strategy.

Abstract Adaptive (Run Time Dynamic) software testing is the counterpart of adaptive control in software testing. It means that software testing strategy should be adjusted dynamically by using the testing data collected during software testing as our understanding of the software under test improves. Previous studies on adaptive testing rely on (CMC) model for software testing which produces several unrealistic assumptions.

Adaptive software testing approach in the context of an improved model that aims to eliminate threats to validity. Hence some new set of fundamental assumptions on the software testing process is proposed and several unrealistic assumptions are replaced by more common situations in real life software testing. The methodology of an upcoming adaptive testing strategy is also developed and implemented. Thereafter Experimental data collected to demonstrate the effectiveness of the upcoming methodology.

DevOps as a trend is on the rise. Studies show that most of the organizations adopting DevOps continues to grow with every passing year. The core requirements in DevOps world is higher automation. So manual testing moving to automated testing in areas of operations. To this effect, automation teams has to create automation frameworks enabling easier and more effective automated testing to be taken up by one and all on the testing team. However one ongoing flaw in all of this, is disconnect between the test automation and the varied test effort management systems including the test case and defect management systems.

This has been an area of research at QA InfoTech in the recent months to help draw a complete connect in the automation effort E2E and enable automation in the truest essence. This has been taking shape where expectation to see measurable outcomes of around 100% automated effort (A test case when fully automated there is no manual effort involved), a tight coupling between defect management and automation execution, and considerable time savings, this is what we call the adaptive automation testing framework.

5.10 TEST DESIGN PATTERNS WITH RESPECT TO ADAPTIVE AUTOMATION

When we find a specific set of problem and a specific set of corresponding solution then the combined set is defined as a pattern.

With respect to Test Pattern it specifies the way in which any product/service or application is been tested for a specific set of problem with respect to get know set of solution.

Design patterns are created to solve common problems in software design. They are not reserved only for software development but useful for software automation. Yes, there are really sophisticated design patterns used to solve complex issues in software automation. And there are ways easy to understand and adopt design patterns that can significantly improve readability and maintainability of our test automation code.

We will see the design patterns used in Adaptive and Dynamic Test Automation framework.

The same design patterns may be useful in other software activities but we will see how they can be used with Adaptive Test Automation.

5.10.1 DATA PATTERNS

This patterns separate the Data Management from Test Logic, hence logic is clearer and there are no mixes with data. Data is managed separately whether in memory or in Data Base. For example data may be kept in file and may be accessed with Data Provider Module as following.

```
@DataProvider

def adaptiveTestDataProvider():

    With open (“C:/adaptiveTestData.xls”, r+) as f:

        For line in f.readlies (): Print line

    Return
```

5.10.2 TECHNICAL PATTERNS

In this pattern, product technology or environment complexities are been kept separately from the Test Steps being executed. It reduces test complexity and improves test maintainability.

5.10.3 PROXY PATTERNS

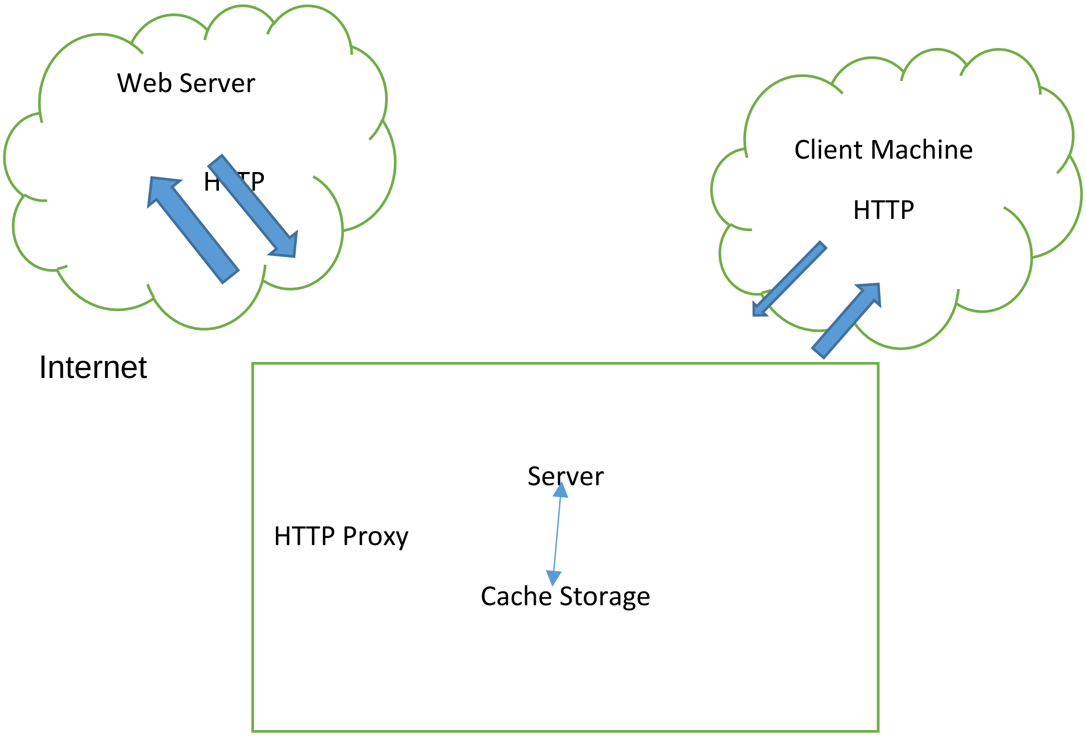


Figure 5.8: Proxy Patterns

Executing Automation against External Vendors via Proxy pattern:

Execution of automation against an external vendor is now possible via a proxy.

This is critical as most of organizations environments control outbound connectivity via a proxy server and running tests against external vendors has been impossible from these environments until now. This post will demo the new Test Framework capabilities and show how to make use of them.

The proxy settings be controlled by environment by default. Adaptive Test Automation Framework recognizes the LOCAL, DevCloud environment, and Productions environments by default and has dedicated property files for each. There are many different environment definitions within organization to define property files. i.e. Default Proxy of cloud environment.

Whether or not a proxy is used is controlled by the presence or absence of two properties. At the moment in STAGE the proxy host & port are the same for all of these vendors but we defined them individually in case that changes in the future. By default ATAF defines proxy host/port in the STAGE property file
Uses of HTTP Proxy:

1) It blacklists external resources
2) Cache Non Functional Resources
3) Collects HTTP Traffic for analysis (Redirects, Loading Time etc.)
4) Speedup Page Loading

5.10.4 Business Patterns

This gives possibility to get actual business requirements and design Adaptive Test Automation accordingly. This makes Dynamic approach more valuable

5.10.5 PAGE OBJECTS PATTERN

In adaptive test automation it allows to create object repository with User Interface elements and these repository is separated from actual Test Automation logic gives freedom of plugging as and when required.

5.10.6 FAÇADE PATTERN

This pattern is useful when we need to make simple interfaces with more complex system/code need to be tested. In adaptive automation strategy as per user and system experience easy to use and maintainable API's need to be designed either externally or internally, hence will give more control dynamically at run time.

5.10.7 FACTORY PATTERN

In adaptive automation scenarios, at run time decision can be taken by the factory for the kind of objects needs to be created for factory as it might not be suitable or we might not know or we are not bothered about the same. Hence specific rules are defined for every factory for creating specific objects.

5.10.8 SINGLETON PATTERN

This pattern is used when in our adaptive Automation Testing we need to deal with exactly one object.

CHAPTER 6

FUNDAMENTAL OF STORAGE CLOUD NETWORKS

CHAPTER 6

FUNDAMENTAL OF STORAGE CLOUD NETWORKS

6.1 INTRODUCTION

A storage device is a hardware that is used in computing either in PC or Server or in Data Center which is used for storing, and extracting information or objects.

This device can hold and store information both temporarily and permanently, and can be internal or external to any computing device (Now a days mobile phones are also a computing devices), called as storage medium or storage media.

6.2 CLOUD COMPUTING DEFINITIONS

6.2.1 SOFTWARE AS A SERVICE (SAAS)

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. Applications can be accessed from various client devices through a thin client interface such as a web browser. The consumer/customer does not manage or control the underlying cloud

Infrastructure including network, servers, storage, or applications, with the possible exception of limited user-specific application configuration settings.

6.2.2 PLATFORM AS A SERVICE (PAAS)

The capability provided to the consumer/customer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider.

The consumer/customer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

6.2.3 INFRASTRUCTURE AS A SERVICE (IAAS)

The capability provided to the consumer/customer is to provision processing, storage, networks, and other fundamental computing resources where the consumer/customer is able to deploy and run arbitrary software, which can include operating systems and applications.

6.3 TYPES OF STORAGE DEVICES

Storage devices are core components of any computing system. They store all the data and applications on a computer, excluding firmware which is directly burnt on the chip.

They are available in different form as following.

RAM, cache, and hard disk, as well as optical disk drives (CD/DVD) and externally connected USB drives.

Hence two different types of storage devices are as following:

- Primary storage: smaller in size, designed to hold data temporarily, internal to the computer. Data access speed is fastest, and include RAM and cache memory.
- Secondary storage: have large storage capacity as compared to primary storage, as well as stores data permanently. They are either internal or external and include the hard disk, optical disk and USB storage.

6.3.1 MAGNETIC STORAGE DEVICES

Today, magnetic storage is one of the most common types of storage, these may include as following.

- Floppy diskette
- Hard drive
- Magnetic strip
- Super Disk
- Tape cassette
- Zip diskette

6.3.2 OPTICAL STORAGE DEVICES

Optical storage uses lasers/lights for reading and writing data. Examples are as following.

- Blu-ray disc
- CD-ROM disc
- CD-R and CD-RW disc
- DVD-R, DVD+R, DVD-RW, and DVD+RW disc

6.3.3 FLASH MEMORY DEVICES

Flash memory devices replacing magnetic media, comparably cheaper, more efficient and reliable technology.

- USB flash drive(Jump/Thumb)
- Memory card
- Memory stick
- SSD

6.3.4 ONLINE AND CLOUD

Storing data online and in cloud storage is becoming popular as data can be accessed from anywhere dynamically as and when required.

- Cloud storage
- Network media

6.3.5 PAPER STORAGE

Early days computers had no method of using any of the above technologies for storing information. Today paper storage are rarely used.

- OMR
- Punch card

6.3.6 Need of storage in computer

A Storage for computers is like a mind or memory to human hence without a storage device, a computer would be considered a dumb device. But in a grid computing number of computers can share common storage, hence no need to separate storage. But to run application on individual machine like browsing, it needs local storage to function properly as cache and temporary file are used and they need to be stored somewhere.

6.3.7 Requirements of so many different storage devices

This is because of advancement of technology and processing power. As and when new applications are being developed. Different version of storage devices with Speed and Space, came into picture.

When new storage devices are manufactured, older are replaced with newer ones.

First punch cards were used in early computers the magnetic media used for floppy disks was not available. Then floppy diskettes had been released, further they were replaced by CD-ROM drives, which were again replaced by DVD drives, and DVD been replaced by flash drives. Today, smartphones are available that have hundreds of times the capacity at a much smaller price and much portable to carry. Each advancement of storage technology gives computing device the ability to store data, save data faster, and access the saved data faster.

6.3.8 Definition of storage location

Storage location may be fixed or removable or may be over the network, its choice of the user to move the data to any of secondary storage, i.e. removable storage device such as a flash drive but default is Hard Drive.

6.3.9 USES OF STORAGE DEVICES IN PRESENT SCENARIOS

Most of the storage devices are no longer used. In present primarily hard disk drive or SSD are used to store data in computing devices. Information and have the options for USB flash drives and access to cloud storage. Desktop computers are equipped with disc drives, capable of reading CDs/DVD's those can write CD-R and other recordable discs.

6.3.10 STORAGE DEVICE WITH LARGER CAPACITY

Storage devices are available in different capacities from Mega Bytes to Tera Bytes. Networked computers may also have access to even larger storage with large tape drives, Bunch of Disks using cloud computing, or NAS devices.
Below is a list of storage devices, smallest capacity to the largest capacity.

1. Punch card
2. Floppy diskette
3. Zip disk
4. CD
5. DVD
6. Blu-ray disc
7. Flash jump drive
8. Hard drive / SSD
9. Tape drive
10. NAS / Cloud Storage

6.3.11 FILES ACCESS ON STORAGE DEVICES

Storage device management depends on the operating system feature called as files system. Microsoft Windows, uses a file manager to access the files stored on storage device/s. called as Explorer. Apple computers uses, Finder as default file manager.

6.3.12 COMMON PROBLEMS ENCOUNTERED WITH STORAGE DEVICES

1. **Hardware failure.** Most of the users are being affected by hardware failures. Hence regular maintenance and appropriate handling is required to prolong the durability of storage devices.

2. **Data Loss.** File deletion makes to lose precious data. To handle the data loss, recovery programs are designed to provide a solution for lost files, deleted data, corrupt documents and hidden files. If data loss is accidental then data recovery software may retrieve back 80% of the lost data.

6.4 FUNDAMENTALS OF STORAGE NETWORKS

Storage Networks is a collection of servers, networked together with other storage devices using storage networking protocols.

Mainly following manufactures are proving Storage and Networking Solutions, Brocade, Qlogic, Emulex, Cisco, Dell, EMC, Hewlett Packard, IBM, Lenovo, Microsoft, NetApp, Oracle and VMware.

Now let’s see how storage devices are organized.

6.4.1 DIRECT ATTACHED STORAGE

A direct-attached storage device is not networked. Direct-attached storage (DAS) is connected to single computing system and not accessible to other computers. HDD and SSD is the usual form of direct-attached storage. Optical devices and tape are also example of DAS. In the enterprise, individual disk drives (bunch of disks) in a server are called *direct-attached storage*, or groups of drives (disk enclosures) that are external to the server directly connected to an Interface(PCI) card plugged into the internal bus of a server but are directly attached through technologies as

1) Small Computer System Interface (SCSI),
2) Serial Advanced Technology Attachment (SATA),
3) Serial-Attached SCSI (SAS),
4) Fibre Channel (FC) or iSCSI.

Are also called as Direct Attached Storage.

6.4.1.1 DIRECT-ATTACHED STORAGE PROS AND CONS

DAS gives better performance than networked storage because the server does not have to traverse a network (Connected Switches). Hence to get high performance DAS is used for some Data Intensive applications.

I.e. Microsoft Exchange.

Direct-attached storage can't be easily shared and it does not facilitate failover in the case of server crash hence being criticized. But with virtualization DAS overcome above mentioned shortcomings and gained in popularity. In centralized and networked shared storage, such as NAS or SAN storage capacity is shared among servers over a dedicated network connection. But DAS is dedicated to a single server. Hence connectivity and expandability are constrained by the number of expansion slots.

The size of the DAS enclosure restricts storage capacity. Sharing with DAS is typically limited to a small number of ports or host connections. DAS is less expensive than SAN or NAS, DAS is easier to deploy when directly plugged into a server. This made DAS a practical storage choice for many small and medium-sized businesses (SMBs), where storage costs are a major factor.

Physical servers continue to boot from DAS storage. SSDs, makes booting locally more effective than a SAN. i.e., through local SSDs, physical server can be booted in few seconds. Direct-attached storage does not offer remote replication and snapshots.

6.4.1.2 DISCOVER AND FIX SECURITY ISSUES IN DAS SYSTEMS

Security may be a concern where storage systems that host an organization's critical data and applications.

Below mentioned steps should be taken to protect data :

- Perform thorough security assessment to discover problems. Following security vulnerabilities may be discovered, such as user permissions, missing patches or misconfigured systems.
- Do audit of user access permissions, and scan the DAS for any information accessible to every person on the network then restrict unauthorized permissions, segment network to protect critical DAS systems.
- Operating system and application software patches should be continuously updated otherwise it can leave DAS vulnerable hence an intruder can gains unwanted access to the server connected to the DAS can access all the data on the system, breached.
- Fault tolerance testing should be done to check the resiliency of the DAS system.

6.4.1.3 FUTURE OUTLOOK AND TRENDS OF DIRECT ATTACHED STORAGE

The SATA Express (SATAe) interface arrived in 2014, as a connecting technology for DAS. It enables DAS to benefit from the reduced latency overhead of more than 50 % of the nonvolatile memory express (NVMe) specification. NVMe, serve as the logical device interface for direct-attached Peripheral Component Interconnect Express (PCIe) storage devices, is architected for higher-performance SSDs.

6.4.2 NETWORK ATTACHED STORAGE

Network-attached storage (NAS) is enables heterogeneous client devices to retrieve data from centralized storage. Standard Ethernet connection can be used to access shared file storage in LAN. NAS devices are managed with a utility based on browser. Each NAS resides on the LAN as an independent network node, defined by unique Internet Protocol (IP) address.

NAS Characterize is ease of access, high capacity and fairly low cost. NAS devices provide infrastructure to consolidate storage in one place to support tasks, such as archiving and backup, and cloud access.

6.4.2.1 USES OF NETWORK-ATTACHED STORAGE

NAS enables users to collaborate and share data more effectively, particularly teams that are remotely located or in different time zones.

By using wireless router connection with NAS, it is easy for distributed work environments to access files from any device connected to the network. A NAS environment is deployed as the foundation for a personal or private cloud.

NAS products are designed for use in enterprises, for home offices or small businesses. Devices usually contain at least two drive bays. Enterprise NAS is designed with more high-end data features to aid storage and usually comes with multiple drive bays.

Enterprises had to configure and manage hundreds or even thousands of discrete file servers before NAS. Thereafter to expand storage capacity, NAS appliances are fitted with more or larger disks known as *scale-up NAS*.

Also NAS vendors using services of cloud storage providers to get flexibility of redundant backup.

Network-attached storage are using hard disk drives (HDDs) to store data. If multiple users simultaneously uses Input/output (I/O) then error may occur to overcome this NAS systems use faster flash storage.

6.4.2.2 NAS USE CASES

The requirements and configuration of HDD selected for a NAS is specified by the applications to be handled. Any routine task shared between coworkers and performing periodic data backup is another use case.

Using a NAS to handle large volumes of streaming media files requires larger size disks, extra size of memory and powerful network processing systems.

NAS system is used in home office environment to process multimedia files or to automate backups. NAS is also used to manage storage for smart TVs, home security systems and other consumer-based internet of things (IoT) devices (Freezer, Washing Machine etc.).

NAS array is used in Enterprises to backup target for archiving and disaster recovery. By running NAS device in server mode server mode, it also serve email, multimedia files, database requests or printing jobs as and when required or automated.

NAS products also provides enough disks to support RAID (redundant array of independent disks), it turns multiple hard disks into one logical unit to speedup performance, high availability (HA) and redundancy.

6.4.2.3 NAS PRODUCT CATEGORIES

NAS devices are categorized in three broad categories based on capacity and scalability.

High-end: This is driven by enterprises that need to store large quantities of file data, including virtual machine images. This also provides rapid access with clustering capabilities. The clustering concept is invented to address drawbacks associated with traditional NAS. This uses Distributed File System.

A DFS runs concurrently on multiple NAS devices connected together to provide access to all files in the cluster, irrespective of the physical node.

Mid-end: The NAS midmarket needs several hundred terabytes (TB) of data. They cannot be clustered, however, which multiple NAS devices can lead to file system siloes as and when required.

Low-end: The low end of the market is aimed at small enterprises and home users that require local shared storage. But now this market is shifting toward a cloud NAS model, by products such as Soft NAS Cloud, Virtual NAS and software-defined storage (SDS) from legacy storage providers.

6.4.2.4 FUTURE OF NETWORK-ATTACHED STORAGE

The functionality of NAS devices extended to support virtualization. High-end NAS products also support flash storage, multiprotocol access and replication. NAS devices run a standard OS, such as Microsoft Windows, while others may also use proprietary OS.

IP is the data transport protocol, Mid-end NAS products may support Network File System (NFS), Internetwork Packet Exchange (IPX), NetBIOS Extended User Interface (NetBEUI), Server Message Block (SMB) or Common Internet File System (CIFS).High-end NAS products support Gigabit Ethernet (GigE) for faster data transfer. In a network-attached storage deployment, the NAS head performs the NAS control functions, provides access to back-end storage through an internet connectivity. This configuration is known as *scale-up NAS architecture*. With scale-out NAS, the administrator installs larger heads and extra hard disks to increase storage capacity. Scaling out provides the scope with an organization's business needs. Enterprise NAS systems can store and process billions of files without the performance trade-off of doing metadata searches.

6.4.2.5 SCALE-OUT NAS AND OBJECT STORAGE

Object storage is designed as an alternative for handling unstructured/dispersed data. Both storage methodologies deal with scale, only in different style/implementation.

In NAS, files are centrally managed via the Portable Operating System Interface (POSIX), which ensures that multiple applications can share a scale-out device without fear simultaneous access of files and provide data safety.

Object storage used another way to work with scalable storage in online run time dynamic web-scale environments. It often works with unstructured data that is not easily compressible, i.e. large video files and Streamed Data. Object storage does not use any file system. Objects are stored in a flat address space. Metadata are added to describe every object, and enables quick identification within a flat address namespace.

6.4.2.6 NAS AND SAN COMPARISON

The similarities and difference between SAN and NAS is as following.

NAS handles unstructured data, SAN handle block storage as *structured data* and organizes storage resources on an independent, high-performance network.

NAS handles I/O requests for individual files, whereas a SAN manages I/O requests for contiguous blocks of data. While NAS traffic moves across TCP/IP, such as Ethernet,

SAN network traffic route over the FC/iSCSI protocol designed specifically for storage networks. A NAS can be a single device, but SAN provides full block-level access to a server's disk volumes.

6.4.2.7 SAN/NAS CONVERGENCE

Technological barriers kept the file and block storage worlds separate, each in its own management domain and each with their strengths and weaknesses.

With the emergence of unified storage, there was a need to improve large-scale file storage with SAN/NAS convergence. This keeps block- and file-based data on one storage array.

Convergence supports (SAN) block I/O and (NAS) file I/O with same scope of switches. The concept of hyper-convergence is pioneered by Nutanix and SimpliVity Corp. (now part of HPE). Hyper-converged infrastructure (HCI) consolidates the computing, network, and virtualization resources on a single appliance. HCI systems pool tiers of different storage media and connects it to a hypervisor as a NAS mount point, even though the underlying shared resource is block-based storage. Because HCI provides only the most basic file services, hence a data center may still need to implement a separate network with attached file storage.

Converged infrastructure (CI) includes servers, networking, and storage and virtualization resources on sets of hardware pre-validated by the vendor.

Hyper Converged Infrastructure, consolidates devices in one chassis, but CI consists of separate devices. This gives excellent flexibility in building storage architecture. Organizations storage management now opting for CI and HCI systems to replace a NAS or SAN environment.

6.4.2.8 CLOUD-BASED FILE STORAGE

In addition to NAS devices, some data centers replace physical NAS with cloud-based file storage. Amazon Elastic File System is the scalable storage in Amazon Elastic Compute Cloud (EC2). And Microsoft Azure File manages file shares based on SMB and CIFS for access by local and cloud-based infrastructure.

NAS gateways formerly enabled files to access externally attached storage namely, connecting to a high-performance area network over FC or JBOD (just a bunch of disk) in attached servers. NAS gateways are still in use but less frequently compared to a cloud storage gateway, object storage or scale-out NAS.

A cloud gateway resides in company's data center network, connecting applications between local storage and the public cloud. For example Nasuni Corp. created the cloud-native UniFS file system software, which is bundled on Dell PowerEdge servers or available as a virtual storage appliance (VSA) for use.

6.5 FUNDAMENTALS OF STORAGE PROTOCOLS

For cloud initiatives, Storage is a major consideration; in terms of cost, performance, throughput, disk, vendor, and most importantly protocol.

Let’s check for the major storage protocols at play in the data center

6.5.1 SMALL COMPUTER SYSTEM INTERFACE (SCSI)

SCSI is the block level access method from storage disk. Blocks are smallest unit that can be read or written to on a storage disk, they exist in different sizes depending on disk type and usage. Block level access means that the server can directly access the disk blocks without the need for a file system, this is opposite of file-based storage discussed later.

SCSI has been used to move data within a single computing system. Here operating system handles data reading/writing using the SCSI protocol to a SCSI drive controller, managed by one or more devices on a SCSI cable within a system chassis. SCSI controller ensured that only one device would be active on the cable at any time which prevents data collision and corruption on the SCSI bus.

As SCSI was managed by a single controller and contained within a system, chance for data loss, are minimal, this meant that SCSI did not require control mechanisms to handle data loss or contention. SCSI is used in its native format but it has also been encapsulated into other protocols for use within storage networks for consolidated storage.

6.5.2 FIBRE CHANNEL (FC)

Fibre Channel invented to extend the functionality of SCSI with point-to-point, loop, and switched topologies. FC allows longer distances transmission compared to SCSI and storage consolidation. This encapsulates SCSI data and Command Descriptor Blocks (CDB) into the payload of FC frames. FC networks provided the addressing, routing, and flow-control required to support SCSI data.

Additionally Fibre Channel networks are providing ‘lossless’ in order of delivery with SCSI. This ensures that in a stable network FC frames will not be dropped, and delivered successfully ensuring that the Upper Layer Protocols (ULP) will not require to reorder/resend frames.

Fibre Channel networks are carried over fiber-optic links on dedicated infrastructures/switches. These infrastructures are traditionally built-in pairs as exact mirrors of each other. This provides complete physical redundancy and high bandwidth and low-latency. Fibre Channel N/W devices come in 1/2/4/8 Gbps speeds with 16/32 Gbps in the works. Also 10Gbps FC links are typically available on a proprietary cards for links between switches.

6.5.3 COMMON INTERNET FILE SYSTEM (CIFS)

CIFS is a shared storage protocol typically used in Microsoft environments for file sharing. This is a file based storage system based on Small Message block (SMB). Windows-based file shares uses CIFS as the transfer protocol of the file level data. File based storage depends on an underlying files system such as FAT32, XFS, NTFS. File level storage is an excellent medium for some applications but may not be effective with some other applications.
Hence an application needs direct block access to disk file based storage is not appropriate. This includes category include databases and most Operating Systems.

6.5.4 NETWORK FILE SYSTEM (NFS)

NFS is file based storage protocol used in Linux and UNIX environments.

This is also used in VMware environments and can offer several benefits for virtual machine storage. As a file based storage protocol NFS facing same limitations mentioned for CIFS.

6.5.5 HYPER TEXT TRANSFER PROTOCOL (HTTP) AND OTHERS

When the data packets leaves the data center (private/internal cloud) and moves up to the service provider level such as Google, Amazon, need to be traversed over HTTP . When thousands of nodes needs to be supported with multiple Terabytes each, traditional storage protocols may not suffice, it needs to take care of scalability of the systems and the administration of the disk.
iSCSI and FC both require management for the RAID, volumes, and LUNs. File Based Storage Protocols (CIFS, NFS) require a fair amount for the security and volumes. HTTP protocols based storage are being used to simplify storage configuration and increase its scalability. Each protocol has its uses, benefits and drawbacks. Most environments can benefit multiple or any required protocols. Cloud storage will need every protocol as a key.

6.6 FUNDAMENTALS OF STORAGE NETWORKING PROTOCOLS

6.6.1 INTERNET/IP SMALL COMPUTER SYSTEM INTERFACE (ISCSI)

Internet SCSI (iSCSI) encapsulate SCSI data and CDBs into payload of IP packets therefore SCSI protocol to can be extended across existing IP infrastructures.
As IP is routable within the data center and across the WAN, iSCSI is not traditionally used/supported over routed boundaries. Hence with iSCSI,

storage data can be extended across the existing infrastructure with some additional cost.

Because of flaws in the protocol and limitations of the traditional Ethernet based data center networks, iSCSI has not gained the market share, as predicted.
1GE links are already saturated hence 10 Gigabit Ethernet been made standard.
Hence implementing iSCSI required additional switching infrastructure. 10GE has increased bandwidth limits but still iSCSI is not been catapulted the mainstream.
There are several reasons for this,

1) There is large existing investment in Fibre Channel, and
2) Limitation with iSCSI protocols.
    a. SCSI protocol expects lossless, in-order delivery, and places it in TCP/IP packets designed to support heterogeneous networks and
    b. ISCSI experience packet loss and out-of-order delivery frequently. There is no additional tools to either SCSI or TCP/IP for handling the SCSI payloads.
    c. ISCSI is unusable or should be written off it just means that additional considerations must be made when designing iSCSI, especially in the Enterprise or larger environment.

In order to provide proper performance for iSCSI on shared networks following needs to be considered
1) Quality of Service (QoS),
2) physical architecture, and
3) Jumbo frame support must be taken into account.

Hence many iSCSI networks been placed on separate network hardware from the data center LAN (isolated iSCSI networks.) This has minimized some of the benefits of consolidating on a single protocol. With 10 Gigabit Ethernet and the standardization of Data Center Bridging (DCB) iSCSI will expand in future.

6.6.2 FIBRE CHANNEL OVER ETHERNET (FCOE)

Fibre Channel over Ethernet (FCoE) protocol suite standard ratified in 2009.
It provides the functionality for moving native Fibre Channel across consolidated Ethernet networks. And relies on the DCB standards. FCoE encapsulates Fibre Channel frames inside Ethernet Jumbo Frame payloads hence utilizing jumbo frames ensure that the FC frame is not fragmented or changed by any means.

The FCoE (With DCB) standards provide a tool set for consolidating existing Fibre Channel workloads on shared 10GE networks and simultaneously provides lossless, in-order delivery SCSI packets.

FCoE does not modify the Fibre Channel protocol suite and allows for the existing management model including zoning, LUN masking, etc.
FCoE has started gaining ground over the past several years pushed by large hardware vendors in the storage, network, and server markets.
FCoE provides tools for encapsulation of FC in 10 Gigabit Ethernet frames.

The purpose of FCoE is to allow

1) Consolidation of low-latency,
2) High performance FC networks onto 10GE infrastructures.
3) This allows for a single network/cable infrastructure which greatly reduces switch and cable count, lowering the power, cooling, and administrative requirements for server I/O.

FCoE is designed to be fully interoperable with current FC networks and require little training for storage and IP administrators.
FCoE operates by encapsulating native FC into Ethernet frames.

Native FC is a 'lossless' protocol, so frames doesn't drops during periods of congestion/collision. Hence by design in order to ensure the behavior expected by the SCSI payloads.

Traditional Ethernet does not provide the tools for lossless delivery on shared networks so enhancements were defined by the IEEE to provide appropriate transport of encapsulated Fibre Channel on Ethernet networks.
Ethernet enhancements are backward compatible with traditional Ethernet devices, meaning DCB capable devices can exchange standard Ethernet frames seamlessly with legacy devices.

The full FC frame is encapsulated in an Ethernet jumbo frame and avoids any modification/fragmentation of the FC frame.

This mapping between Ethernet and FC is done through a Logical End-Point (LEP) which is a translator between the two protocols and is responsible for providing the appropriate encoding and physical access for frames traverse from FC nodes to Ethernet nodes and/or vice versa.

Following devices that act as FCoE LEPs:

1) Fibre Channel Forwarders (FCF) are switches capable of both Ethernet and Fibre Channel, and
2) Converged Network Adapters (CNA) provide the server-side connection for an FCoE network.

Additionally the LEP operation can be done using a software initiator and traditional 10GE NICs but this places extra workload on the server processor rather than offloading it to adapter hardware.

Advantage:

1) One of the major advantages of replacing FC layers when mapping onto 10GE is the encoding overhead. This dramatically reducing the protocol overhead and increasing throughput.

2) The second major advantage is that FCoE maintains FC layers which allows seamless integration with existing FC devices and maintains the Fibre Channel tool set such as zoning, LUN masking etc.

FCoE relies on another standard set known as Fibre Channel initialization Protocol (FIP) in order to provide FC login capabilities, multi-hop FCoE networks, and FC zoning enforcement on 10GE networks.

Hence FCoE is a protocol to choose from when designing converged networks, or switched architectures.

6.7 STORAGE VIRTUALIZATION

Storage virtualization is the way of pooling physical storage from multiple storage devices into what appears to be a single storage device. It's a pool of available storage capacity managed from a central console. Here the management software identifies available storage capacity from physical devices and to then aggregate that capacity as a pool of storage that can be used in a virtual environment by virtual machines (VMs) or underlying operating systems. The virtual storage software intercepts I/O requests from physical or virtual machines and sends those requests to the appropriate physical location of the storage devices that is a part of the pool of storage in the virtualized environment. Here Virtual storage appears like a standard read or write to a physical drive. A RAID array can also be considered a type of storage virtualization. Multiple disks array are presented to the user as a single storage device and in the background, and it replicates data to multiple disks in case of a single disk failure.

6.7.1 TYPES OF STORAGE VIRTUALIZATION

There are two basic methods of virtualizing storage:

1) File-based: File-based storage virtualization is a specific use case, applied to network-attached storage (NAS) systems. Using the SMB or NFS protocols, file-based storage virtualization breaks the dependency between the data being accessed and the location of physical memory. This enables the NAS system to handle file migration in the background to improve performance.

2) Block-based or block access virtual storage is widely applied in virtual storage systems than file-based storage virtualization. Block-based systems separates the logical storage (i.e. drive partition), from the real physical memory blocks in a storage device, (HDD or SSD). Virtualization management software collect the capacity of the available blocks of memory space and pool them into a shared resource to be assigned to any number of VMs, bare-metal servers or containers.

Note: To access that data in the physical storage devices, the virtualization software needs either to create a map using metadata or, sometime use an algorithm to dynamically locate the data at run time.

6.7.2 EXAMPLE OF STORAGE VIRTUALIZATION

1) Block-based virtualization was IBM's SAN Volume Controller (SVC), called IBM Spectrum Virtualize. The software runs on storage array and creates a single pool of storage by virtualizing logical unit numbers (LUNs) those are attached to servers and connected to storage controllers. This also enables customers to tier block data to public cloud storage.

2) Another storage virtualization product is Hitachi Data Systems' also known as Hitachi Virtual Storage Platform (VSP). Hitachi's array-based storage virtualization which enabled users to create a single pool of storage across separate arrays, even those from other leading storage vendors.

6.7.3 VIRTUALIZATION METHODS

Storage virtualization is a way to gather and manage storage capacity that is accumulated/collected from multiple physical storage devices and then made available and reallocated in a virtualized environment. Modern IT technologies, such as hyper-converged infrastructure (HCI), takes advantage of virtual storage, with virtual compute power and virtual network capacity.

Below are the ways storage can be configured to a virtualized environment:

HOST-BASED STORAGE VIRTUALIZATION

In this case, the host, or a hyper-converged system made up of multiple hosts, presents virtual drives of a fixed capacity to the guest machines, whether they are VMs in an enterprise environment connecting to cloud storage. Virtualization and management done at the host level via management software, and the physical storage can be any device or disk array. It's mostly seen in HCI systems and cloud storage

ARRAY-BASED STORAGE VIRTUALIZATION

refers to the method in which a storage array presents different types of physical storage for use as storage tiers. How much of a storage tier is made up of solid-state drives (SSDs) or HDDs is handled by management software in the array and is hidden at the guest machine or user level.

NETWORK-BASED STORAGE VIRTUALIZATION

commonly used in enterprises today. A network device, such as a smart switch, connects to all storage devices in a Fibre Channel (FC) storage area network (SAN) and presents the storage as a virtual pool wherever needed for easy management.

Storage virtualization separates the actual complexity of a storage system, such as a SAN, which helps a storage administrator perform the tasks of backup, archiving and recovery effectively in less time.

6.7.4 IN HOUSE CLOUD NETWORK (DEVELOPMENT CLOUD) APPLICATIONS AND LIMITATIONS

- Misconceptions About Development Cloud
- Cloud Stuff
- Cloud 101
- On-premise/internal vs Off-premise/external
- vCloud Director
- Workloads
- Environments and Networks
- Development Cloud is a New Bubble
- Relationship to Other Environments
- Stage and Production Clouds
- Promotion to Higher Environments
- Experience Gained via Development Cloud
- Self-service and SLAs
- Security and Access
- Support and Role of Administrative Groups
- No Special Snowflakes
- Hosting Shared Services in Development Cloud
- Model-driven Automation

6.8 FUNDAMENTAL OF CLOUD COMPUTING

Basics of Cloud Storage

Cloud Storage includes the concepts of data center are the houses which facilitates cloud storage systems and related servers, systems and services by means of underlying hardware, Storage Protocols and Storage networking.

6.8.1 CHARACTERISTICS OF CLOUD STORAGE:

Important characteristics of cloud storage is to dynamically interface with other cloud services, which includes

SaaS (Software as a Service),

PaaS (Platform as a Service),
IaaS (Infrastructure as a Service),

IaaC (Infrastructure as a Code) and

BPaaS (Business Process as a Service).

6.8.2 WHAT IS CLOUD COMPUTING?

6.8.2.1 DIFFERENCE BETWEEN A PUBLIC CLOUD, A PRIVATE CLOUD AND A HYBRID CLOUD?

A private cloud infrastructure is provisioned for exclusive use of Organization. It is owned, managed, and operated by Organization, a third party, or combination of them, and it may exist on or off premises. Public cloud infrastructure is deployed for open use by the public Infrastructure which exists on the premises of the cloud provider. A Hybrid cloud is the composition of two or more distinct cloud infrastructures (private or public) bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

6.8.2.2 DEFINITION OF SAAS

Software as a Service. SaaS are applications that are developed and hosted by the SaaS vendor which the end user accesses over the Internet.

Unlike traditional packaged applications the SaaS vendor owns the software and runs it on computers in its data center.

6.8.2.3 DEFINITION OF PAAS

Platform as a Service. Although less established and not as widely available as compared to SaaS and IaaS offerings, ready-to-use application platforms offer great promise for organizations that aren’t compelled to own and manage the underlying infrastructure

6.8.2.4 DEFINITION OF IAAS

Infrastructure as a Service. The virtualized processing, storage, and networking services along with automation and management capabilities in this area offer the most flexible level of services in the cloud computing model.

6.8.2.5 DEFINITION OF DAAS

Database as a Service. A Physical Data Management strategy for managing an on-site private cloud made up of several different database architectures. These architectures are capable of providing varying degrees of database service to an application based on business requirement’s related to availability, scalability and performance. The main objective of the DaaS strategy is to provide a high quality database service while maximizing ROI on database software/hardware and supporting agile development efforts through faster service delivery.

6.8.2.6 MAJOR PUBLIC CLOUD PROVIDERS

There are a number of established cloud providers (Amazon/AWS, Microsoft/Azure, and Rackspace) and new entries every day. Many traditional hosting companies (Savvis, Affiliated Computer Services) are rebranding there services or delivering new cloud computing offerings.

6.8.2.7 "SERVICE" WITH RESPECT TO CLOUD

A service is something that provides “value” to the Organization lines of business. Infrastructure and Platform services are not used directly by the business, however they are required to provide a business service (i.e. email, HR application to track benefits, etc.). Infrastructure services include storage, compute, network, backup & recovery, etc.

6.8.3 WORKING OF CLOUD STORAGE

Cloud storage works with the help of cloud Storage Access Protocols

6.8.3.1 Cloud storage access methods

1) **Web services application programming interfaces (APIs):** are RESTful APIs (according to the principals of Representational State Transfer) to integrate with other related applications.

2) **File-based protocols:** are used to transfer files and provide integration independent of the application being connected. They also provide a faster integration than web service APIs. Those includes

    a. Network File System (NFS),

    b. Common Internet File System (CIFS)

    c. File Transfer Protocol (FTP)

3) **Block-based APIs:** use iSCSI to connect a front end to storage middleware that supports services like data replication and data reduction.

4) **Web-based Distribution Authoring and Versioning (WebDAV):** based on Hypertext Transfer Protocol (HTTP).

6.8.4 ADVANTAGE OF CLOUD STORAGE:

1. Accessibility: Data/Files/Applications can be retrieved/accessed/managed from anywhere without any constraints as just needs Internet/Network Connectivity.

2. Have choice of not carrying Physical Storage Devices or Processing Devices/computers and applications as all can be available over cloud to use dynamically.
3. Cloud storage grows as per demand, hence storage allocation may be chosen as and when required.
4. Users are allowed to access data and cloud environment includes platforms and application to work on a project as a collaborative effort.
5. Data and Business Process, Infrastructure, Platform and Software services can be shared.
6. Usability: Data can be moved between Cloud storage and Local Storage.
7. Bandwidth Sharing: Web Link of Data stored at cloud may be shared to recipients.
8. Data Recovery: In case of emergency cloud storage may be used as backup plan as files can be accessed through network as and when required.
9. Reduction in Operational Cost as no need of internal data management, power and resources.
10. Synchronization: Cloud service effect the any committed changes to affiliated devices.
11. Metered Services: It's an example of elasticity and adaptability where cloud model works on the principal of Pay as per use i.e. Storage and bandwidth, it's a case of as users never pay for more resources than they need and used.
12. Availability: It's always on and available for use.
13. Data Security: As soon as data is being transferred and stored at cloud, extra layer of security protocols and encryption (RSA, 3DES etc.) is added.
14. Customization: ability to customize cloud with other applications. For this API Integration capability with cloud can give lot of benefit including verification etc.

15. Automation Infrastructure development can be done with closed as per need basis, work can be accomplished from anywhere and anytime.

6.8.5 MISCONCEPTIONS ABOUT DEVELOPMENT CLOUD (IN HOUSE)

Much of it is great feedback and spot on, but given that the service is so new and provides capabilities that organization has not traditionally provided, it's worth highlighting the top misconceptions about the service.

6.8.5.1 IT'S FREE

Development Cloud is never free, although there is a free tier of usage, more than modest consumption may require project to contribute funds to expanding the shared resource pools. It's chargeback model where you pay for only what you use.

6.8.5.2 CLOUD STUFF

How is Development Cloud an example of a true "cloud"? Isn't "cloud" just a buzzword?

Following services are provided by DEVELOPMENT Cloud.

Infrastructure as a Service (IaaS)

a key element of the Development Cloud implementation is the integration of an IaaS or cloud platform. Although it is still early days in this market, there are some off-the-shelf products that are stable and feature rich enough to support the Development Cloud service. Every large systems company, systems management vendor and virtualization vendor has offerings in this area. Some offerings are evolutions of and extensions to traditional systems management tools while others are green field cloud oriented products.

What are some of the important capabilities provided by an IaaS and cloud platform solution?

- Catalogue of services

- Users can easily browse and select deployable stacks or images of interest.
- Self-service web interface for selecting and managing standard services
- API for automation
- Transparency of resource consumption
- Initially, some degree of "show back" or the ability to inform users of the costs of the services they are consuming
- Multi-tenant, heavily virtualized compute, storage and network
- Decoupled capacity management

Platform as a Services (PaaS)

Although the market for on premise PaaS solutions is less mature than on premise IaaS solutions, But it’s been intend to strive to deliver a PaaS-like experience for the web container and DB services.

It won't achieve the level of productization and packaging expecting from vendor and provider PaaS offerings, but will begin to offer PaaS-like offerings of our standard application and database platforms in the spirit of Google App Engine, AWS Elastic BeanStalk, AWS RDS, etc. One of the attractive aspects of off-premise cloud offerings is the ready availability of such PaaS services.

Immature market and risk management

to apply much of these capabilities in off-premise cloud environments and that the market is relatively immature with an absence of standard interfaces apart from defacto standards such as the AWS APIs, to bias our technology and tools selections toward those offerings that are more modular and adaptable. It’s also expected that to swap out tools over time and even support perhaps two IaaS deployment platforms over time.

6.8.5.3 ON-PREMISE/INTERNAL VS OFF-PREMISE/EXTERNAL

Why starting with an on premise, internal build out?

Can't ignore on premise production value of cloud

It is clear that garnering some of the benefits of a true cloud platform (see Experience gained section below) will be important to our higher level environments.

Therefore, the experience gained with an on premise Development Cloud will help us prepare to realize higher level Cloud environments.

Off-premise is interesting

at the same time, the establishment of on premise Development Cloud as providing great experience on multiple fronts that will help us adopt off-premise cloud environments for certain use cases, workloads and data. The costs associated with the on premise Development Cloud will help establish more of an apples-to-apples comparison

6.8.5.4 VCLOUD DIRECTOR

The initial form of the Development Cloud service uses VMware's vCloud Director Product as the Infrastructure as a Service (IaaS) layer. vCloud Director or "vCD" is similar to a variety of emerging IaaS products and platforms that are positioned to enable both enterprises and service providers to offer IaaS-based services. These products and tools are broadly characterized as "cloud platforms" with capabilities that overlap with an emerging set of "cloud management" tools.

Other examples in this space include Citrix's Cloud Stack, Eucalyptus and a variety of relatively nascent OpenStack-based offering such as Nebula. The most popular off-premise or public cloud form of IaaS is Amazon's EC2 service followed by Rackspace's offerings.

6.8.5.5 WORKLOADS

What workloads, app types and infrastructure platforms are supported?

Workloads, app types: Common web application and web service development, CI-based builds and automated regression tests, new/modified stack prototyping and development, infrastructure automation development are all within scope.

X86/x64-based: Along the lines of both general industry trends and practical vendor and provider investments in IaaS platforms and tools, focusing on support of x86/x64-based OS platforms and workloads that are compatible with those platforms.

Today, this implies support for Red Hat Enterprise Linux (RHEL) and Windows Server. In the Prototyping with Modified Stacks and Platforms use case, on an exception basis, other x86-compatible OSs may be experimented with, but RHEL and Windows Server are the platforms on which the standard selectable services will be offered.

Other Platforms: Although Development Cloud will not be focusing on other platform architectures, but will continue to enhance ability to manage deployments targeting other platforms.

Carrot: The targeted platform mix aligns with our overall direction to move more of the web-based infrastructure to a RHEL / JBoss platform vs. a Win / AIX / WebSphere environments. Utilization of the Development Cloud services can help projects accomplish these goals.

Incompatible workloads: Given that some workloads won't be compatible with the Development Cloud environment even over a longer period of time, and some capacity dedicated to supporting these workload deployments may be required for years. For example, packaged and custom applications that require a specific platform and/or stack that is not supported by in House Cloud may still require hosting in traditional DTL environment. Hardware appliance based services are similar in that they would not be hosted on Development Cloud itself, but may be positioned on the periphery or, as an alternative, virtual forms of those appliances may be deployed on Development Cloud for development purposes.

6.8.5.6 IN HOUSE CLOUD IS A NEW BUBBLE

Why to need yet another environment? Can't it just evolve how to do things in traditional in House Cloud Environment?

6.8.6 CONVERTING EXISTING APPLICATION PLATFORM AND INFRASTRUCTURE IN CLOUD

NOT A COMPLETELY DISTINCT OR DUPLICATE ENVIRONMENT

Existing application platform and infrastructure services will be reused in support of Development Cloud.

SETTING THE STAGE

By establishing a new environment, bubble, context, etc., and giving it a name, and set a clear message to all involved that this is not "Your father's Olds mobile". i.e. this is a clear, transformational effort to take a different, but informed tack to delivering development-oriented platform services.

BY DEFAULT, NOT WEIGHED DOWN

Along the lines of setting the "bit" as to this environment being different, it’s consciously avoiding the need to reuse and/or adapt newer implementation approaches to existing implementation methods.

Many existing policies will continue to apply, but many implementations will change over time to support our goals. For example, it’s intend to use best practices for positioning and allocation of hardware resources to best support a true elastic cloud environment.

TRUE IAAS PLATFORM AND MANAGEMENT TOOLS

A true API-driven IaaS platform solution into the manner in which currently provided and support the VPN environment is not functionally feasible and is not a recommended best practice for organizations striving for cloud-like internal deployment platforms.

REUSE OF EXISTING SHARED SERVICES WILL BE PROMINENT

Apart from the newly established core compute and IaaS platform and some of the associated cloud management tools, there will be many examples of existing shared infrastructure services that will be reused as is by deployments in in house Cloud.

IT'S NOT ALL ABOUT A NEW BUBBLE

In addition to rallying behind in house Cloud and higher Cloud environments, are already applying some of the common underlying capabilities to our existing deployment environments. Wherever it makes business sense, it will reuse across contexts.

For example, our ability to model and automatically provision Red Hat Enterprise Linux (RHEL)-based stacks in our VMware environments is being applied to both the existing deployment environments and will form part of the in House Cloud solution. Another good example is access to Internet resources from the development environment.

If it's feasible to achieve that access from both in house Cloud and VPN, this initiative will strive to help make that happen across both environments.

6.8.7 RELATIONSHIP TO OTHER ENVIRONMENTS

Let’s check how in House Cloud is different than existing environments

Traditional Storage: Not all workloads that use VPN are immediately compatible with the initial Development Cloud service. See the Use Cases documentation for more information about deciding whether your needs may be met by the Development Cloud service.

Over the next several years, it’s been expected in house Cloud to take on more of the workloads that have traditionally targeted the VPN environment. This migration and redirection of workloads won't happen overnight.

Desktop Development: Although there are similarities with respect to the degree of latitude offered to developers, clearly in house Cloud is intended to provide a more persistent, shareable and realistic deployment environment as compared to desktop-based development. Although some current use cases supported by desktop-based development will be better served through the use of in House Cloud, there is clearly still a role for desktop-based development.

Labs: There are some clear parallels between the In House Cloud service and Labs' development environment, but there are also some key differences. Apart from these differences, In House Cloud to leverage the practical experience gained by Labs and groups working Labs as they have already faced some similar challenges. Examples of differences include:

Generally accessible to developer community: By design, Labs' development environments is focused on serving people participating in Labs' development and prototyping efforts.

Accessible from Corp: Given its general purpose orientation, In House Cloud resources must be accessible, through the appropriate firewalls and controls, from corporation based clients.

Access to internal shared services: There may be a greater degree of connectivity required between In House Cloud and existing shared services than currently enabled between Labs' Development environment and Organization internal shared services.

Stratified roles and degrees of access: Along with the broader audience, there will likely be interest in stratifying the degrees of access allowed in In House Cloud. See Use Cases below.

Existing Higher Environments: In some cases, apps will progress from Development Cloud to traditional stage. In other cases, as mentioned above, apps will be developed in a pre-Development context before progressing to traditional VPN for initial formal deployment.

6.8.7.1 STAGE AND PRODUCTION CLOUDS

What about Production Cloud? I need some elasticity for my current app soon, how does Development Cloud help? Frankly, I don't care that much about a Development Cloud at this stage of my development and rollout cycle.

Staggered, overlapping with Development Cloud: intent is to establish use of IaaS and cloud management solution in support of Development Cloud first while overlapping at least the planning for a companion Production Cloud. Don't want to commit to and pilot the solution in Production Cloud context prior to establishing at least some experience in the Development Cloud context.

Stage Cloud: After Production Cloud, It’s been expected that Stage Cloud goes hand-in-hand.

Existing workloads needing cloud benefits: For those workloads already deployed and in need of a more elastic, horizontally scale-able responses too hard to determine demand, some optimizations of the current deployment and management processes might be applicable. For example:

Adjusting/adding capacity in place: Through readjustment of existing CPU and RAM allocations, applying additional underlying compute capacity and other in-place approaches, some degree of enhancing the overall scalability of existing deployment may be feasible.

Horizontal expansion: Some of the existing stack deployment automation for RHEL-based deployments could be brought to bear to help improve some of the turnaround times to add capacity. However, that is only one piece of an overall puzzle in our current process.

6.8.8 PROMOTION TO HIGHER ENVIRONMENTS

This is great, will have a more flexible, self-service shared development environment, but won't this freedom let us drive into a ditch much faster than before?

Existing policies and standards still apply: Although developers will gain a bit more freedom and speed to realize development services, existing policies and standards continue to apply.

Experimenting with new components: When an individual becomes interested in a new component that is not yet on an approved list (e.g. Tools Portfolio Management list), the individual must still follow the Software Business Case process even for evaluation use.

Deployment designs (TADs) for new app systems: Prior to build out of new application systems to higher environments, the standard deployment design process will be applied where necessary according to existing conventions.

Deployment in VPN will apply in some cases: Especially prior to the realization of higher Cloud environments, some applications will need a deployment design and build out in VPN even though some initial development was done in In House Cloud.

Data modelling and reuse: Data Modelling continues to be a critical part of the overall analysis and design process.

New tools and procedures: Over time, additional deployment modelling and packaging tools will enable development and infrastructure teams to better describe their deployment needs such that greater degrees of promoting deployment configurations across environments can be realized. For example:

Java Web Deployment Packaging: In the Java web development context, developers will be able to declare dependencies on the web container environment such that a Meta package containing both the web container requirements and the application archive (WAR, EAR) can be provided for deployment and be applicable across environments.

Stack Modelling: Application development and infrastructure teams responsible for experimenting with, evaluating and, in some cases, preparing new stack combinations for deployment to higher environments, will have the tools and standards to perform most of the preparation tasks on their own. For example, a joint application development and infrastructure effort to standardize a Tomcat web container stack can, per available standards and tools, prepare a draft submission for the new stack without needing to depend on a variety of teams to carry out development and integration activities.

6.8.9 EXPERIENCE GAINED VIA DEVELOPMENT CLOUD

Examples of the practical experiences will gain through the establishment of Development Cloud

Self-service: At least for the use cases in scope, this degree of self-service will be a learning experience for all. It will help further position us for additional self-service as it will apply to higher level environments albeit with greater levels of controls that required in those environments.

Coarse grained lifecycle management: Practical experience will be gained in taking a completely different approach to life cycle management of our deployments. Using a combination of modelled stacks and services, automation and an agile cloud platform will enable us to gain real world experience with these arguably simpler methods of maintaining deployed systems before attempt to apply those techniques, tools and platforms to higher level environments.

Modelling and deployment automation: Developers and infrastructure service teams will be given the tools, guidance and standards to take on much of the modelling work required to prepare their deployments for automated deployment and redeployment to both tradition deployment and *Cloud environments.

Largely decouple capacity management from consumption: Unlike today's heavily virtualized deployment environments, Development Cloud will be a learning experience for us to manage a true IaaS cloud environment by anticipating overall capacity consumption trends and easily adding capacity in front of actual demand.

Cost transparency: The manner in which Development Cloud is operated as a service and through the use of cloud management tools that can provide "show back" of usage will help all involved better understand the costs involved in providing and consuming the services. This transparency of costs will also help us compare the benefits of establishing, maintaining and consuming on premise, internal cloud resources vs consumption of off-premise, external cloud resources.

What are examples of turnaround times to gain access to common resources?

Minutes to deploy common, standard services: Once an overall role has been granted to a developer, it is expected that access to common and standardized services such as a personal web container instance and/or DB instance or schema for development will be a matter of tens of minutes - all handled without human intervention. Of course, the number of such on-demand services will be relatively small at the outset given the standardization and automation required.

Day or so for custom, but supported services: In cases where important customizations are needed, say a custom FQDN for your app/service vs using the standard generic FQDNs drawn from an existing pool, you may need to wait a day or so. Similarly, if you need certs for the customer FQDNs, the request and provisioning can be automated, but there will still be a manual approval that will take time to process.

6.8.10 SECURITY AND ACCESS

How open is Development Cloud?

Role-based access: You request an Identity role to access Development Cloud services.

Shared service access: Connectivity has been established between Development Cloud and many key shared services housed in VPN. Rather than assuming wide open access to these other environments, our intent is to incrementally justify and request access such that the known dependencies become well documented and can help us understand the impact of extending Development Cloud with an off-premise cloud resources in the future.

Outbound Internet access: There is real business value in enabling developers to have ***outbound*** access to Internet-based resources much like their capabilities from their desktops today. Unsolicited inbound access is NOT going to be supported in Development Cloud. That access may be best addressed in a distinct demo cloud (see Labs for an example).

6.8.11 SUPPORT AND ROLE OF ADMINISTRATIVE GROUPS

Won't the aggressive, self-service oriented nature and access to resources within minutes be a huge additional burden on various administrative teams?

Development Cloud admin support: Strategy is to leverage the admin teams to help design, implement and support the overall Development Cloud and higher Cloud environments, procedures and high degree of automated actions rather than putting the onus on admins to be involved in day-to-day user interactions with Development Cloud. For example, standing up a dedicated JBoss web container instance stack will not require any admin intervention. However, admins will be involved in specifying how such stacks are built and provisioned.

Admin per transaction support: NOT expecting manual intervention by various admin groups in support of most normal and standard interactions with Development Cloud. That is the bar are setting. Sys admins, web admins, etc. will be oriented toward designing how to deliver Development Cloud, implementing some of the key building blocks (e.g. standard stack/service definition) and providing some level of support for the service. i.e. rather than being the machine, people should design and support the machine.

In-place updates and upgrades: Along the lines of above, it’s not been expected to put the burden on admin teams to perform in-place updates and upgrades to deployments in Development Cloud. Several approaches will be used to help developers effect updates and upgrades on their own:

- **In-place Tools**: In some cases, tools will be made available to update components in-place. For example, to enable developers who have proper privileges the ability to simply execute "yum update" to update their OS instance and any other packages that were installed via the yum tool. Later on, considering tools that can run within Development Cloud to automatically apply updates to systems. May even use Development Cloud as a proving ground for such automation before applying such capabilities to higher environments.

- **Redeployment of newer stacks:** Where in-place update and upgrade tools are not available, developers will always be able to deploy an updated form of the stack of interest. Owners of stacks will be responsible for publishing updated stacks. To make redeployment a streamlined process, developers will need to be able to export configuration and application data from older versions of the stack and import that data into the newly deployed form of the stack.

  Fleeting OS instances: Through tools, procedures and capabilities supported by Development Cloud, expected to move the broader organization to a point where OS instances aren't treated as much as special snowflakes that individually evolve through a long series of manual configuration changes.

  Working in a true IaaS cloud environment implies that there is much less reliance on static set of host OS instances. Providing users of Development Cloud with the ability to easily redeploy their artefacts to newly provisioned stack instances should help us move in this direction.

  Modelling and automation: Developers and infrastructure admins will be provided with tools and guidance that will enable them to take on much of the onus of formalizing the packaging of experimental configurations that have been shown to bear fruit and to be worth formally deploying in upper level environments or as shared services hosted in Development Cloud. i.e. Developers should not have to wait on admin and other groups to at least package their experiments such that they can progress through a review and integration process to make them available in higher level environments.

6.8.12 HOSTING SHARED SERVICES IN DEVELOPMENT CLOUD

Shared services are hosted in Development Cloud s and when required.

Production quality vs development instances of shared services: Although most production-capable shared services required by these use cases won't be hosted in Development Cloud, expected that development/test instances of some shared services will be migrated to and live within Development Cloud.

Eventually, as higher level *Cloud environments come on line, some of the production shared services will be hosted in those Cloud environments and accessed by apps and services hosted in Development Cloud.

Stubs of services: As Labs has done in some cases, where connectivity is not feasible, stubs or facsimiles of shared services may be hosted on Development Cloud.

6.8.12.1 MODEL-DRIVEN AUTOMATION

How does Development Cloud relate to modelling and automated deployment of infrastructure software stacks that was being worked before?

Provide content for some standard services: The standard RHEL-based stacks including JBoss and Apache HTTP Server and the ability to automatically provision such stacks will be directly reused in support of Development Cloud. Depending on the management requirements of Development Cloud deployments, may remove certain components from the Development Cloud-oriented stacks, but our use of Chef to model and assemble stacks using a set of modelled building blocks positions us to easily deliver a series of standard VM-based stacks for deployment within minutes.

Further enhancements:

- Formalizing automation of web container instances as part of the stack deployment process.
- Potentially expanding scope of stacks to include DB oriented stacks depending on how self-service DB instance, schema, etc. will be realized.

- Providing better documentation such that developers and admins can build, test and qualify their own stack derivatives for review and potential standardization.

The key characteristics of the cloud are the ability to scale and provision computing power dynamically in a cost efficient way and the ability of the consumer (end user, organization or IT staff) to make the most of that power without managing the underlying complexity of the technology. Cloud architecture itself can be private ("on premise" or "off-premise") or public ("off-premise").

In addition to dynamic, cost efficient and ease of use in a private or public architecture, Organization adds to that definition "within the parameters of our Information Security and regulatory policies".

For Organization these characteristics have specific benefit:

Faster: faster time to market via standard services, dynamic platforms and automated provisioning. It supports rapid increases in demand for processing and storage capacity

Better: Higher quality deployments through standardization and automation

Cheaper: Greater systems utilization and less manual provisioning and rework Commodity and open-source processing platforms

6.8.12.2 CLOUD DELIVERY MODEL

Cloud services deployment models and related topics are dominating the IT landscape.

Organizations are actively addressing these deployment models and has developed an approach to enable Organization to leverage them in a consistent manner to meet business needs.

Before delving into the cloud services deployment models and their security considerations a distinction must be made between cloud services offered by Organization and those offered by third parties.

Cloud services are typically delivered in two ways:

On-premise where Organization provides all cloud infrastructure and/or services for consumption. Organization is responsible for cloud infrastructure and/or service security controls and is the custodian for data residing within the deployment.

Off-premise where a cloud service provider manages the infrastructure and/or services and supplies them to Organization for consumption. Organization and/or the cloud service provider are responsible for security controls and data custodianship.

On-premise cloud solutions are currently being deployed within the Organization environment.

A cross functional team of subject matter experts have applied key learnings from various cloud offerings to deploy platforms that provide tangible benefits to business owners. Colloquially known as “Development Cloud and “Production Cloud” these offerings take full advantage of Cloud technology to drive time to market, operational efficiency and innovation; while ensuring that the same level of security controls implemented on the physical workloads of today are extended to the cloud workloads of tomorrow.

In many cases the trade-offs articulated throughout remainder of this document do not apply to Organization’s on premise cloud solutions.

On-premise cloud delivery greatly reduces the barriers to entry for cloud adoption as sensitive data (PII and PCI) remain under Organization’s control. The security controls that are deployed by organization meet and in many cases exceed PCI DSS recommendations providing high assurance for data protection.

However, offerings and capabilities will vary between on premise and off-premise cloud deliver solutions. Key capabilities that are offered with various off-premise cloud solutions may not be available with the Organization on premise solutions. Thoughtful and careful consideration must be done to ensure the right deployment model is selected. To learn more about Organization's on premise cloud offerings

The remainder of this document focuses exclusively on off-premise cloud services deployment models;
There is a responsibility and liability shift between cloud computing models for both Organization and the cloud service providers that must be addressed through the legal/contract processes, as well as the technology processes.

From a security perspective, Organization is responsible for evaluating cloud based systems in the same manner as any other third party solution. Organization will leverage existing processes where possible, particularly in the areas of risk assessment, legal, and Security Due Diligence. It is important that all requests and contracts be reviewed by Information Security and Data Privacy which can be started by contacting for risk assessment.

The purpose of this document is to outline the cloud computing types, appropriate utilization, and overall direction that Organization will take to address cloud services. In general, data classified as Highly Confidential – Special Handling, Privacy Data must have known locations, not spread data across state or country borders. Data also classified as Highly Confidential, Special Handling, Card Data must meet the Privacy Data requirement and be certified PCI compliant annually.

6.8.13 CLOUD CONCEPTS AND MODELS

From a strategy perspective, Organization is leveraging the existing NIST definitions for cloud standard characteristics, service models, and deployment models.

It is important to apply consistent definitions to any cloud service analysis or discussion in order to properly assess the security and operational controls that need to be applied as part of the risk management process.

It is important to engage Organization Technologies and Information Security prior to engaging any cloud computing companies or resources.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

6.8.13.1 CLOUD SERVICE MODELS

There are three generally recognized cloud service models:

- IaaS – Infrastructure as a Service
- PaaS – Platform as a Service
- SaaS – Software as a Service

Cloud security requirements may not differ by service model in essence, but how they are designed, applied, and enforced does differ. As more environmental control is relinquished moving from service model IaaS to PaaS to SaaS, the method and responsibility of incorporating security controls changes.

For Infrastructure as a Service (IaaS), the customer is responsible for building in the necessary security controls for the application, data, and systems. For

Software as a Service (SaaS) models, the customer has to RFP security controls into the cloud provider assessment, legal, and contract processes.

Organization has direct experience with the consumption of Software as a Service (SaaS) applications to deliver complete business solutions. Examples is business environment include Salesforce.com, and Workday. Organization has direct experience with the Infrastructure as a Service (IaaS) cloud model.

Examples include Development Cloud, Production Cloud & Simplify Commerce.

The consumer/customer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

NOTE: The term virtualization is used heavily in the promotion of cloud computing offerings. Virtualization is a technology that allows multiple operating systems to run on a single server as if that server were actually several separate devices.
This allows for greater resource utilization and flexibility which can drive down cost and increase deployment speeds. It is often part of a myriad of processes and technologies which are combined to provide cloud computing, but does not provide cloud computing by itself.

6.8.13.2 CLOUD DEPLOYMENT
Cloud services are typically deployed to the following environments:

1) **Public** where the cloud infrastructure is made available to the general public or large industry group.

2) **Private** where the cloud infrastructure is operated solely for one organization.

3) **Community** where the cloud infrastructure is shared by several organizations and Supports a specific community that has shared concerns.

4) **Hybrid** where the cloud infrastructure is a composite of two or more clouds bound together by technology that enables data and application portability across both to allow for load balancing between the clouds.

6.8.14 SECURITY RISKS WITH CLOUD

The following outlines several high level security risks when considering cloud computing services and models:

1) **Data protection and privacy** - protecting customer and/or constituent information becomes increasingly difficult as layers of abstraction are added
2) **Information ownership and custodianship** - those who own the data are not in a position to directly control or manage that data
3) **Regulatory and policy compliance** - Ensuring compliance with diverse regulations and enterprise specific policies is challenging due, in part, to the inability to effectively audit and maintain situational awareness
4) **Law and jurisdiction** - Clouds cross lines, borders, and oceans representing a highly diverse set of liabilities, ramifications, and risk
5) **Vulnerability and malware management** – Ensuring vulnerabilities to threats are mitigated in a meaningful timeframe and effectively, and obtaining assurance can be complex and time consuming
6) **Access and authorization management** - Having the degree and granularity of control required for the company is not always possible or feasible in the cloud
7) **Identity management** – Governing identities becomes increasing complex and Authoritative control may not be possible
8) **Infrastructure sharing** – Impact to the enterprise due to vulnerabilities exposed on shared infrastructure increase brand and reputational risk

The risks of using cloud computing must be compared to the risks of staying with traditional solutions. It is sometime possible for the cloud customer to transfer risk to the cloud provider, but not all kinds of risks.
If a risk leads to the failure of a system, serious damage to reputation or legal implications, it is difficult to impossible for any other partner to compensate for this damage.

6.8.15 EVALUATING CLOUD SERVICES

Normal channels will be leveraged to support the business through a complete evaluation of the proposed cloud based service or solution. This involves cross-functional coordination between the business, legal, and technical teams within Organization. Additional questions related to cloud computing models, types, and environments will be part of the Security Due Diligence risk assessment process.
As Organization develops solutions to secure and support cloud based services, goal would be to raise awareness across the enterprise on the overall Organizations strategy.
Although cloud models represent alternatives to provide solutions, the process of assessing risk and developing security and operational models can leverage the same assessment processes and procedures that are in place today at organizations.

Cloud service assessment can leverage the following existing Organization processes

1) **Data classification –** appropriate classification drives overall security control. Information Security helps assess the overall business solution early on in the concept/design phase.

2) **Security Due Diligence:** This program helps Organization meet the requirements Outlines in the Risk Assessment Standard. Detailed information is available internally on

3) **Legal and Contract:** Normal processes must be followed to establish any business relationship with a third party vendor or cloud service provider.

4) **Privacy and Data Usage Policy:** It includes usage and distribution controls. Any Organization using cloud based solution will require contractual agreement between Organizations and the Cloud Service Provider. Cloud based solutions must adhere to the Legal, Risk Assessment, and Security due Diligence policies, Standards and procedures.

When assessing a cloud based service or solution, the following policies and standards represent the starting point for identifying proper security and operational requirements:

- Organizations Information Security Policy
- Global Privacy and Data Protection Policy
- Risk Assessment Standard
- Data Lifecycle Standard; Classification, Protection, Media and Disposition
- Virtualization Security Standard
- Security Logging and Monitoring Standard
- Unix Security Standard
- Network Security Standard
- Mobile Device and Personal Equipment Standard
- Windows Security Standard

6.9 CLOUD COMPUTING AT ORGANIZATION

6.9.1 CLOUD COMPUTING PART OF IT TRANSFORMATION

Yes! Cloud Computing has the potential to address all three areas of focus (Technology, Operations and Cultural) and provide a platform that will be critical component of our transformation.

6.9.2 ORGANIZATIONS CURRENTLY USING CLOUD COMPUTING

Organizations currently use a number of SaaS offerings including human resources (Work Day), Organizations University (Cornerstone), sales/account management (SaleForce.com) and website acceleration (Akamai). They have invested heavily in technologies that are required for cloud computing such as virtualization and automation. Plan include limited deployments of applications in development on both a private cloud and a public cloud.

6.9.3 GETTING APPLICATION INTO THE CLOUD

There is significant interest in exploring the deployment of Organizations applications in the cloud. Deployment of applications through a public or private cloud presents a number of challenges for Organizations including security, access, and management. Working closely with technical architects, security experts and legal, are creating a framework for evaluating application readiness or fit for cloud.

6.9.4 BACKGROUND AND STRATEGIC USE OF CLOUD COMPUTING.

It is the objective of Organization Operations & Technologies (O&T) to drive benefit to Organization, its customers, and cardholders through the use of Cloud Computing techniques, technologies, and third-party services. Cloud Computing is a buzz word of the information technology and business world. Cloud Computing has as many definitions as there are vendors and service providers selling its virtues.

The purpose of this document is two-fold:

1. To provide a consistent definition and view of the value of Cloud Computing.
2. To document our high level strategy and the current status of Cloud Computing at Organization.

6.9.5 IT TRANSFORMATION AND CLOUD COMPUTING

Organization is on a journey to transform itself. Have an imperative to take advantage of current position of strength to prepare for the future. Transformation is organized into three areas of focus:

- *Technology Transformation* - more effective and efficient use of technology used, focusing on technologies that serve multiple business goals.
- *Operations Transformation* - maintaining a high level of service of our infrastructure and applications while responding efficiently and effectively to the changing needs of our business and customers.
- *Cultural Transformation* - looking at the way things are done within Organization, from career development, communications and international opportunities to increasing our business savvy while recognizing and rewarding behaviors that help us achieve our corporate goals.

Cloud Computing has the potential to play a key role in all three areas of focus and provide a platform that will become a critical component of Organization’s transformation.

6.9.6 CLOUD COMPUTING DOES

- Addresses the goal of effective and efficient use of technology by standardizing and simplifying the platforms used.
- Supports our current high level of service and the speed can meet new and changing business objectives through automation.
- Provides us the versatility and scalability needed to meet the array of technology and capacity requirements are expected to satisfy now and in the future.

- Changes the way operate. It empowers teams to quickly define their needs and deploy the platforms they need to solve business problems and meet business needs.

6.9.7 CLOUD COMPUTING BENEFITS TO ORGANIZATION

The characteristics of the cloud computing as defined by the Open Cloud community[1] are the ability to scale and provision computing power dynamically in a cost efficient way and the ability of the consumer (end user, organization or IT staff) to make the most of that power without managing the underlying complexity of the technology. Cloud architecture can be private (both "on premise" or "off-premise") or public (only "off-premise").

In addition to dynamic, cost efficient and ease of use in a private or public architecture, Organization adds to that definition "within the parameters of our Information Security and regulatory policies".

For Organization these characteristics have specific benefit:

Faster

- Faster time to market via standard services, dynamic platforms and automated provisioning
- Supports rapid increases in demand for processing and storage capacity

Better

- Higher quality deployments through standardization and automation

Cheaper

- Greater systems utilization and less manual provisioning and rework
- Commodity and open-source processing platforms

---

[1] Open Cloud Manifesto - http://opencloudmanifesto.org/opencloudmanifesto1.htm

6.9.8 CLOUD COMPUTING EVOLUTION AND PROGRESS AT ORGANIZATION

While the impact of adopting Cloud Computing technology is revolutionary in terms of the change that it enables, it is really part of an evolutionary journey that began at Organization several years ago. That evolution can be viewed in the graphic below.

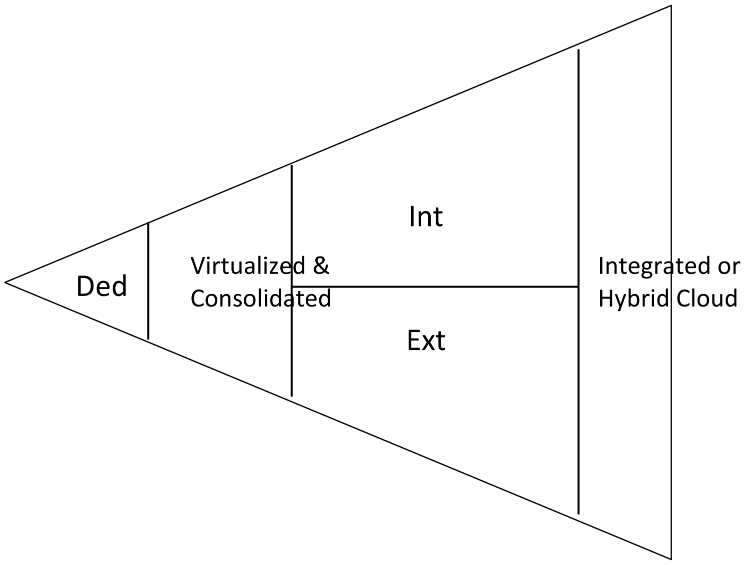


Figure 6. 3: Evolution of Cloud Computing (Y-axis - Maturity)

Each stage can be defined in terms of five essential characteristics – service provisioning, network access, resource utilization, elasticity, and service measurement.

6.9.8.1 DEDICATED

Dedicated processing environments are traditional information technology implementations where infrastructure (servers, network devices, storage devices, etc.) are dedicated or largely dedicated to a single application or service. Dedicated environments are typically higher cost environments than other environments in this continuum due to the lower utilization of the infrastructure.

Current State: Today Organization still employs a declining number of dedicated environments for applications. Typically, dedicated environments are used for applications with unique infrastructure requirements or applications which must be segregated for security or regulatory purposes.

6.9.8.2 VIRTUALIZED

At their basic level, virtualized environments employ software technology which allows multiple operating systems to run on a single server as if that server were actually several separate devices. The value of the virtualized model is that a single piece of infrastructure can support many more applications. This model is more cost effective but requires applications to run in more standard configurations.

Current State: Today Organization heavily utilizes virtualization technology in its UNIX/Linux and Windows server environments. Note: Outdated reference? -> Organization has over 71% of all server workloads and 7% of desktops virtualized. By comparison, the industry average for server workloads is estimated at around 50%.

It is important to understand that virtualization is a technology and that Cloud Computing is an operational model.

The next three stages in the evolution towards Cloud Computing leverage the foundation of virtualized server platforms and begin the transition to a fully integrated Cloud Computing environment. Moving from virtualization to achieve utilization economies to Cloud Computing requires additional capabilities, such as policy-driven automation, metering systems, self-service provisioning portals and development and processing platform standardization.

Figure 6.4 - Cloud Computing Model below shows the interrelationships between each of the Cloud Computing models described in the remainder of this section.

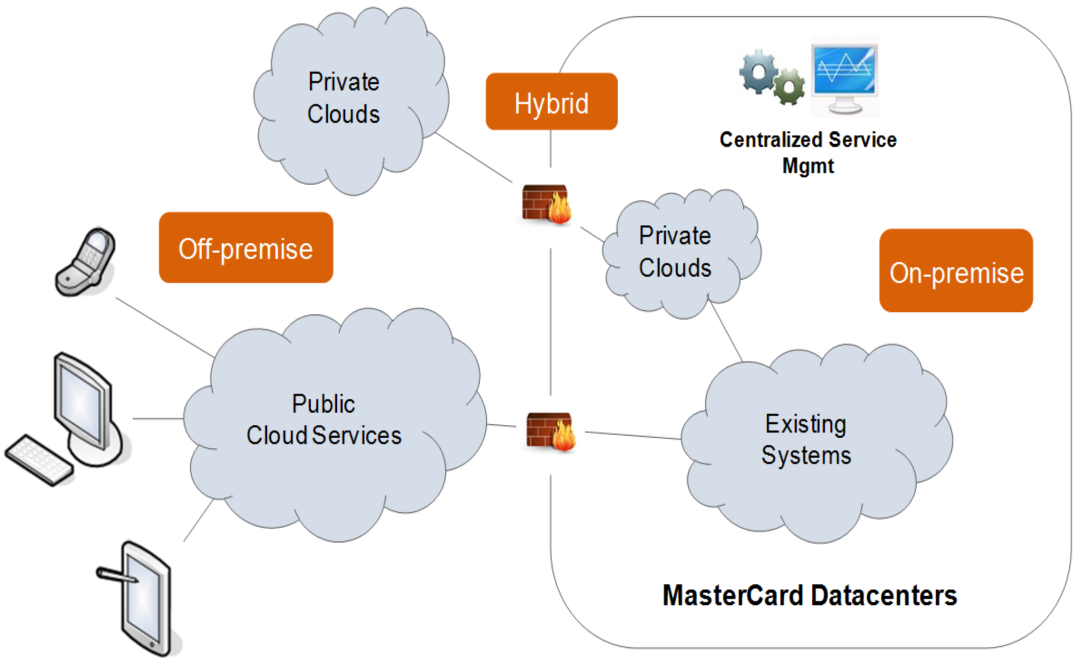


Figure 6.4 - Cloud Computing Model

6.8.9.3 INTERNAL CLOUD

An Internal Cloud Computing model implements the additional capabilities described above within a private data center to gain the efficiency, scalability, and speed benefits of Cloud Computing, but with the added benefits of easier and more secure integration with other on-premise infrastructure and application services.

In 2011, Organization O&T funded the project “Development Cloud On-Premise Infrastructure as a Service” to begin developing the capabilities needed for an Internal Cloud Computing environment. This project was the first step in enabling Application Development and Operations teams to accelerate the provisioning of computing environments. The resulting Development Cloud service has been a huge success, enabling development teams to spin up hundreds of environments in ‘self-serve’ fashion and greatly reducing the time to deliver and associated costs.

In 2013, Organization O&T funded the project “Production Cloud On-Premise Infrastructure as a Service” to allow for similar capabilities in higher-level stage and production environments. This effort continues into 2014.

6.8.9.4 EXTERNAL CLOUD

An External Cloud Computing model utilizes third-party service provider's infrastructure as a platform for developing and delivering new solutions. It has the same efficiency, self-service, and scalability characteristics as the Internal Cloud. The external cloud does offer additional benefit in that the operations, capacity management, hardware and software maintenance, and facilities management are the responsibility of the service provider. Use of the External Cloud can be limited by the architecture of an application, unique hardware requirements, and security and data privacy policies.

6.8.9.4.1 Software as a Service (SaaS)

In addition to an alternate method of providing infrastructure and platform related services External Cloud services include a service typically referred to as SaaS - Software as a Service. SaaS applications are developed and hosted by the SaaS vendor which the end user accesses over the Internet. Unlike traditional packaged applications, the SaaS vendor owns the software and runs it on computers in its data center. ***Current State:*** Organization today is already utilizing SaaS solutions from companies such as Salesforce.com, Workday, and Success Factors.

Looking forward, continue to support the further adoption of ready-to-use business applications whose development is not core to our business. As expanded use of such services for applications, will continue to refine existing policies and standards that will make such adoption a more conventional part of our "Build vs. Buy vs. Subscribe" decision-making process.

6.8.9.4.2 Platform as a Service (PaaS)

Although less established and not as widely available as compared to SaaS and IaaS offerings, ready-to-use application platforms offer great promise for organizations that aren't compelled to own and manage the underlying infrastructure.

Current State: Organization is not currently utilizing Platform-as-a-Service offerings in an External Cloud.

As progress with the development of our External Cloud services use, continue to evaluate and potentially adopt application services such as ready-to-use database or messaging services. The evaluation of more complete cloud-based application platforms will occur as this market matures.

6.8.9.4.3 Infrastructure as a Service (IaaS)

The virtualized processing, storage, and networking services - along with automation and management capabilities in this area - offer the most flexible level of services in the Cloud Computing model. Along with the flexibility of these generic services comes the cost of an organization managing all of the services on top of the IaaS platform.

Current State: At this time, Organization O&T is actively engaged in initiatives that utilize external IaaS offerings – the most notable example is the Simplify Commerce product suite at the Internet ISP peering centres. Organization also uses Akamai's Edge Caching solutions as both a solution to enhance web-site performance as well as a mitigation strategy against Distributed Denial of Service attacks (DDoS).

6.8.9.5 HYBRID CLOUD

A Hybrid Cloud uses a combination of external and internal cloud services. This architecture attempts to combine the best of both worlds – offering the security, data privacy, compliance, and control of the internal private cloud as well as the flexibility and speed of the public cloud without having to provision peak capacity. For example: Using an external cloud service as a temporary platform for development and test environments and then migrating products onto the Internal Cloud as they "go live" and require the more rigorous access and security controls of the Internal Cloud. Or conversely, starting the initial offering of a product on a limited basis on the Internal Cloud and then migrating the product to the secured, External Cloud as usage and geographic expansion occurs.

While the promise of this model is significant, there are plenty of issues around actually implementing hybrid clouds. For example, moving seamlessly between the Internal and External Clouds requires strong standardization and cross-cloud cooperation. Today's Cloud Computing standards continue to evolve and mature, and therefore make this requirement difficult to design for.

The implementation of a full Hybrid Cloud Computing model is a future consideration in Organization's Cloud Computing strategy. At this time, focus on ensure driving towards the implementation of standards that will support the ease of movement between our Internal Cloud services and External Cloud services.

6.9.9 CLOUD COMPUTING WORKING GROUP

In order to manage the definition, direction, and value received from Cloud Computing, Organization O&T formed a Cloud Computing Working Group. The working group is made up of cross-functional representatives across Operations, Application Development, Enterprise Architecture, and Global Information Security. The working group is sponsored by Edgar Aguilar and Gary VonderHaar.

The primary purpose of the Cloud Computing Working Group is to ensure that initiatives inside and outside of Organization O&T utilizing cloud services or technology have direction and support that maximizes the investment and meets established objectives, all while adhering to Organization's security and operational requirements. Through coordination with the working group, initiatives also gain further visibility as part of a holistic, corporate cloud story.

6.9.10 IMPLEMENTATION STRATEGY

The implementation strategy for Cloud Computing at Organization is three-fold:

1. To utilize Cloud Computing techniques and technologies to enhance our internal development and operational capabilities. Resulting in faster deployment of technology and more cost-effective utilization of our infrastructure.

2. To utilize external third-party Cloud Computing services to take advantage of their ability to
    a. Provide unique application functionality,
    b. Supplement our infrastructure and processing capacity, and
    c. Provide accelerated services to enhance time-to-market for our products and services.
3. To develop interoperability and portability between our Internal and External Cloud Computing capabilities.

It should be emphasized that this strategy will be implemented within the necessary parameters of our Information Security and regulatory policies.

6.9.11 INITIATIVES

There are many ongoing initiatives that fall within the visibility of the Cloud Computing Working Group. Intent is not to manage these but rather to provide assistance and direction to the initiatives.

6.9.11.1 CLOUD SECURITY

- Cloud computing at a Glance: Information Security policy reference material which is intended for key stakeholders such as business owners.

6.9.11.2 DEVELOPMENT CLOUD EXPANSION

- A new Development Cloud region is being established in Europe to better serve the needs of our European development centers by reducing latency between developer workstations and workloads deployed to Development Cloud
- Additional levels of connectivity to shared services in adjacent environments, new and refreshed turnkey deployment templates, enhanced documentation and automation are planned to enable more workloads to migrate from traditional development to the more efficient and dynamic Development Cloud hosting environment

6.9.11.3 STAGE AND PRODUCTION CLOUD

- The Production Cloud project is establishing an on-premise private cloud service oriented towards supporting elastic workloads.
- Building on capabilities from Development Cloud this initiative established a new elastic, on-premise private cloud hosting capability for Stage and Production workloads in Organization's STL data center. This capability will help relieve pressure from projects to host on External Cloud platforms by providing an on-premise solution.

6.9.11.4 PUBLIC CLOUD

- Complementing Labs-driven effort by exploring operational, security and life cycle management considerations for Production deployment on the cloud
- Involves running existing apps without architectural change in parallel to on-premise copies for a limited duration followed by decommissioning
- Continue to leverage and evaluate partners through multiple initiatives:
    - Proof of concept for Big Data in the Cloud through Amazon Web Services (AWS)

CHAPTER 7

SETTING UP AUTOMATED BUILD/TEST SYSTEM (JENKINS) OVER CLOUD

CHAPTER 7
SETTING UP AUTOMATED BUILD/TEST SYSTEM (JENKINS) OVER CLOUD

7.1 INTRODUCTION

This Chapter specifies the setting up the infrastructure over cloud to achieve automated (dynamic and adaptive) Build system to achieve continuous code integration , doing automated build periodically on the latest code checked In repository over cloud, doing code coverage and static code analysis with tools(SonarQube)and plugins working with Build System and then continuous delivery and deployment of software. And Generates reports for inspections and corrections. This Build System supports any programming language and Static Code analysis supports continuous inspection of code quality and automatic review to detect bugs, code smalls, code complexity and security vulnerabilities works with 20+ Programming Languages

Jenkins is used for setting up adaptive build and test system.

Jenkins is a free and open-source continuous integration, Continuous Build, Continuous Test and Continuous Delivery software tool written in the Java programming language for testing and reporting on isolated changes in a larger code base dynamically in real time. Hence as per the reports generated engineers can find and solve defects/bugs/problems in a code base rapidly and to automate testing of their builds.

Jenkins have around 1,400 plugins to support the automation of all kinds of development and software testing tasks. i.e. building projects, running tests, doing static code analysis, and deploying are only few of many processes that engineers automate with Jenkins. Plugins span in five areas includes platforms, User Interface, administration, source code management, and, build management. It can be hosted in a cloud-based container or Virtual Machine. It follows the concept of Agent, which is typically a server machine, or container, which connects to a Jenkins master server and executes tasks as and when directed.

7.2 PREREQUISITE

7.2.1 HARDWARE REQUIREMENTS:

1) A Physical/Virtual Machine
   a. Installed memory(RAM) 16.0 GB or 32.0 GB
   b. Either 32 or 64-bit Operating System.
      i. Now a day's most of the server machines are coming with 64 bit configuration.
   c. Processor® CPU @2.80 GHz
      i. The more virtual CPU, more it will be beneficial to execute concurrent threads.
   d. Hard Disk : 200 GB
      i. The more hard disk space we will have the more it will be beneficial.
2) There may be 2 or more machine (virtual, physical) but more machines may be required as per project requirements.
   a. Setting up Slave machine is followed in coming sections.

7.2.2 SOFTWARE REQUIREMENTS:

7.2.2.1 Java: AS JENKINS ITSELF A JAVA APPLICATION, HENCE PROPER SUPPORTING JAVA VERSION SHOULD BE INSTALLED BEFORE INSTALLING JENKINS.

a. Java 8 Latest Edition (I installed 8u_151, 32 or 64 bit version or both together)
b. Download java from http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html

Select “Accept License Agreement” and Windows x86 and follow the installation instructions.

Set **JAVA_HOME**:

Right click My Computer and select Properties.

a. On the Advanced tab, select Environment Variables, and then edit JAVA_HOME to point to where the JDK software is located, for example, C:\Program Files (x86)\Java\%jdk%

   Here jdk = jdk1.8.0_151.

b. Verify Java version at command prompt by running command **java –version.**

   Command will run successfully it Java is installed properly and will show values as following.

```
C:\Users\dc-user>java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) Client VM (build 25.151-b12, mixed mode)

C:\Users\dc-user>_
```

Figure 7.1: Command to Check Java Version

This is essential as we need to get project content from git repository, windows system may come with preinstalled git but it’s always better to install latest version of git.

c. Download “GIT” from https://gitforwindows.org/ , you will get latest version always. Or just go to http://git-scm.com/download/win and the download will start automatically. It will download latest 64 bit GIT version.
d. Run Setup and follow the instructions, let default components be installed.

i. Select Get Editor of your choice or keep the default one as shown in picture.

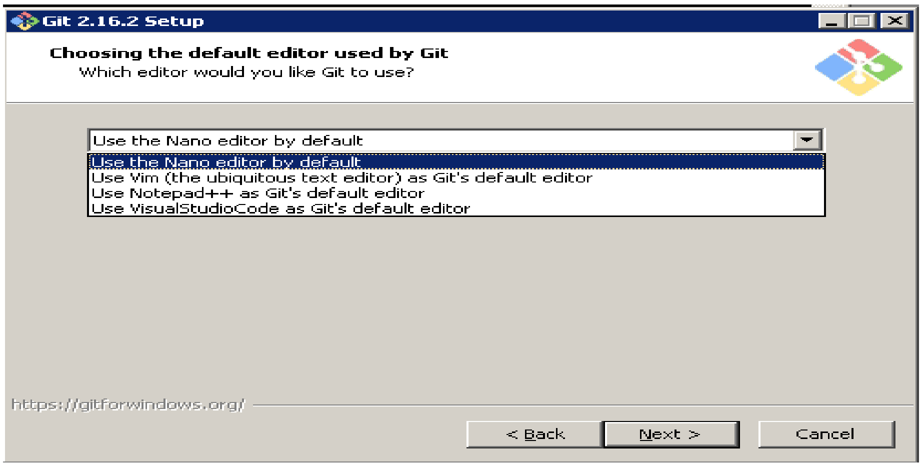


Figure 7.2: Choosing the default editor by Git

e. Either select Use Get from Get Bash only or "Use Get and optional Unix tools" optional UNIX tools may help in configuring having more command line option

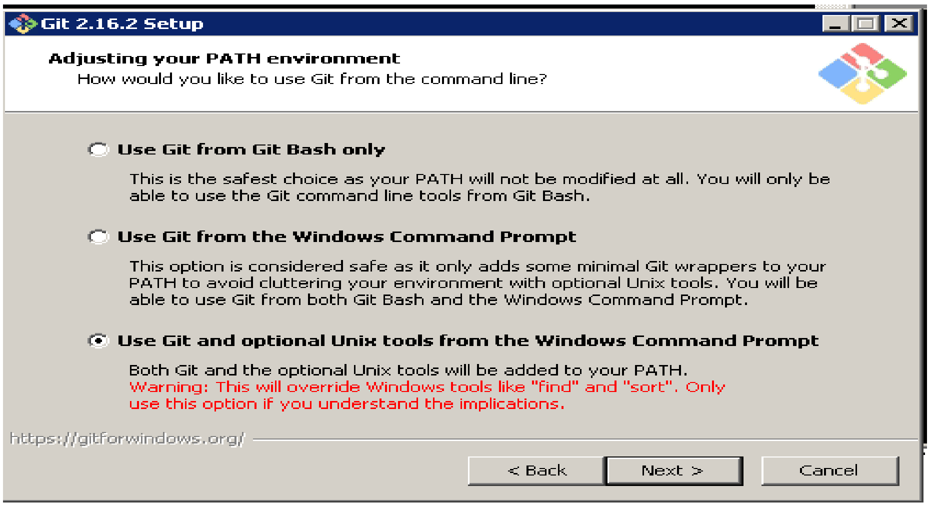


Figure 7.3: Adjusting Git Path Environment

f. Select First Option "Use open ssl Library"

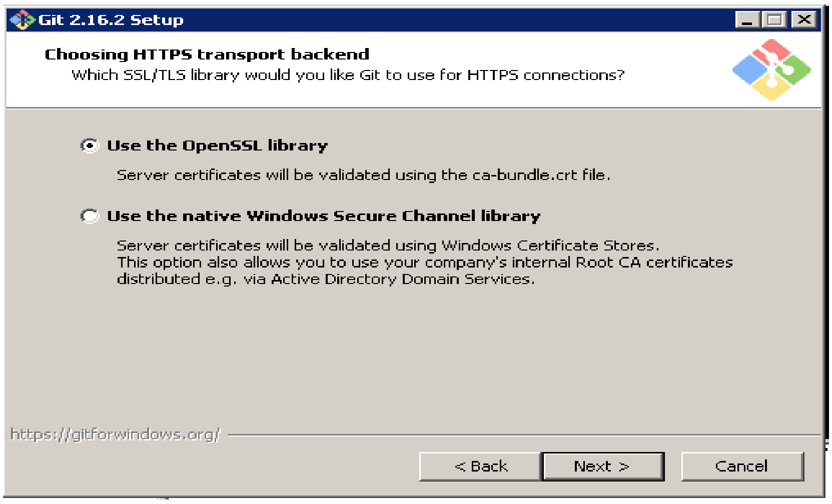


Figure 7.4: Choosing HTTPS transport backend

g. Click Next and configure "line ending conversions as mentioned below"

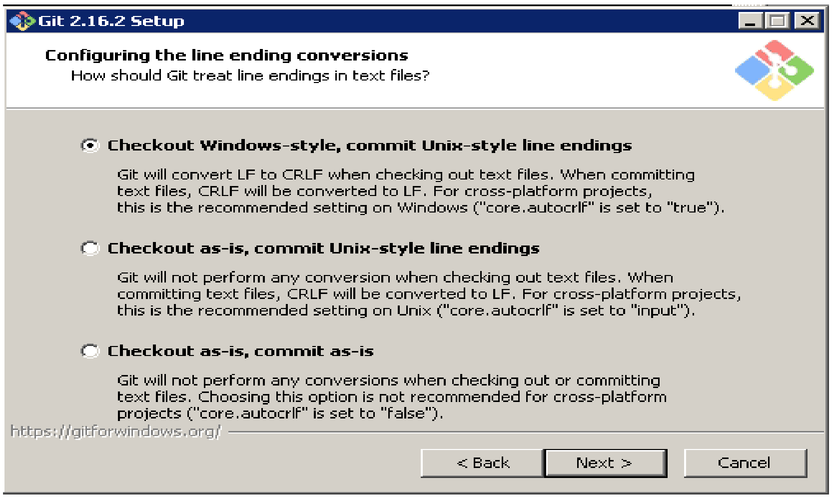


Figure 7.5: Configuring the line ending conversions

h. Click next and select default MinTTY terminal.

i. Click Next and let default option be selected.

j. Finally click install and setup will install selected components.

k. Click Finish.

l. Now Right Click My Computer -> Properties ->Advanced System Settings->EnvironmentVariables->New

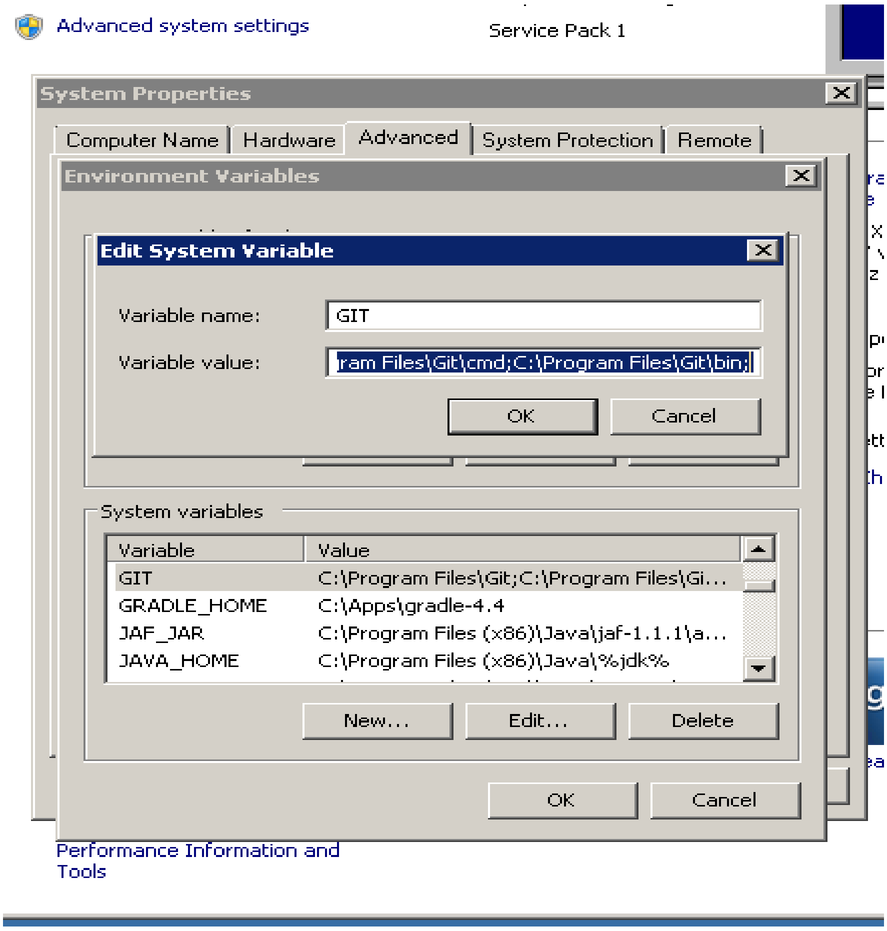


Figure 7.6: Adding system variable Git

m. Set New system variable GIT =C:\Program Files\Git;C:\ProgramFiles\Get\cmd;C:\Program Files\Get\bin;

n. Add %GIT% to Path variable.

7.2.2.2 **Download and Install Notepad++ to edit configuration files in projects and Sonar.**

7.2.2.2.1 https://notepad-plus-plus.org/download/v7.5.5.html

7.2.2.2.2 Either 32 bit or 64 bit.

7.2.2.2.3 Let install with default options.

7.2.2.3 Download 7-Zip from https://www.7-zip.org/download.html

7.2.2.3.1 Just click and install.

7.2.2.3.2 This utility is useful to extract compressed files and decompress the contents to send them over mail.

7.2.2.4 **DOWNLOAD AND INSTALL** Android

7.2.2.4.1 Android Studio will install all necessary tools and Libraries required for any android project.

7.2.2.4.2 Download latest android studio from https://developer.android.com/studio/index.html

7.2.2.4.3 To install Android Studio on Windows, proceed as follows:

Launch the .exe file you downloaded. Follow the setup to install Android Studio and additional SDK tools (Install all tools).

You may see short video at https://developer.android.com/studio/install.html

7.2.2.4.4 Setup "ANDROID_HOME=C: \Users\dc-user\AppData\Local\Android\Sdk" environment variable, here dc-user is y username, in your case it may be different.

7.2.2.5 Download and Install MS Visual Studio Latest Version**.**

7.2.2.5.1 This need only for the VS projects and No need to install on every Machine but on a specific machine as this take much space.

7.2.2.5.2 All components need to be installed on a specific slave node as it takes about 40 GB of space.

7.2.2.5.3 Download community edition from https://www.visualstudio.com/downloads/

7.2.2.5.4 Community edition may expire after a month hence if you have full Visual Studio Professional with Key then use it.

7.2.2.5.5 Follow the instruction mentioned at :

7.2.2.5.6 https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio

7.2.2.5.7 Add “MSBUILD_HOME=C: \Program Files (x86)\MicrosoftVisual Studio\2017\Community\MSBuild\15.0” in environment variable and add %MSBUILD_HOME” in path variable.

7.3 SETUP SONAR

This includes setting up sonarqube and sonarqube scanner.

7.3.1 DOWNLOADING SONARQUBE SERVER AND SCANNER

a. Download latest sonar version from https://www.sonarqube.org/downloads/

b. You may either download latest or LTS (Long term supported) version.

c. Once it is download then follow the instructions from https://docs.sonarqube.org/display/SONAR/Get+Started+in+Two+Minutes

1. Unzip the SonarQube distribution after its downloaded. (Download Page)

2. Start the SonarQube server:

https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner

Either select Windows 64 bit or Any and extract under “C:\Apps”.

I used Windows 64 bit ver.

7.3.2 SETTING UP ENVIRONMENT VARIABLES

Set SONARQUBE_HOME=C:\Apps\sonarqube-7.0

Set SONAR_RUNNER_HOME=C:\Apps\sonar-scanner-3.0.3.778

Add %SONARQUBE_HOME%\bin\windows-x86-32 to path variable.

After updating path as per step Sonar can be run just by typing StartSonat.bat

7.3.3 RUNNING SONAR AS A SERVICE (INSTALL AND RUN FROM THE LOCATIONS)

Run Sonar as a Windows Service SONARQUBE_HOME%/bin/windows-x86-32/InstallNTService.bat

StartSonar service %SONARQUBE_HOME%/bin/windows-x86-32/StartNTService.bat

Sonar service will run automatically and same need to configure with slave nodes as well.

7.3.4 MODIFY THE PORT WHERE SONAR IS RUNNING, GO TO FOLLOWING.

C:\Apps\sonarqube-6.7\conf (you will see sonar.properties and wrapper.conf)

1. Update the sonar-properties file for port as 9010 sonar.web.host: 0.0.0.0 sonar.web.port: 9010 sonar.web.context: /

2. Change the wrapper.conf and add following line wrapper.java.additional.3=-Djava.io.tmpdir=../../temp/

3. Run Sonar as a Windows Service %SONARQUBE_HOME%/bin/windows-x86-32/InstallNTService.bat

4. Start Sonar service %SONARQUBE_HOME%/bin/windows-x86-32/StartNTService.bat

5. Change the Sonar runner properties file sonar.host.url=http://localhost:9010

Other Sonar Related Configuration with Jenkins will be given next.

7.4 SELF-SIGNED CERTIFICATES AND WINDOWS CREDENTIAL MANAGER

7.4.1 CONFIGURE SELF SIGNED SSL ON GET FOR WINDOWS

The self-signed certificate can be configured on Windows using the following steps:

7.4.1.1 USING INTERNET EXPLORER.

1. open the URL
2. Once the page has loaded click on the padlock next to the address bar
3. Select the root certificate
4. Now click on the "View Certificate" then "Details" tab and then the "Copy to file" button
5. Select the "**Base64 encoded**" option and then export to a **.cer** file on your local file system
6. Now configure Get to use the downloaded .cer file using the Get config command

```
get config --global http.sslCAInfo C:/Users/e069511/certificate.cer
```

7.4.1.2 Exporting a certificate using the Chrome browser

1. Connect to the website using SSL (https://whatever)

2. Since Chrome version 56, you do the following:
   go to the Three Dots Menu -> More Tools -> Developer Tools,
   Then click on the Security Tab.
   See **Security Overview** with a **View certificate** button.

3. Click on the **View certificate** button.
   A modal window will open. It has two panes.

   The top one shows the trust hierarchy of the

   sites certificate (the last one listed), the intermediate certificate(s),

   And the root certificate (the topmost one).

   The second pane, shows the details of the certificates.

   There may be zero or more intermediate certificates.

   The root certificate equipped with a gold-bordered icon. The others have a blue border. See the screen shot below.

4. To export a certificate:
   1. First click on the certificate's icon in the trust hierarchy.
   2. The certificate will be shown in the main part of the modal.
   3. Click on the certificate's large icon in the main part of the modal.
   4. Export the certificate to your desktop.

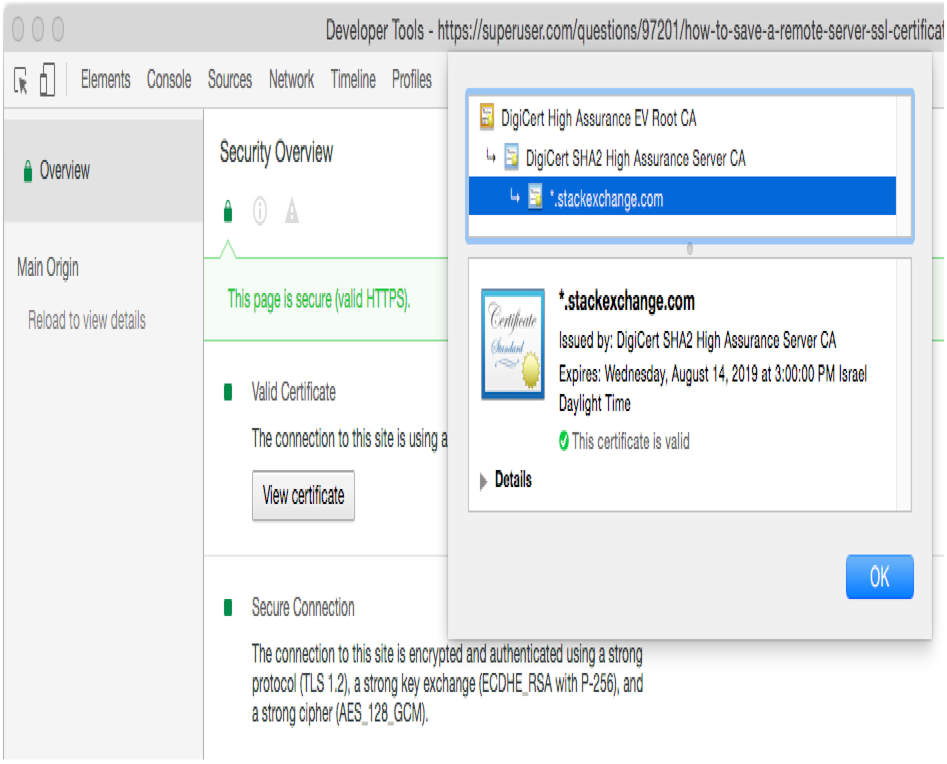


Figure 7.7: How to save a remote server ssl certificate

Use Unix2Dos for appending certificate information in existing certificate.

7.4.2 CONFIGURING CREDENTIAL STORAGE WITH GIT

We are using Standard Windows Credential Manager with GIT to store credentials called as wincred.

Run Following command get config --global credential.helper wincred

And then do checkout/update from repository, a window will popup asking GIT credentials (username and password). Then it will store them permanently until changed with GIT itself.

Changes at Windows credential Manager can be made manually as well. Type “Credential Manager” inside Run

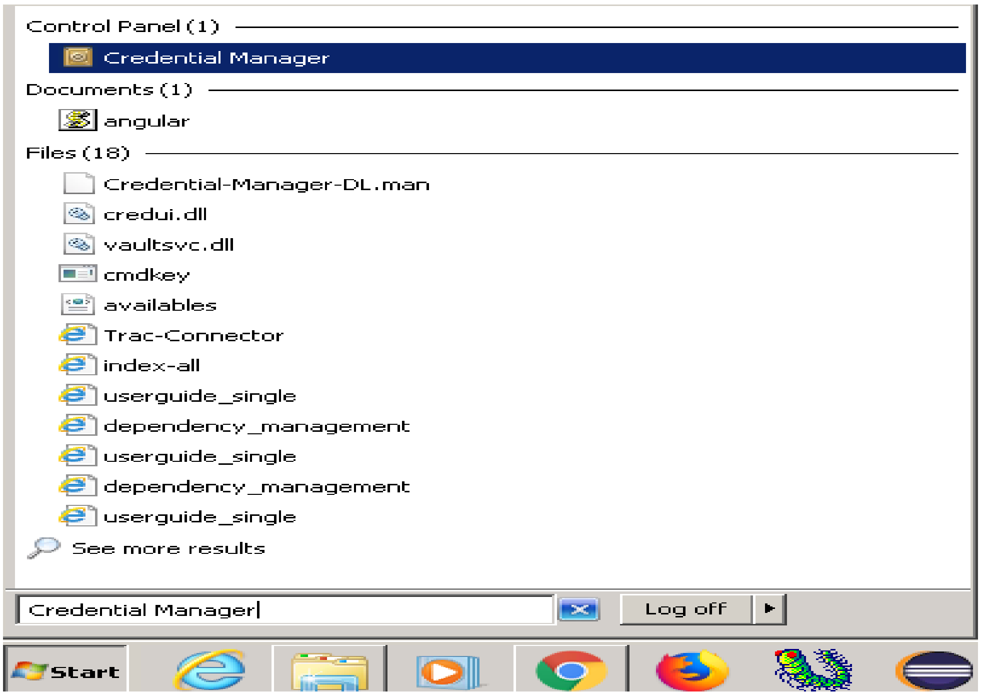


Figure 7.8: Windows Credential Manager

Click on "Credential Manager", you may see options to Add, Edit and Delete under section "Generic Credentials".

7.5 SETTING UP JENKINS

7.5.1 DOWNLOAD AND INSTALLATION

a) Download latest Jenkins windows installer from https://jenkins.io/download/

Either LTS or Weekly version

Windows

To install from the website, using the installer:

- Download the latest package (Jenkins.msi)
- Open the package and follow the instructions

b) Check installation instructions at https://jenkins.io/doc/book/installing/

After Downloading Jenkins zip file, decompress it and install as following

WAR file

The Web application Archive, file version can be installed on any operating system or platform which supports Java.

To install the WAR file version of Jenkins:

1. Download the stable Jenkins WAR file on your machine.
2. Open up a terminal/command prompt window to the download directory.
3. Run the command `java -jar jenkins.war`.
4. Browse to `http://localhost:8080` and wait until the **Unlock Jenkins** page appears.
5. Continue with the Post-installation setup wizard mentioned next .

After Downloading and installing, we need to do post installation setup wizard.
https://jenkins.io/doc/book/installing/#setup-wizard

7.5.2 POST-INSTALLATION SETUP WIZARD

After successful installation and run of Jenkins using one of the procedures above, the post-installation setup wizard begins.

This setup wizard takes through few quick "one-off" steps to unlock Jenkins, customize it with plugins and create the first administrator user through which you can continue accessing Jenkins.

7.5.2.1 UNLOCKING JENKINS

When new Jenkins instance starts, asked to unlock it using an automatically-generated password.

In windows it will be at following location: C:\Program Files (x86)\Jenkins\secrets

1. Browse to http://localhost:8080 (or whichever port you configured for Jenkins when installing it) and wait until the **Unlock Jenkins** page appears.

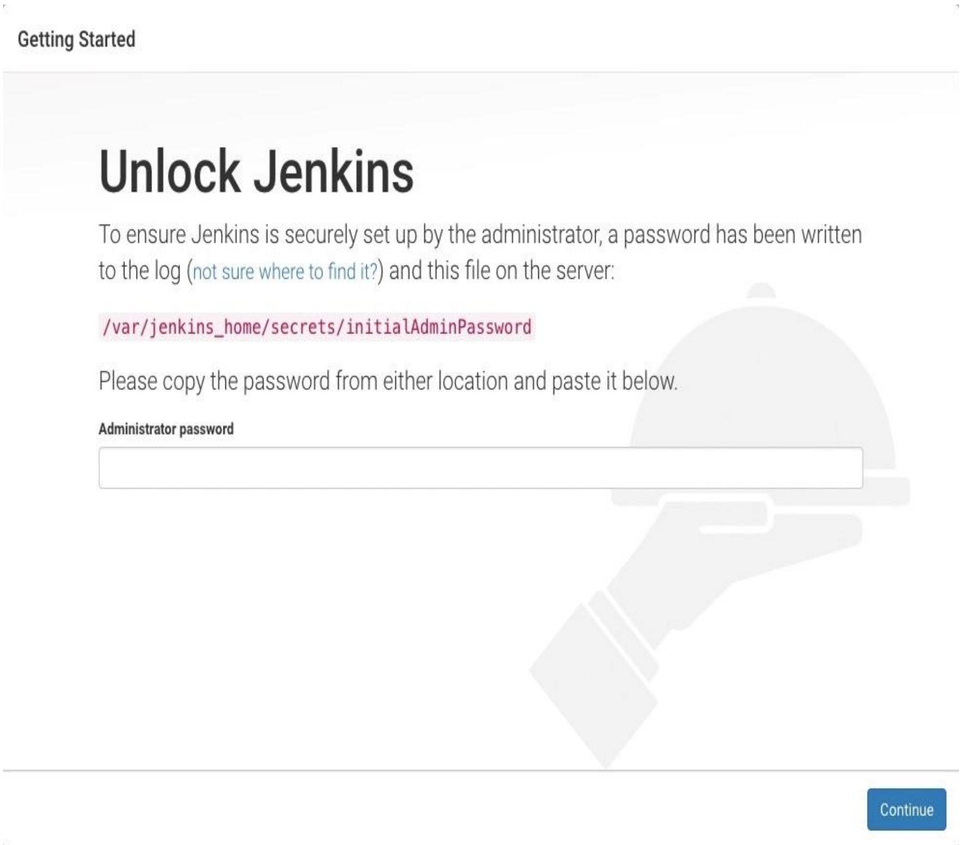


Figure 7.9: Unlocking Jenkins

2. From Jenkins console output, copy automatically-generated alphanumeric password (between the 2 sets of asterisks).

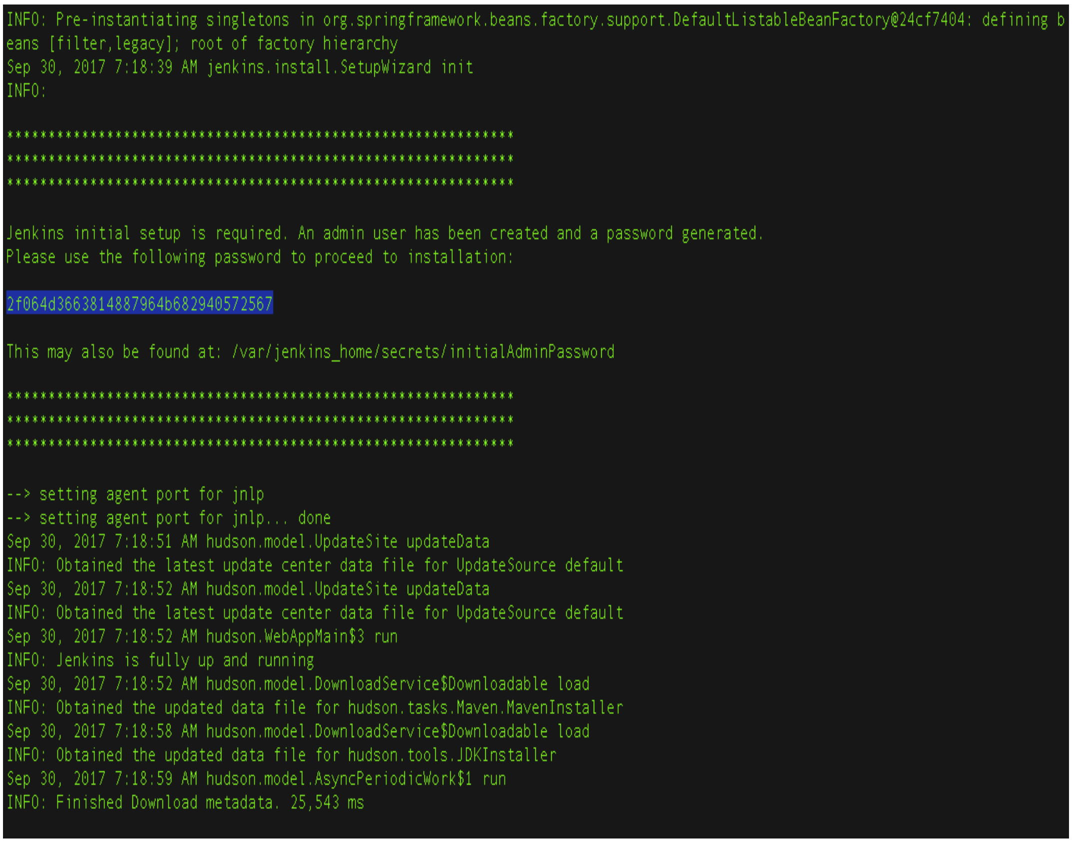


Figure 7.10: Jenkins Default Password

3. On the **Unlock Jenkins** page, paste this password into the **Administrator password** field and click **Continue**.

   Notes:

   - If you ran Jenkins in Docker in detached mode, you can access the Jenkins console log from the Docker logs (above).
   - The Jenkins console log indicates the location (in the Jenkins home directory) where this password can also be obtained. This password should be entered in the setup wizard on new Jenkins installations before you can access Jenkins’s main UI. This password also serves as the default administrator account’s password (with username "admin") if you happen to skip the subsequent user-creation step in the setup wizard.

7.5.2.2 CUSTOMIZING JENKINS WITH PLUGINS

After unlocking Jenkins, the **Customize Jenkins** page appears. Any number of useful plugins can be installed as part of initial setup.

Click one of the two options shown:

- **Install suggested plugins** - install the recommended set of plugins, which are based on common use cases.
- **Select any plugins to install** – may choose which set of plugins to initially install.

Choose **Install suggested plugins if not sure of plugins initially.**..

You can install (or remove) additional Jenkins plugins at a

Later point in time via the **Manage Jenkins** > **Manage Plugins** page in Jenkins.

The wizard shows progression of Jenkins being configured and chosen set of Jenkins plugins being installed. This process may take a few minutes.

7.5.2.3 CREATING THE FIRST ADMINISTRATOR USER

Finally, after customizing Jenkins with plugins, Jenkins asks to create first administrator user.

1. When "**Create First Admin User"** page appears, in the respective fields then click **Save and Finish**.
2. When the **Jenkins is ready** page appears, click **Start using Jenkins**.

   Notes:
   - This page may indicate **Jenkins is almost ready!** Instead and if so, click **Restart**.
   - If the page does not automatically refresh after a minute, use your web browser to refresh the page manually.
3. If necessary, log in to Jenkins with the credentials of the user, just created and ready to start using Jenkins!

From this point on, the Jenkins UI is only

Accessible by providing valid username and password credentials.

7.5.3 INITIAL CONFIGURATION:

a) Jenkins installs as a service. Hence it always restart as soon as system reboots for any update. No need to start manually. Just need to access with IP: Port combination with login credentials.

b) Is somehow Jenkins is stopped then same may be restarted by typing following in browser

   a. Type in browser IP:port/restart,

   b. Or alternatively start services.msc (start->Run) and restart Jenkins as following.

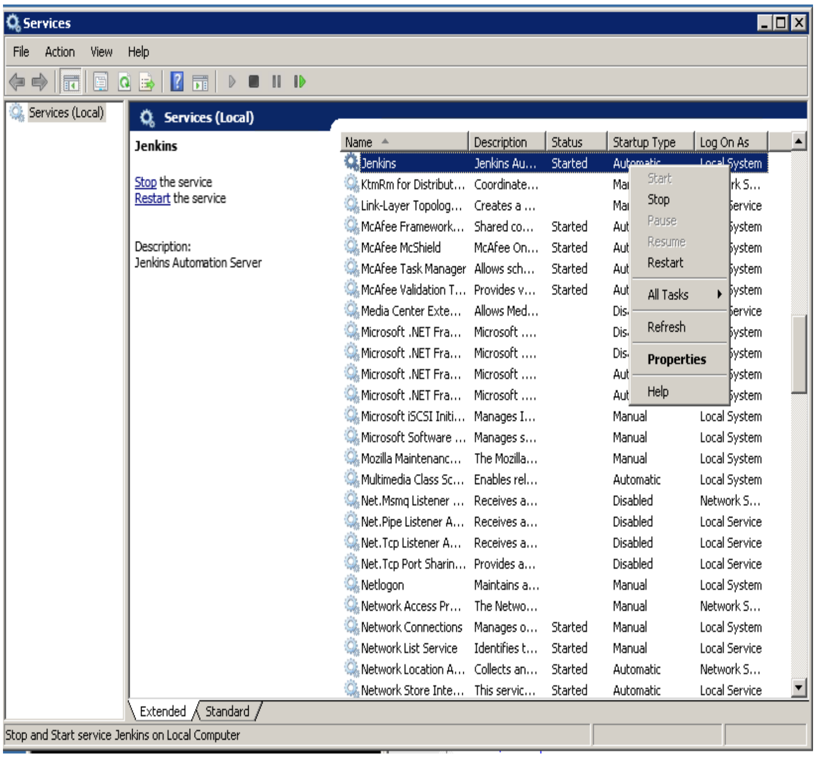


Figure 7.11: Starting Jenkins Service

After creating first Administrator Account, any number of users can be added to Jenkins with credentials as following.

A) Go to Manage Jenkins -> Manage Users

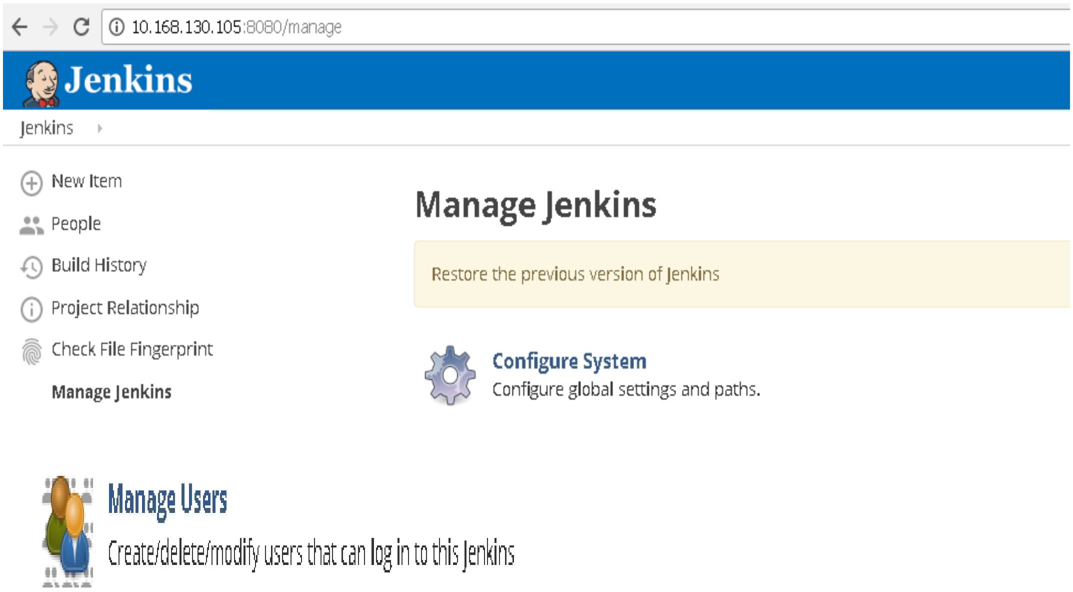


Figure 7.12: Manage Jenkins/Users

b) Click on Manage Users and Then Create User

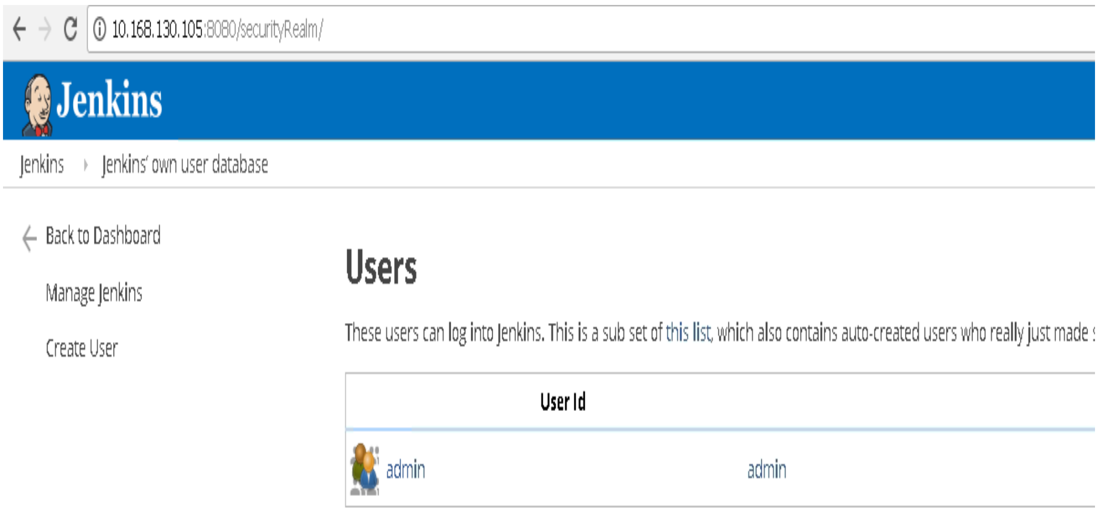


Figure 7.13: Jenkins Users Database

c) Fill the details and click on create user. As soon as user is created, then same credentials can be used to login with Jenkins.

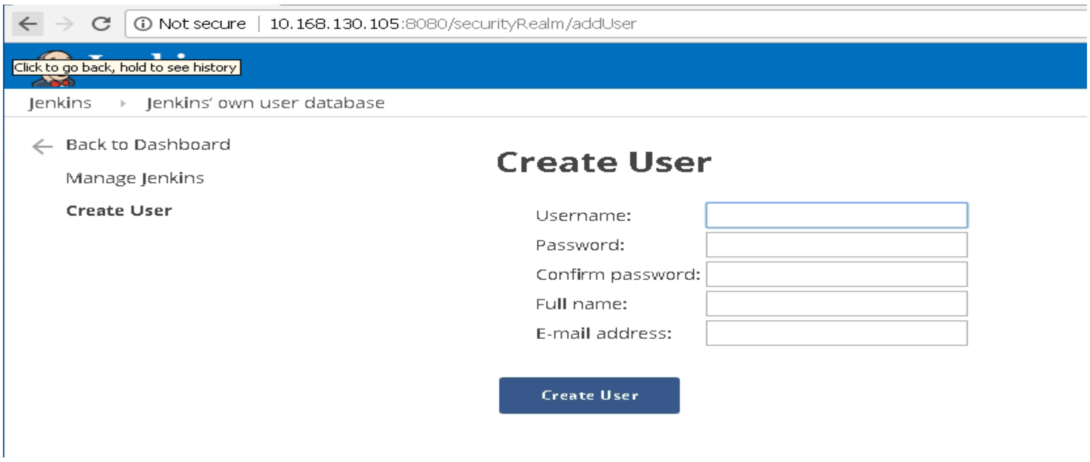


Figure 7.14: Create Jenkins Users

7.5.4 CREATING SAMPLE PROJECT

➔ **Go to Jenkins home page and click on New Item**

Figure 7.15: Jenkins New Item

➔ Select Freestyle Project and Enter Project Name

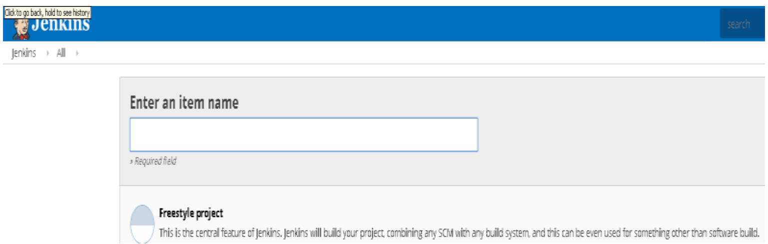


Figure 7.16: Jenkins Project Name

➔ Click Ok and a Project Configuration Wizard will open.

7.5.4 INSTALL AND CONFIGURE PLUG-INS

FROM THE WEB UI:

The simplest and most common way of installing plugins is through the

Manage Jenkins > **Manage Plugins**, available to administrators of a Jenkins environment.

Under the **Available** tab, plugins available can be downloaded.

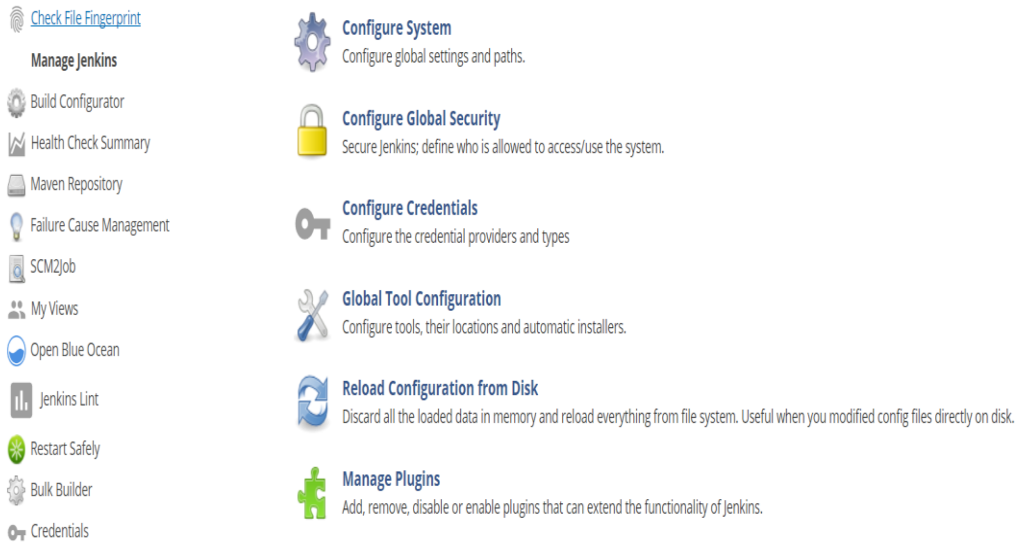


Figure 7.17: Manage Plugins

Click on Manage Plug-In and search for the plugin you need to install.

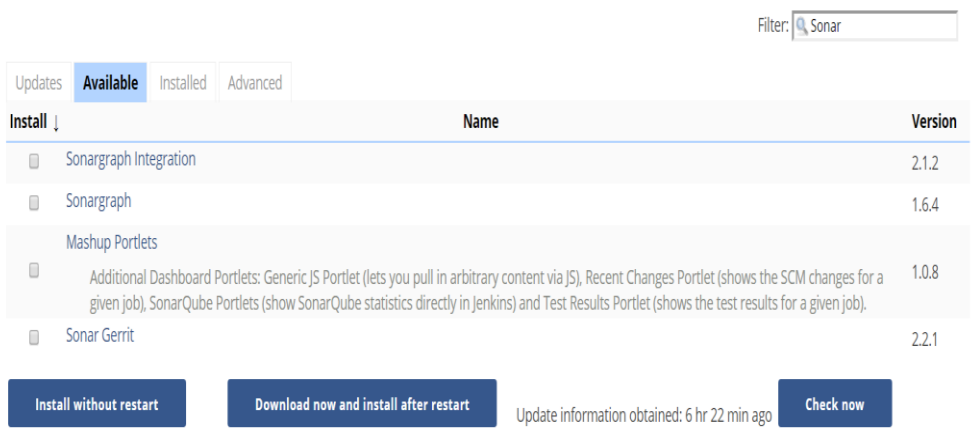


Figure 7.18: Search available plugins

Select the Plugin and choose either Install without restart or Download now and install after restart.

ADVANCED INSTALLATION:

Click on Advanced Tab, You may upload any downloaded .hip (plugin file) and upload annually via below mentioned method.

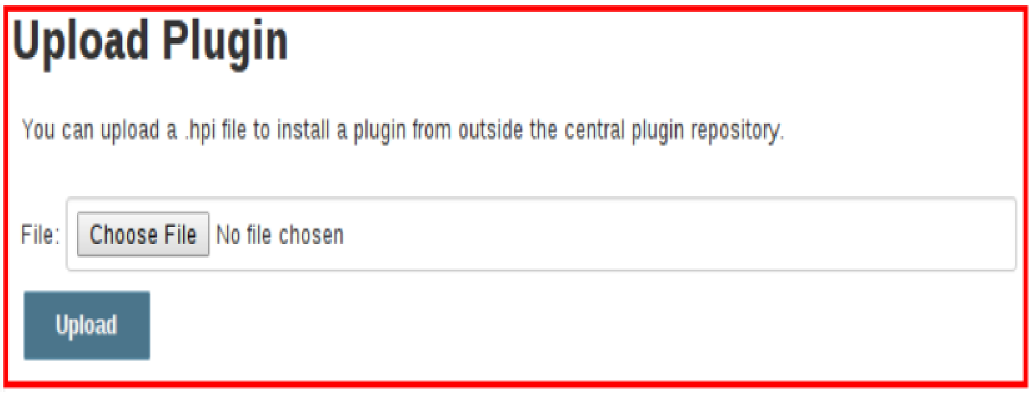


Figure 7.19: Upload plugin

All plugins can be found and browse from https://plugins.jenkins.io/

Now Following Plugins need to be installed by above mentioned methods

LIST OF RECOMMENDED PLUGINS

1) Get Plugin https://wiki.jenkins.io/display/JENKINS/Git+Plugin
2) SonarQube plugin
3) SonarQube Scanner
4) Fire Line: is a static code analysis program to look for bugs in Java Code.
5) Sonar Quality Gates: If done setup, it fails the build whenever Quality Gate criteria in sonar analysis aren't meet.
6) Junit Plugin: Allows Junit-format test results to be published
7) Grade Plugin, used to invoke a Grade build script as the main build step.

   Configured under Manage Jenkins → Global Tool Configuration

   Dependency: struts (version: 1.3)
8) Apache Maven comes by default with standard Jenkins installation
9) MS Build Plugin: allows MS Build to build .NET Projects

   https://wiki.jenkins.io/display/JENKINS/MSBuild+Plugin

   Dependency: Struts (1.3)
10) Notification: Default plugin comes with Jenkins but another plugin also available in market from Tikal Knowledge which allows sending Job Status notifications in JSON and XML formats
11) Distributed Build: A Slave computer can be configured to offload build projects from master computer. It's an inbuilt feature and can be done with manage Jenkins, where it asks the details of node slave machine.
    1) Workspace Cleanup plugin: Deletes workspace after a build is finished
    2) Windows slave plugin : connects to windows machine and start slave agents on them

7.5.5 Configure Plugins: Global Tools Configuration:

Check Jenkins → Manage Jenkins → Global Tool Configuration

Figure 7.20: Manage Jenkins

1. Keep Maven Configuration as it is. “Default Maven Settings”

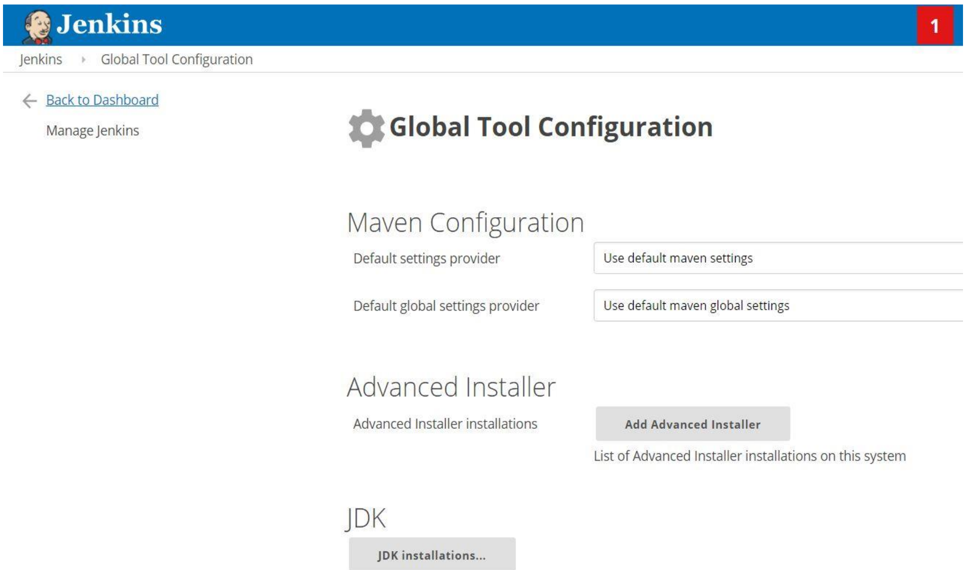


Figure 7.21: Global Tool Configuration

2. Configure JDK as shown in figure

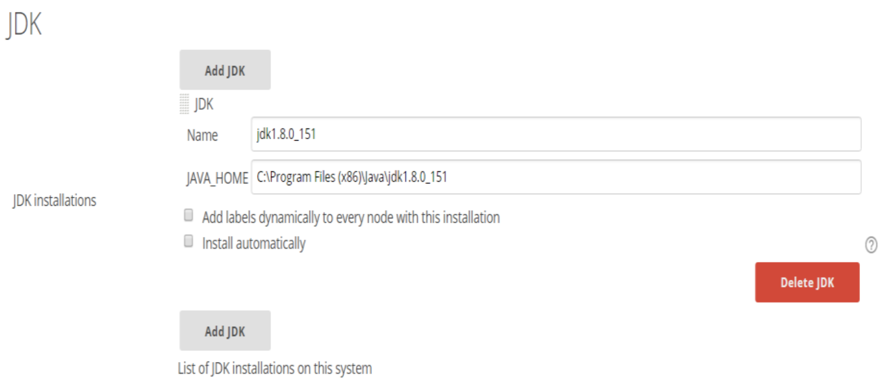


Figure 7.22: Configure JDK

3. Configure GIT as following.

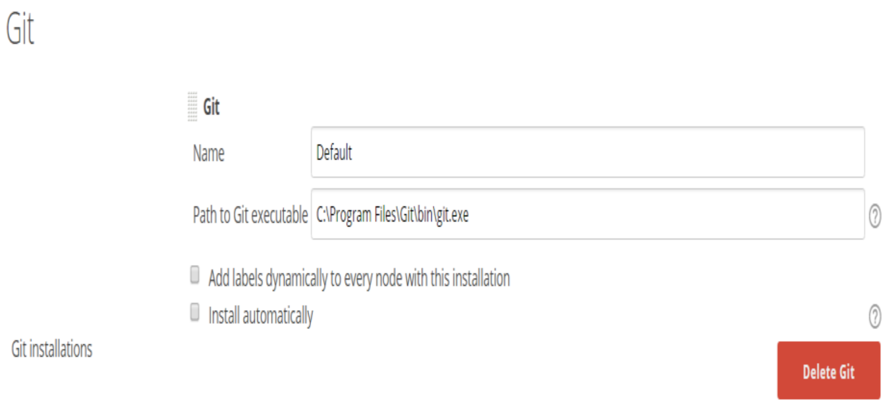


Figure 7.23: Configure GIT

4. Configure Grade as following.

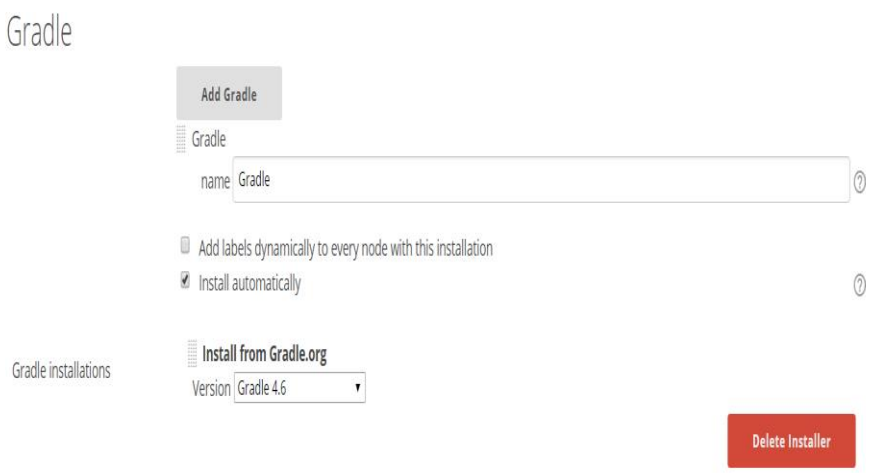


Figure 7.24: Configure Gradle

5. Configure MS Build as following.

Figure 7.25: Configure MSBuild

1) Here installation path may also exist on Slave nodes, May not necessary all on master.

6. Configure SoanrQube Scanner for MS Build. Select latest version from drop down list.

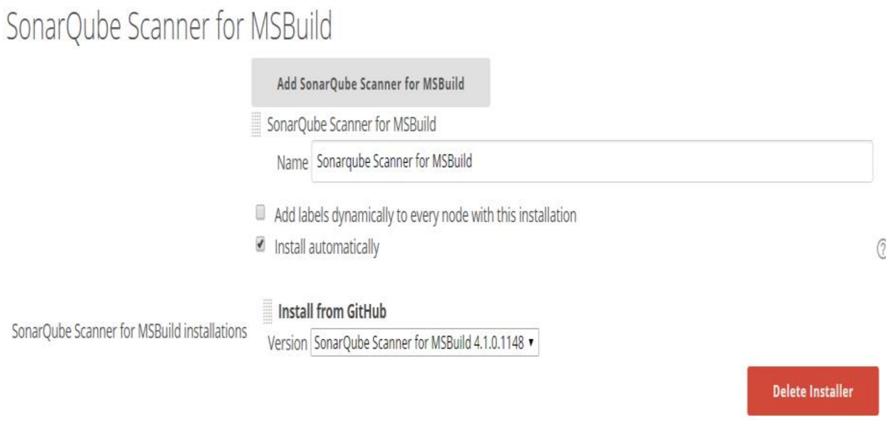


Figure 7.26: Add SonarQube Scanner for MSBuild

7. Configure SonarQube Scanner as following. Select Latest version from drop down

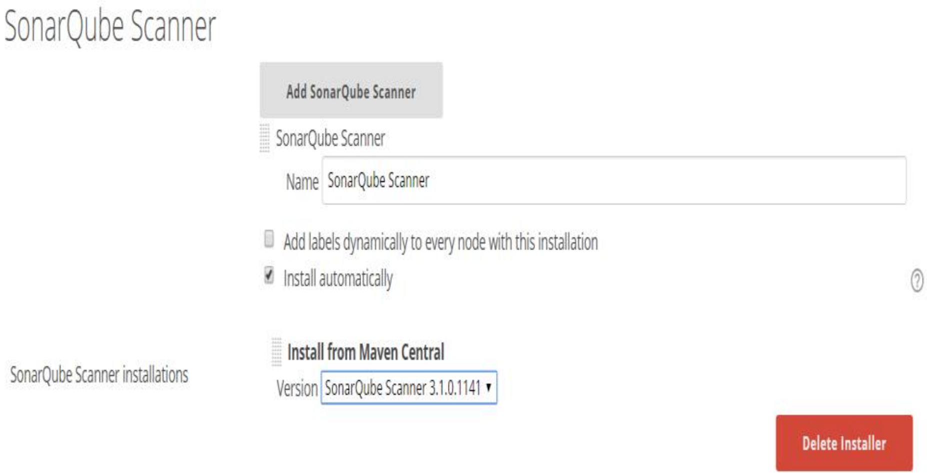


Figure 7.27: Add SonarQube Scanner

8. Configure ANT as following. Select latest version from dropdown list.

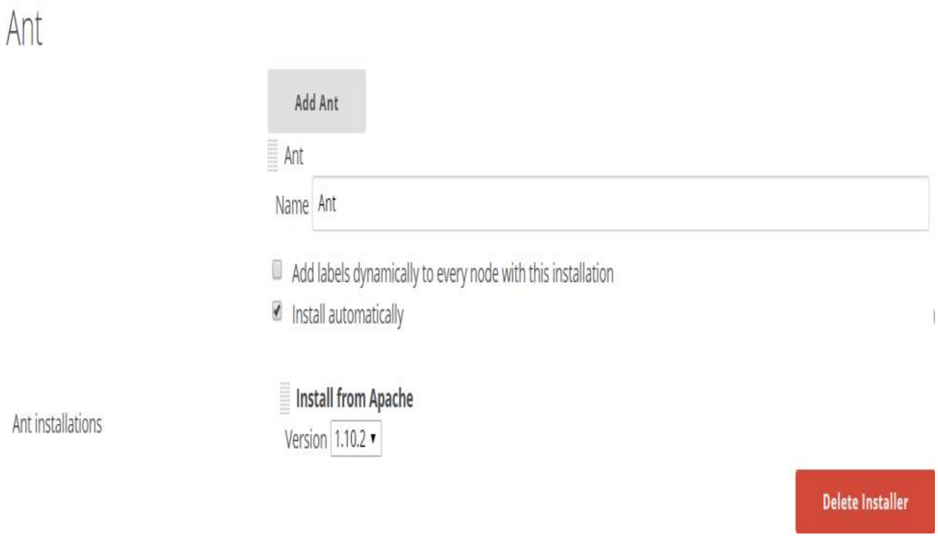


Figure 7.28: Configure ANT

9. Configure MAVEN as following. Select latest version from dropdown list.

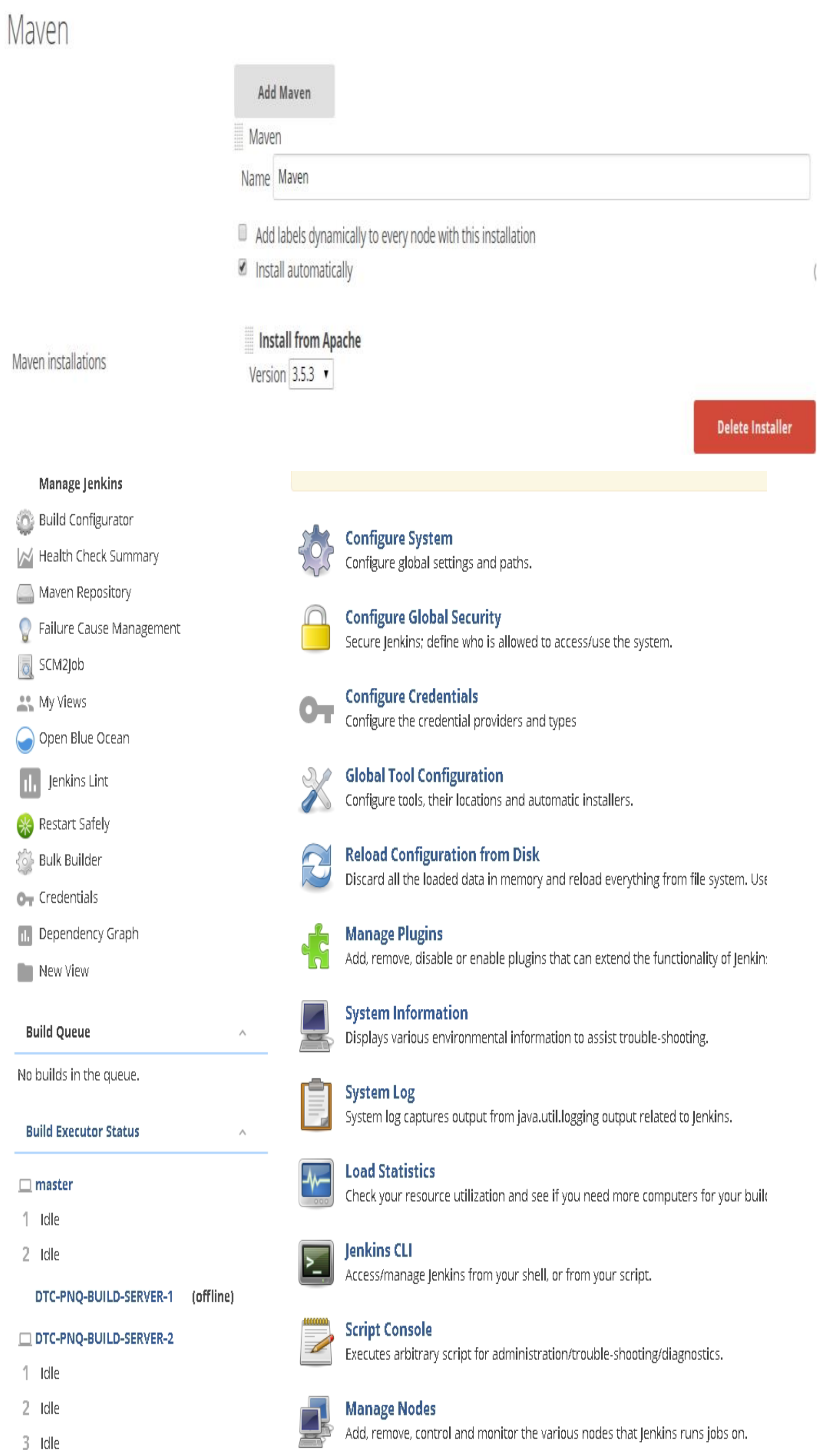
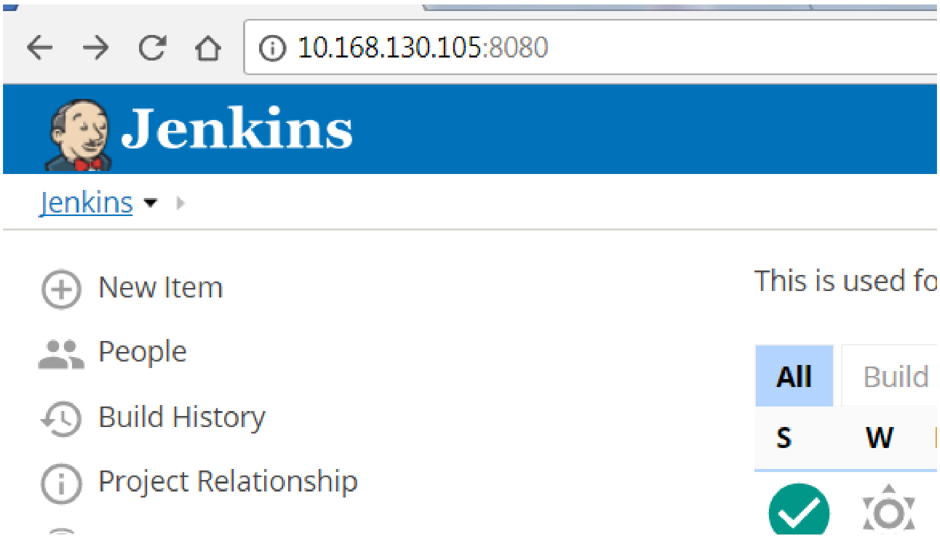


Figure 7.29: Configure Maven

7.5.6 CONFIGURE GLOBAL SETTINGS AND PATHS.



Go to Manage Jenkins→ Configure System

Figure 7.30: Global Tool Configurations

1) Configure Maven Project Configuration as following.

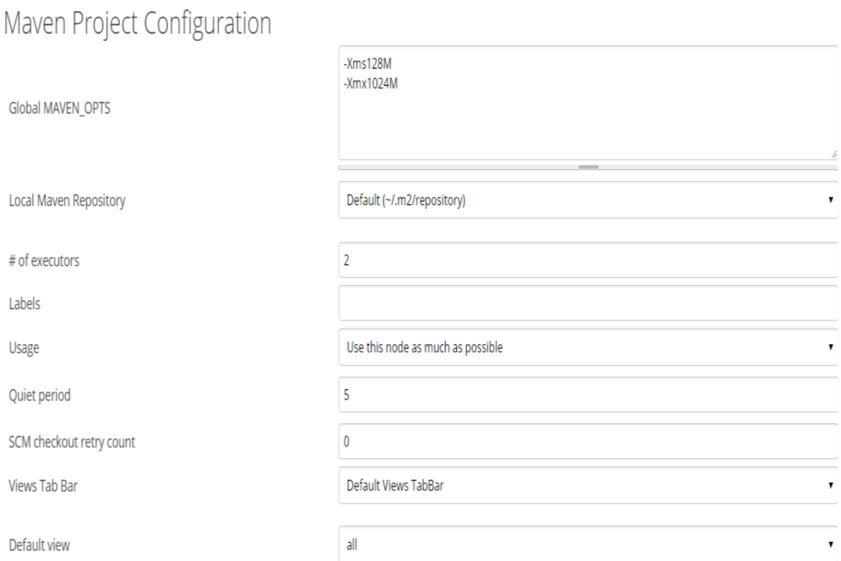


Figure 7.31: Maven Project Configuration

1. I have selected default view as all, but any specific view can be selected.

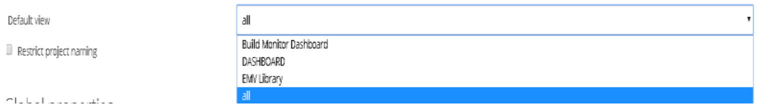


Figure 7.32: Default View

2) Configure setting for SonarQube server as following. Mention full URL: port instead of localhost.

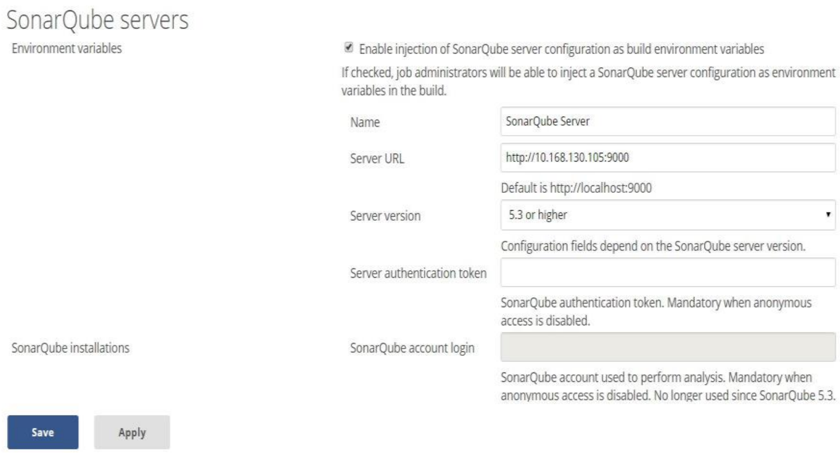


Figure 7.33: Configure SonarQube Servers

3) Configure Slave Status as following.

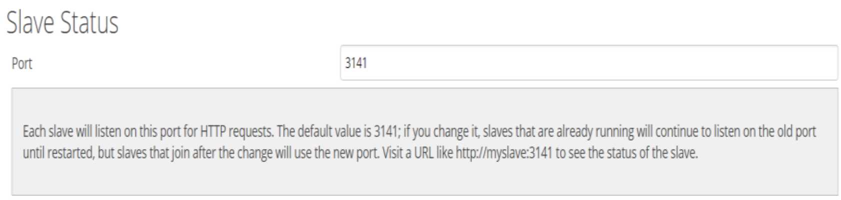


Figure 7.34: Slave Port Status

The status of slave machine can be shown as following.

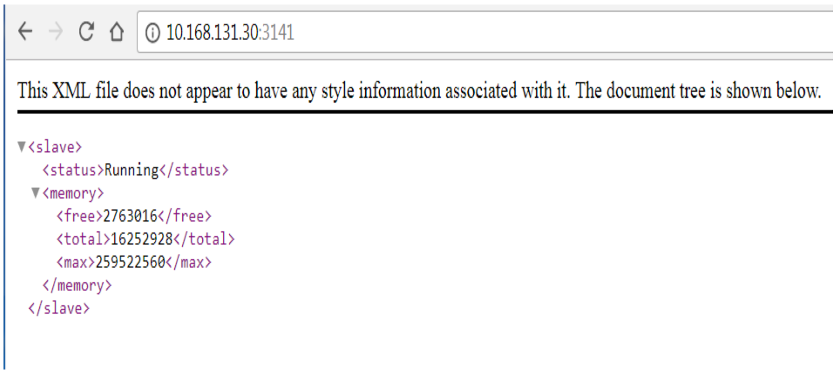


Figure 7.35: Slave Machine Status

4) Jenkins Location.

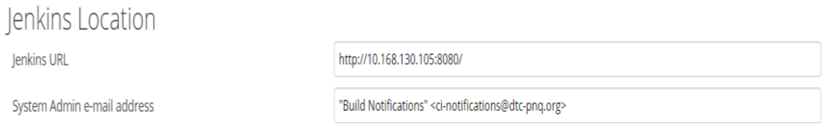


Figure 7.36: Jenkins Location

5) Configure Quality Gates - Sonarqube

Figure 7.37: Configure Sonarqube Quality Gates

6) Configure Get plugin user.name and user.email values.

7) Configure Extended Email configuration.

8) Configure Email Notification.

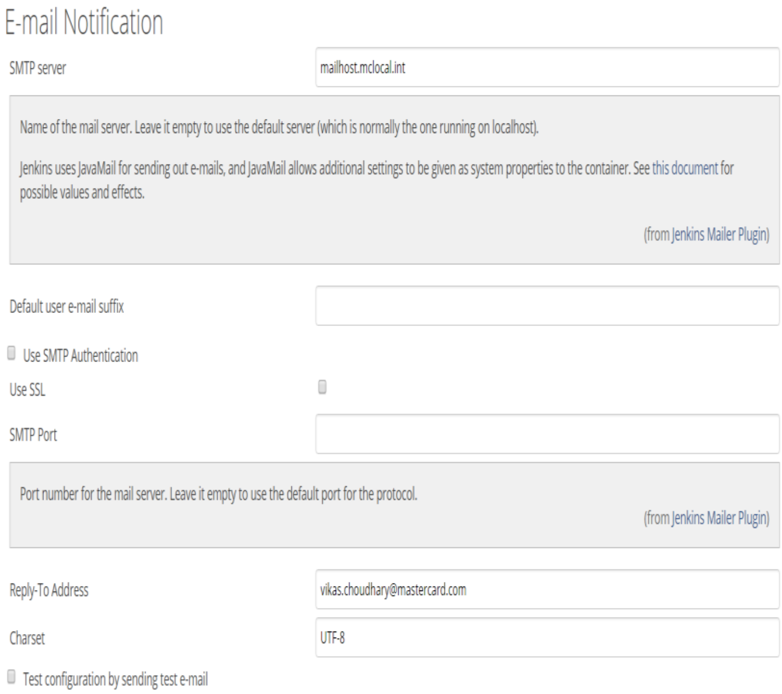


Figure 7.38: Configure E-mail notification

We need to configure Java Mail as same is used by Jenkins and it doesn't comes with default java installation. Always take latest version of Java Mail API.

Configuring Java Mail for Email notification

1. Update and configure latest Java Mail API, add jar file in classpath. Refer following documents
    1. http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-eeplat-419426.html#javamail-1.4.5-oth-JPR
    2. https://javaee.github.io/javamail/#overview_description

2. Extract Java Mail latest version under Installed Java. i.e. C:\Program Files (x86)\Java

1. Add Java Mail “mail.jar” into environment variable JAVAMAIL_JAR as following.

   JAVAMAIL_JAR= C:\Program Files (x86)\Java\javamail-1.4.5\mail.jar

2. Add CLASSPATH = %JAVAMAIL_JAR% into environment variable.

3. If Using Windows Platform, need to do settings in McAfee
   1. Right-click Access Protection and select Properties
   2. Click the Access Protection tab
   3. Under Categories on the left, select Anti-virus Standard Protection
   4. In the right pane, select Prevent mass mailing worms from sending mail, then click Edit
   5. In the Processes to exclude section, type the process name, then click OK to close the Rule details window
   6. Click Apply then close the Access Protection Properties window

4. Add following (processes to exclude) chrome.exe, eclipse.exe, java.exe, csrss.exe, jenkins.exe

9) Configure ANDROID_HOME as following

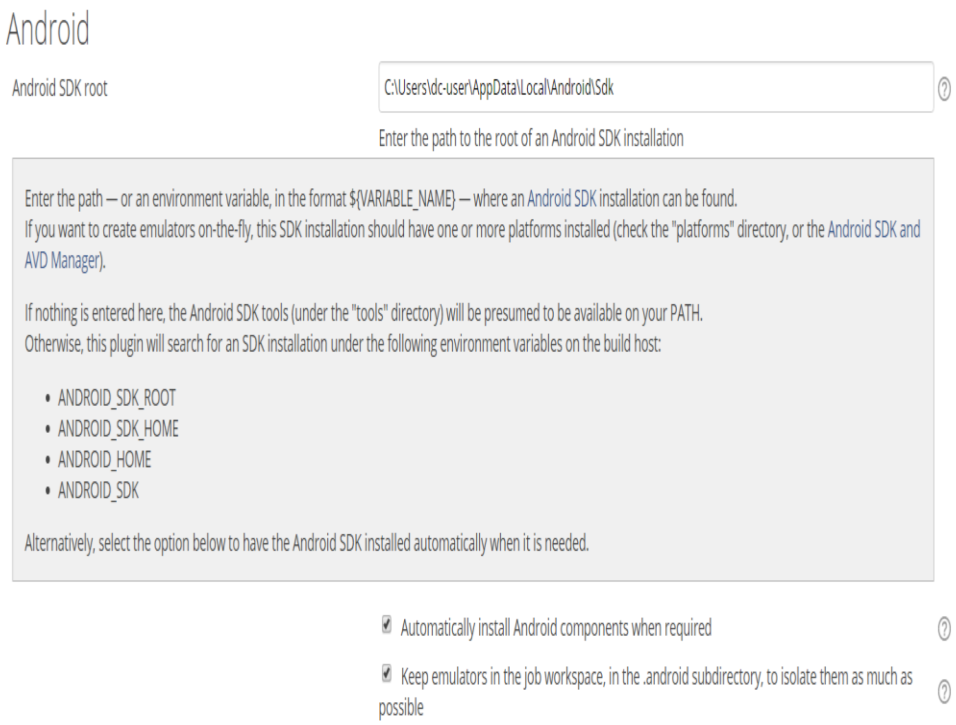


Figure 7.39: Configure Android SDK

7.6 SETTING UP PROJECT IN JENKINS

7.6.1 CONFIGURE PROJECT

Step 1 − Go to the Jenkins dashboard and Click on New Item

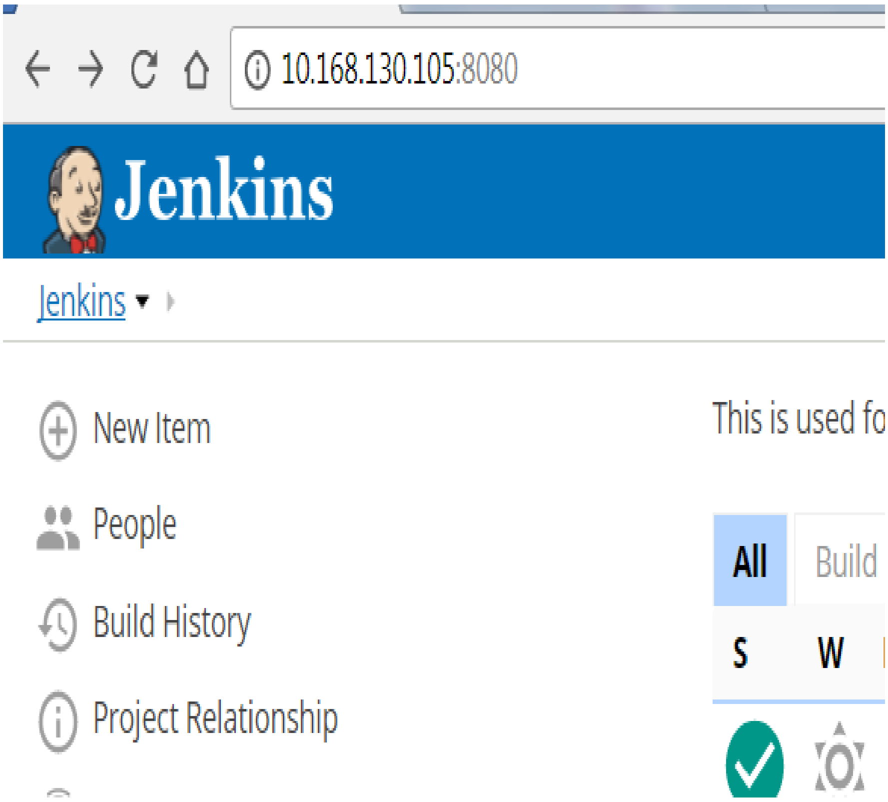


Figure 7.40: Automation Build Setting

Step 2 − from next screen, enter Item name, in this present case item named is HelloWorld. Choose the ‘Freestyle project option’

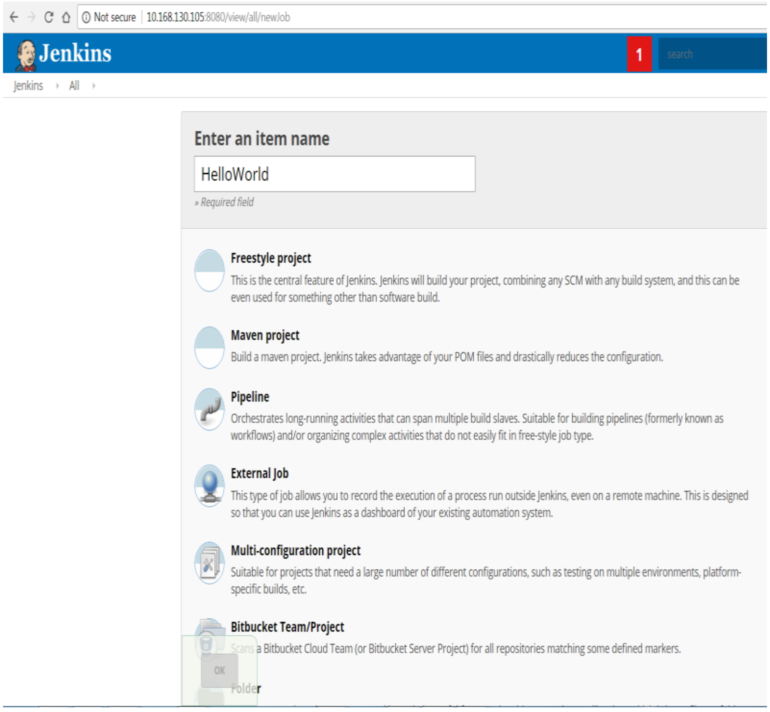


Figure 7.41: Configure Project Name

Step 3 − the following screen can specify the details of the job.

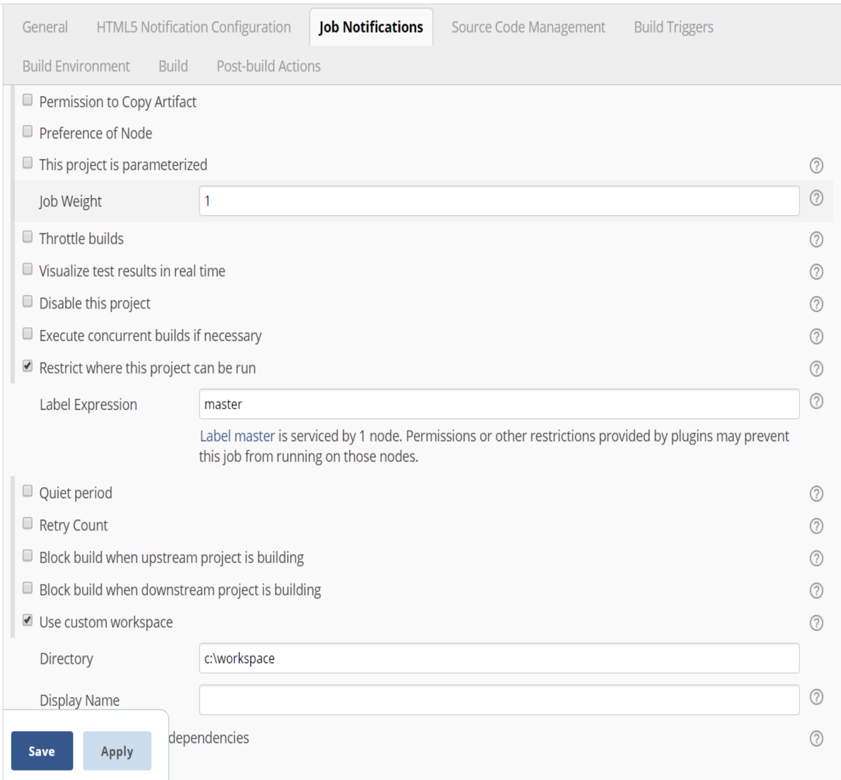


Figure 7.42: Configure job notifications

Give Label expression Name of Nodes (slaves), you may also specify custom workspace as mentioned.

Step 4 − we need to specify the location of files which need to be built. Enter the URL of that repository here. In addition to this, you would need to click on the Add button for the credentials to add a user name and password to the repository so that the code can be picked up from the remote repository.

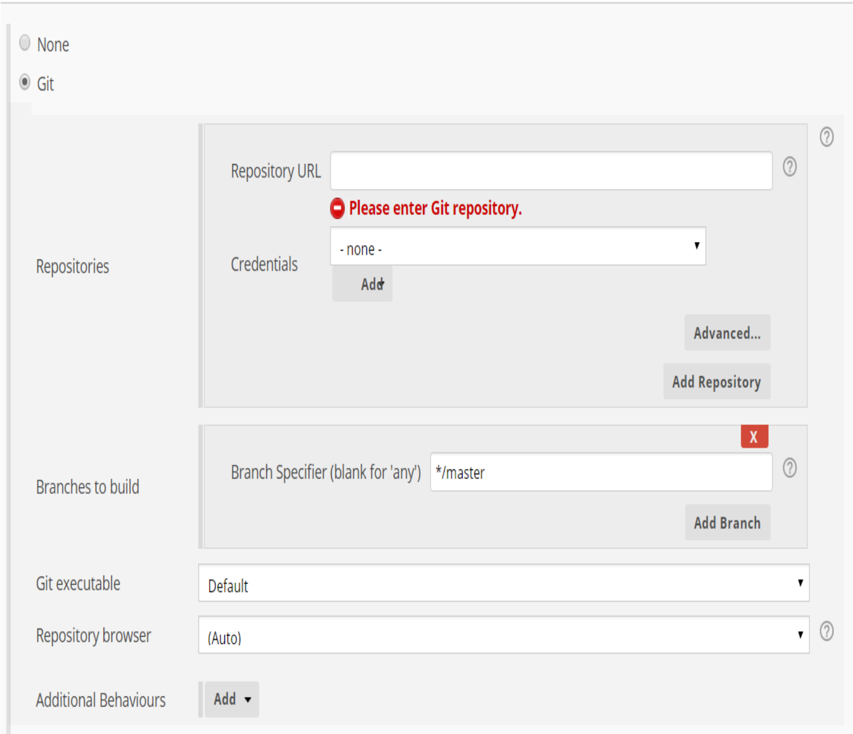


Figure 7.43: Configure GIT Repository URL

Step5: Configure Build Triggers

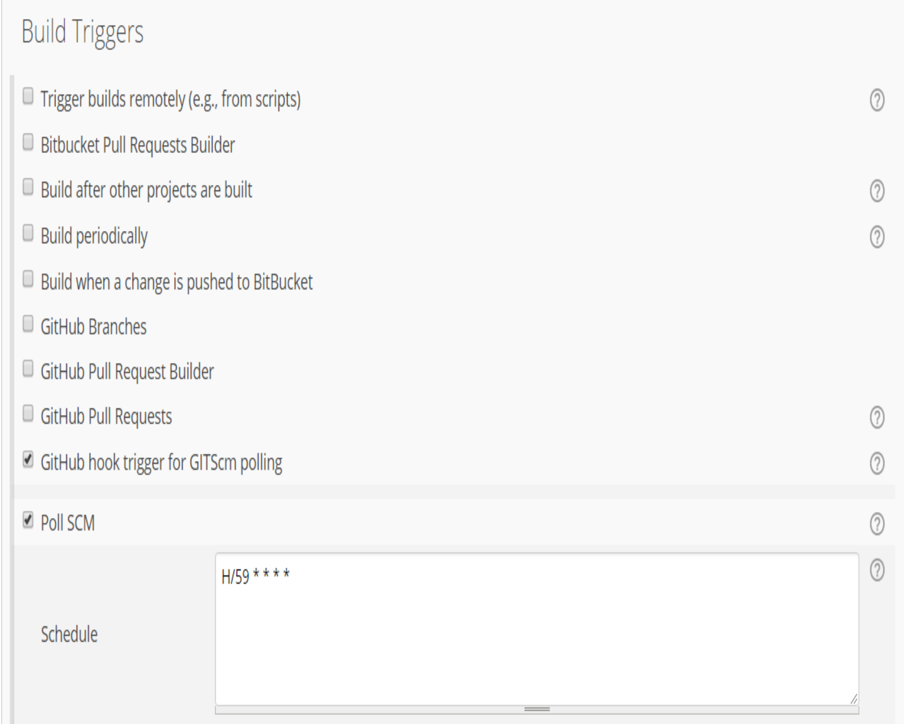


Figure 7.44: Configure Build Triggers

If Project have dependency on another project then select option "Build after other projects are built" as well.

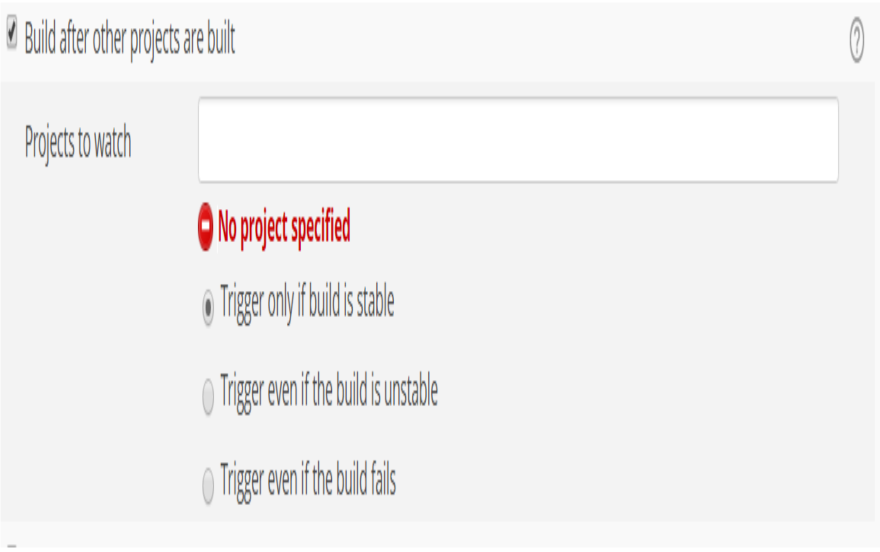


Figure 7.45: Build Triggers

Step 6 − Now go to the Build section and click on Add build step → Execute Windows batch command, You may Invoke Ant, Grade Script as per your project configuration.

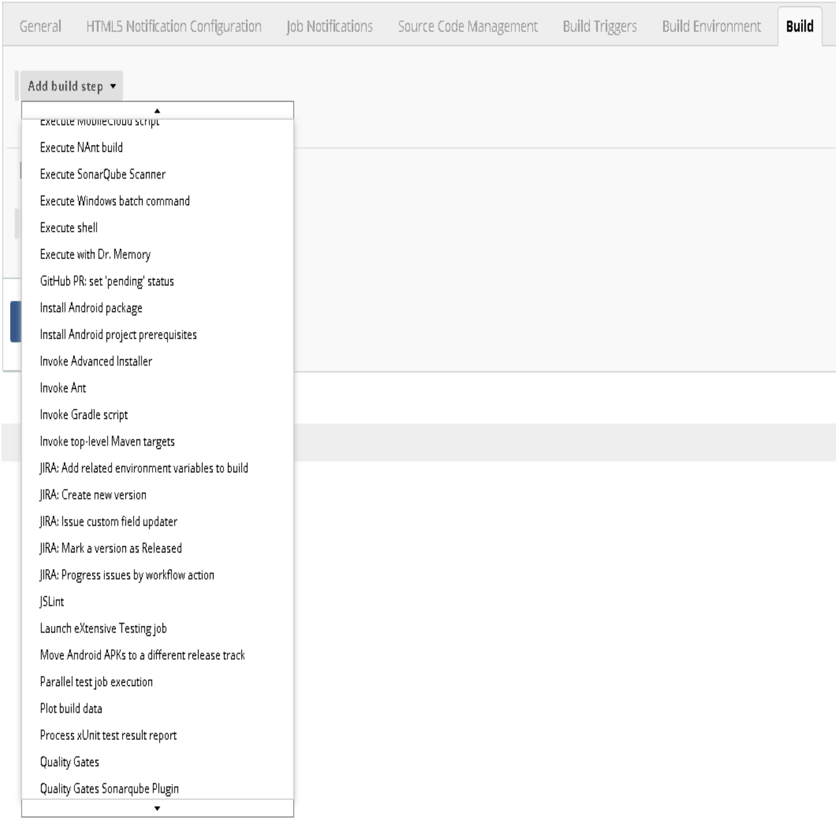


Figure 7.46: Add Build Step

Step 7 − in the command window, enter the following commands and then click on the Save button.

```
javac HelloWorld.java

Java HelloWorld
```

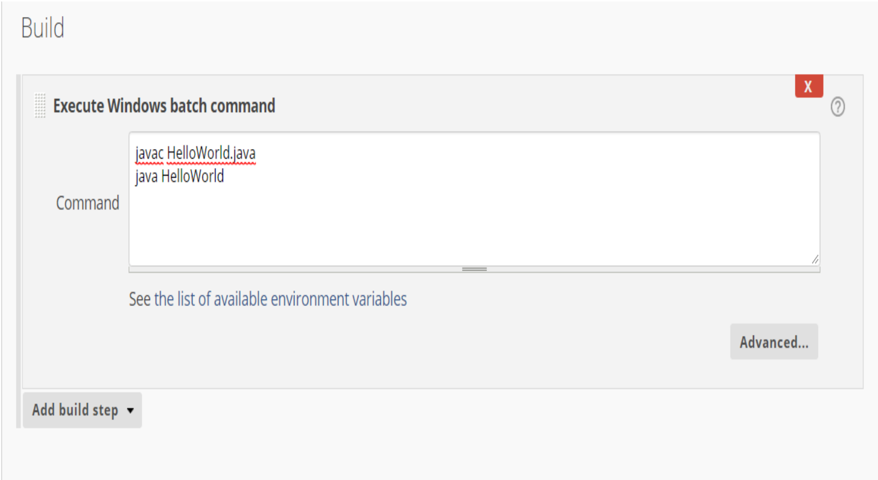


Figure 7.47: Configure Windows Batch Command

Alternatively->

Figure 7.48: Configure Root POM

Step 8 - Configure Sonar Scanner

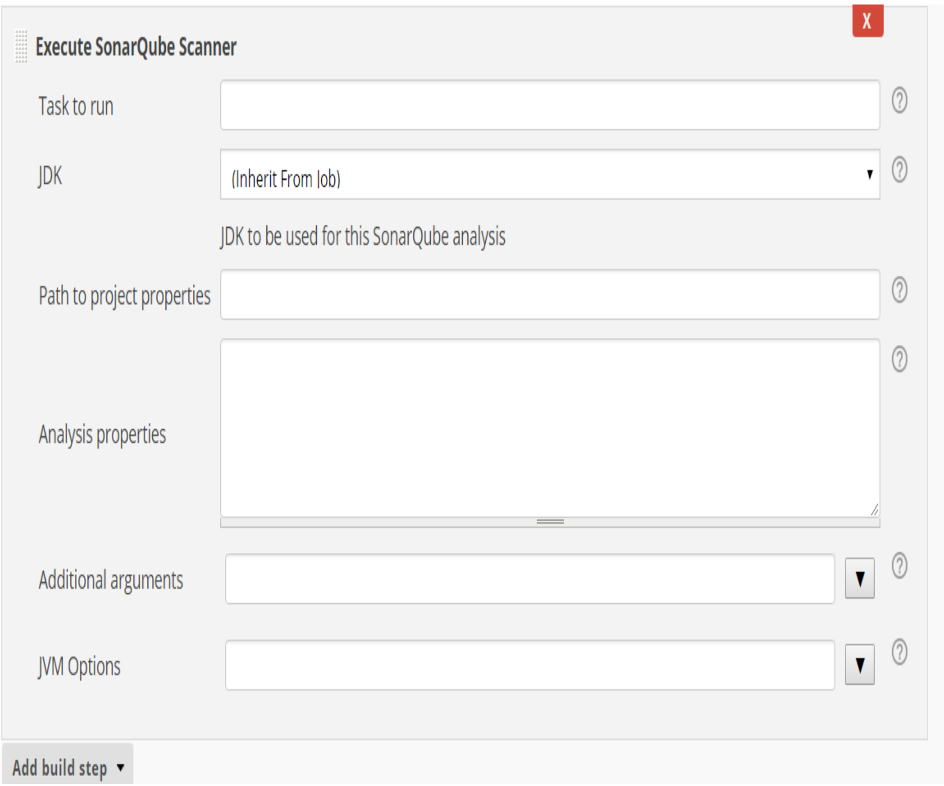


Figure 7.49: Configure Sonar Scanner Properties

Analysis properties-> can be written as following.

7.6.2 SAMPLE SONAR-PROJECT.PROPERTIES AS FOLLOWING

#Configure here general information about the environment, such as SonarQube DB details for example

#No information about specific project should appear here

#Default SonarQube server

sonar.host.url=http://0.0.0.0

```
#Default source code encoding
sonar.sourceEncoding=UTF-8

# required metadata
sonar.projectKey= HelloWorld

#Java project analyzed with the SonarQube Runner
sonar.projectName=HelloWorld
sonar.projectVersion=1.0

#Language
sonar.language=java

#java version used by source files:
sonar.java.source=1.8

#Comma-separated paths to directories with sources
(required)
sonar.sources=src/

#path to Java project compiled classes (optional)
sonar.java.binaries=bin/com
#sonar.java.binaries=target/classes

#comma-separated list of paths to libraries (optional
sonar.java.libraries=lib

#path to test source directories (optional)
#sonar.tests=testDir1, testDir2
```

Step 9 – Configure Editable Email Notification

Once saved, you can click on the Build Now option to see if you have successfully defined the job.

Step 10− Once the build is completed, a status of the build will show if the build was successful or not. In our case, the following build has been executed successfully. Click on the #1 in the Build history to bring up the details of the build.

Step 11 − Click on the Console Output link to see the details of the build

7.7 SETTING UP DASHBOARD

Visit

https://plugins.jenkins.io/build-monitor-plugin

Go to archive and download latest from

https://updates.jenkins.io/download/plugins/build-monitor-plugin/

Upload downloaded plugin (.hip file) and restart the Jenkins.

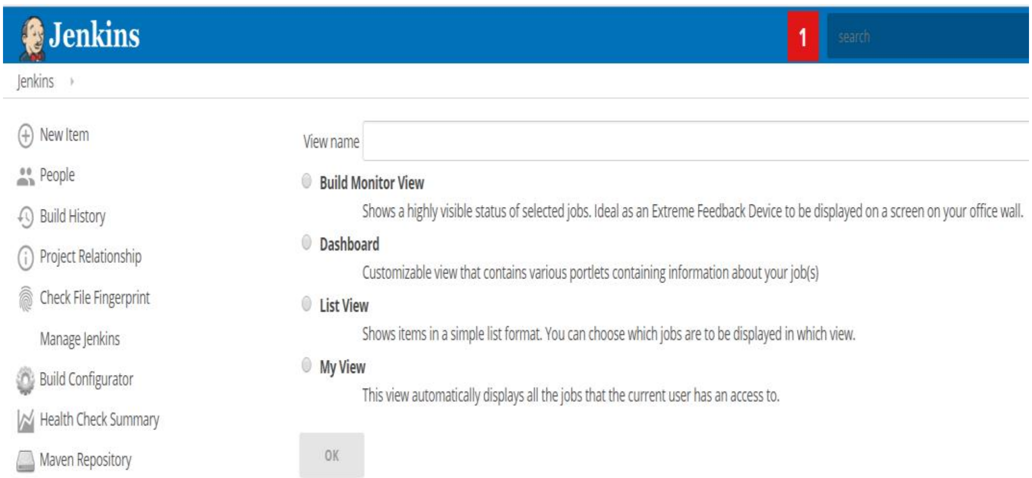


Figure 7.50: Uploading plugins

Now you may select the Build Monitor View to see the Status of Builds.

Give the Name Sample and Configure the Projects included.

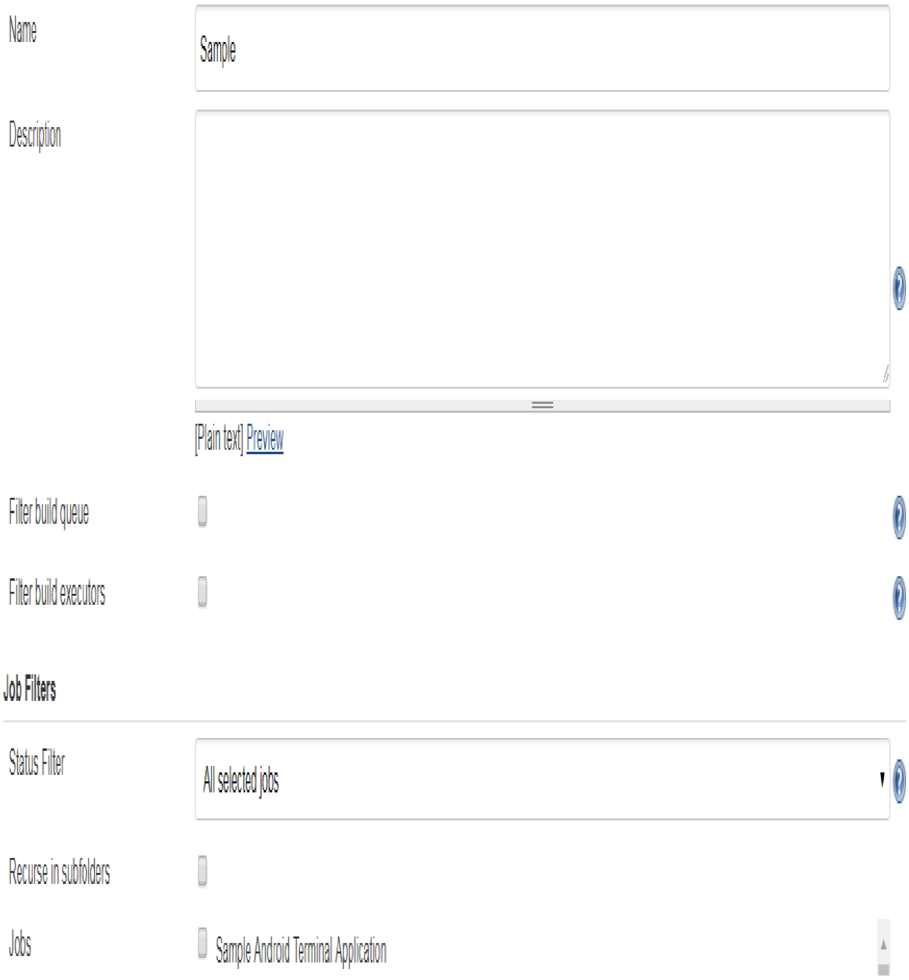


Figure 7.51: Creating Sample Projects

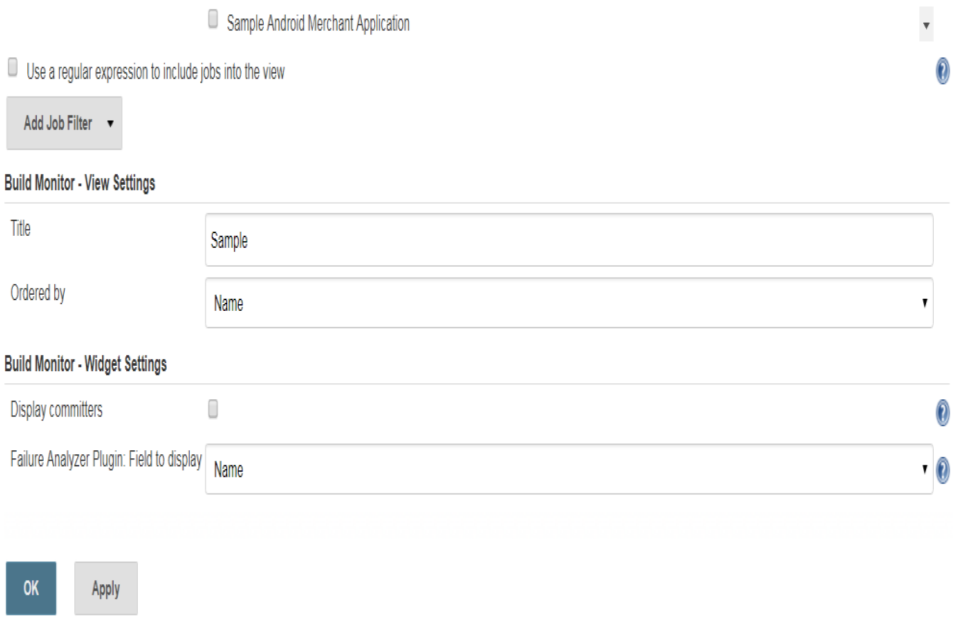


Figure 7.52: Configure Build Monitor

7.8 DISTRIBUTED EXECUTION (MASTER AND SLAVE NODE)

Note: Prepare VM or Physical Machines have same kind of environment configuration and software installed. Bur additional software may be installed on slave nodes for additional task but there is no need to install Jenkins on slave node.

7.8.1 CONFIGURE MASTER MACHINE

1. On Master Machine go to Manage Jenkins -> Manage Nodes

Figure 7.53: Manage Nodes

2. Click on New Node

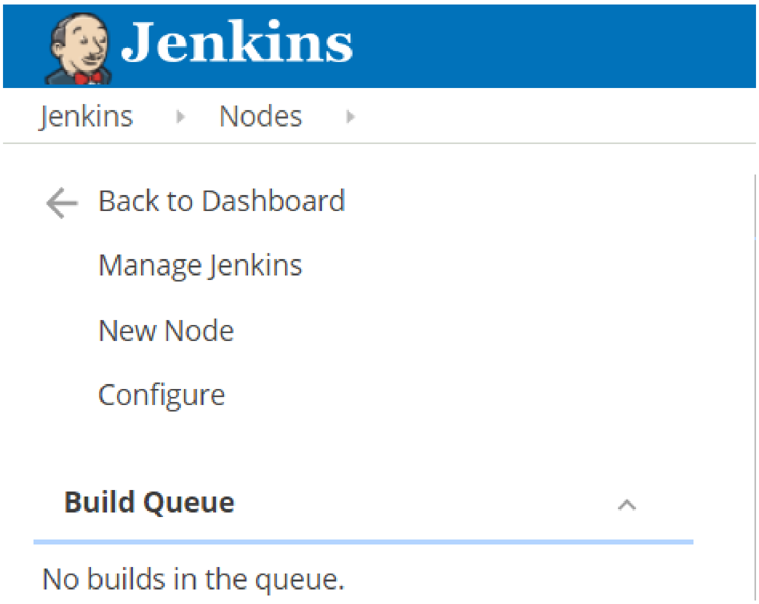


Figure 7.54: Configure New Node

3. Enter Node Name and Select Permanent Agent then press OK.

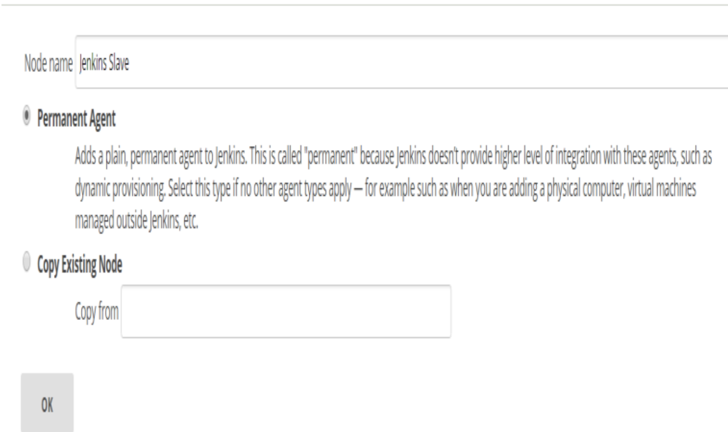


Figure 7.55: Select Agents

4. Fill the information as following.
    1. Set a **number of executors**
        1. (One or more) as needed.
    2. Set a **Remote FS Root**
        1. A home directory for the master on the slave machine.
        2. For a *Windows slave*, use something like: "C:\Jenkins\"

3. Select the appropriate **Usage** setting:
    1. For an additional worker: *Utilize this slave as much as possible*
    2. For specialized jobs: *Leave this machine for tied jobs only*
4. **Launch Method**:
    1. An easy way to control a Windows slave is by using *Launch slave agents via Java Web Start* (Recommended for Windows)
    2. TODO: add steps for other methods.
5. **Availability**
    1. *Keep this slave online as much as possible*
    2. TODO: add details for each option.
6. Press **OK**.

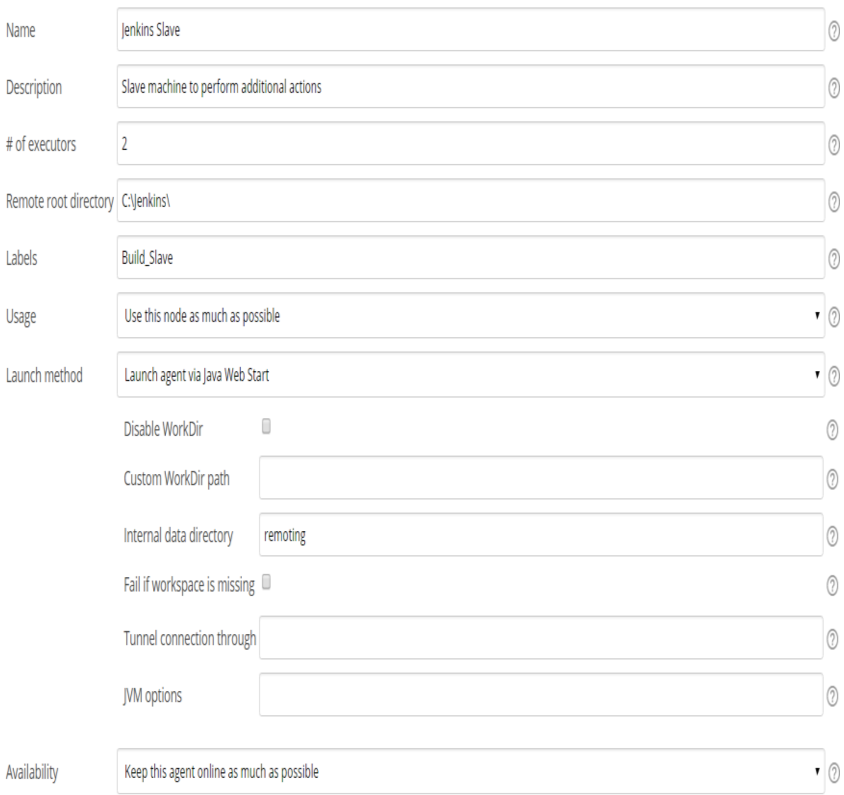


Figure 7.56: Configure Build Slave Machine

7.8.2 CONFIGURE SLAVE MACHINE

a. Open a browser on the **slave machine** and go to the **Jenkins master server** URL (http://yourjenkinsmaster:8080).
b. Go to **Manage Jenkins** > **Manage Nodes**,

   Click on the newly created slave machine.

   Click on the **Launch** button to launch agent from browser on slave.

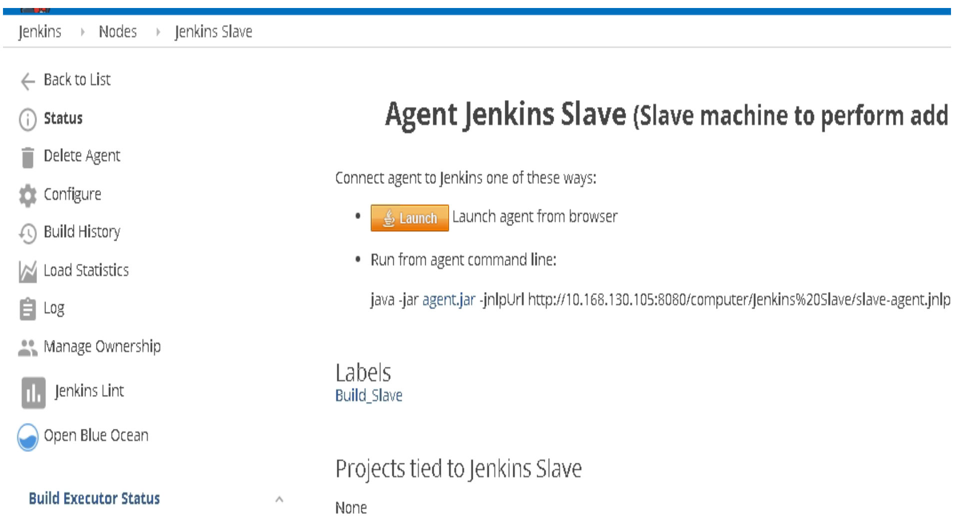


Figure 7.57: Launch Agent Jenkins Slave

Agent will be downloaded and then you may start the same

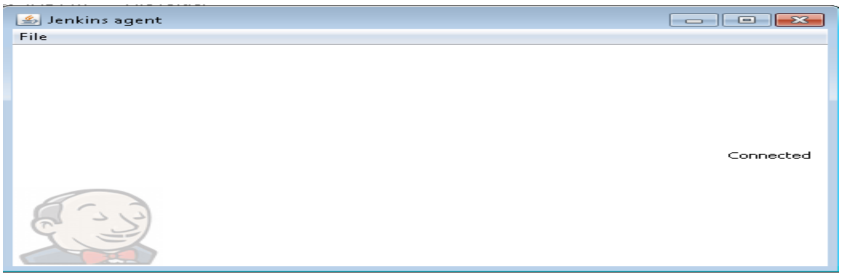


Figure 7.58: Jenkins agent launch

7.9 ADVANTAGES OF AUTOMATED BUILD/TEST SYSTEM

Continuous build is a necessary condition for continuous testing

It helps to improve product quality

Minimize the chances of build failures at the time of delivery

Eliminate dependency of manual build and Unit Testing Process

Keep track of History of Builds

Produces consistent builds with clean build for every check-in

Test Status and results can be presented on dash board.

Automated Regression and Functional Test Suite can be executed with/for every Build

Multiple execution of build process Load can be shared with Master and Slave configuration

CHAPTER 8

RUN TIME DYNAMIC ADAPTIVE AUTOMATION TESTING TOOLS

CHAPTER 8
RUN TIME DYNAMIC ADAPTIVE AUTOMATION TESTING TOOLS

Key Words

Web Testing, Web Driver, Mobile Testing, Browser Testing, Automation Framework, Python, Java, Performance Testing, Functional Testing, Unit Testing, Integration Testing, System Testing, Acceptance Testing

8.1 INTRODUCTION

These are some tools and technologies available to achieve results on run time dynamic adaptive automation testing (RTDAA) with cloud storage networks. These automation tools has specific features to offer in addressing the growing challenges of software automation in the years ahead. They provide capabilities for continuous testing and integration, management, and reporting. These tools supports continuous increasing automation needs for Web and Mobile testing. Intelligent verification and smart analytics for adaptive and heterogeneous environments are to be desired for automation tools and technologies. We will discuss about Open Source/Free and Commercial Tools and methods to choose right tools.

8.2 LIST OF RTDAA TOOLS

- Selenium
- TestComplete
- QMetry Automation Studio
- HP QTP/UFT
- Testim.io
- HP Quality Center (HP ALM)
- TestComplete
- Test Studio
- Katalon Studio

- IBM Rational Functional Tester
- Ranorex
- Appium
- Robotium
- Cucumber
- EggPlant
- SilkTest
- Watir
- Sauce labs
- Sahi Pro
- Sikuli
- IBM Performance Tester
- Apache JMeter
- BlazeMeter
- HP LoadRunner
- WAPT by SoftLogica
- NeoLoad
- Perfecto Mobile
- WebLOAD
- Test Anywhere
- Visual Studio Test Professional
- FitNesse
- TestingWhiz
- Tosca Testsuite
- WatiN
- SoapUI

8.3 OPEN SOURCE TOOLS

8.3.1 SELENIUM WEBDRIVER

Selenium is an open source library with bindings in multiple languages (Java, C#, Python, etc.) that allows an engineer to write code that is then translated into human-like interactions with various browsers and mobile devices. At its core Selenium spins up a lightweight server on a machine that sends commands in the JSON format to a browser or device. These JSON commands typically include information such as the action to be performed (click, enter text, submit form, etc.) as well as information about how to identify the element for the action to be performed on.

These pieces of identifying information are based on the Document Object Model (DOM) or a web page or app and, for all intents and purposes, can be thought of as the HTML of a page.

8.3.2 SELENIUM

It is test automation framework for Web applications. Selenium is an automation framework of choice for Web automation engineers, particularly for those who possess advanced programming and scripting skills. Selenium become a core framework for other open-source automation tools such as Katalon Studio, Watir, Protractor, and Robot Framework. Selenium supports multiple operating systems (Windows, Mac, and Linux) and multiple browsers (Chrome, Firefox, IE, and Headless browsers). And it can be programmed with scripts can be written in various programming languages such as Java, Groovy, Python, C#, PHP, Ruby, and Perl. Engineers have flexibility with Selenium, can write complex and advanced test scripts to meet various levels of complexity, it requires advanced programming skills and effort to build automation frameworks and libraries for specific testing needs.

It is the most popular automation testing tool for web applications. Selenium can be run in multiple browsers and operating systems**.** It is compatible with other automation testing frameworks.

With selenium, a browser-centred automation test scripts can be created which are scalable across different environments. Scripts can be created using Selenium that is of great help for prompt reproduction of bugs, regression testing, and exploratory testing.

8.3.3KATALON STUDIO

This is a powerful test automation solution for web application, mobile, and web services. This is built on top of the Selenium and Appium frameworks. This supports different levels of testing skills. Manual Testing Engineers can find it easy to start an automation testing, while automation engineers can save time from building new libraries and maintaining their scripts. Katalon Studio can be integrated into CI/CD processes and works with tools in the QA process including qTest, JIRA, Jenkins, and Git. It offers a feature called Katalon Analytics which provides users comprehensive views of test execution reports via dashboard including metrics, charts, and graphs.

8.3.4 WATIR

It is an open-source testing tool for web automation testing based on Ruby libraries. It supports cross browser testing including Firefox, Opera, headless browser, and internet explorer. It supports data-driven testing and integrates with BBD tools Cucumber, and Test/Unit.

8.3.5 ROBOT FRAMEWORK

Robot Framework implements the keyword-driven approach for acceptance testing & acceptance test-driven development (ATDD). Test capability can be extended by implementing additional test libraries using Python and Java. Selenium WebDriver is an external library in Robot Framework. Test engineers can leverage Robot Framework as an automation framework for web testing as well as for Android and iOS test automation. Robot

Framework can be easy to learn for engineers who are familiar with keyword-driven testing.

8.3.6 JBEHAVE

JBehave is an open source BDD (Behaviour Driven Development) library that allows users to write their test cases in plain English and have them automatically translated into chunks of Java code to be executed.

JBehave allows someone like a product owner or scrum master to write test cases, hand them off to automation engineers and have those engineers write the automation scripts.

JBehave also creates easily digestible and human readable reports after execution, including information such as what test cases were run, how many test cases passed/failed and provides screenshots for any failed test cases.

Everything in JBehave is customizable and flexible, giving each team the power to define their own test runs and even create custom reports.

8.3.7 RESTASSURED

RestAssured is an open source framework that allows for easy and flexible testing of API based applications.

8.3.8 PHANTOMJS

An extension of Selenium WebDriver that allows users to run tests on their local machine in a headless state, meaning they do not have to have a particular browser installed to run a test against it.

8.3.9 DOCKER

Docker is an open source tool that allows users to "containerize" applications and environments.

Using a simple Docker image and 1-2 commands a user can instantly deploy an environment on their local machine with a set of predefined conditions such as installed browsers with specific versions, specific applications installed or preconfigured network settings.

8.3.10 TESTNG

TestNG is a lightweight testing framework in between JUnit and JBehave/Cucumber. TestNG is ideal for teams that don't want to deal with the overhead of configuring BDD frameworks or are writing tests (such as API level) that do not lend themselves to BDD concepts such as stories or features.

8.3.11 CUCUMBER

Cucumber is another commonly used BDD library. It is very similar to JBehave but will be more familiar to those coming from a non-Java coding background.
It is designed over the concept of BDD (Behaviour-driven development). It performs the automated acceptance testing by running the stories that best describe the behaviour of the application. It gets a single up-to-date living document that is having both specification and test documentation.
Cucumber is scripted in Ruby. It also supports Java and.NET. It also has cross-platform OS support.

8.3.12 SIKULI

It works on image recognition and has the capability of automating anything that is seen on the screen. Currently, it supports desktop apps which run on windows, Mac or Unix/Linux. This tool is good at reproducing bugs and its users have reported it to be very useful as compared to other tools when are going to automate an application which is not web-based.

8.3.13 APACHE JMETER

It is Java desktop application designed for load testing. It mainly focuses on web applications. This tool can be used for unit testing and limited functional testing.

Its architecture is centred on plugins with the help of which JMeter provides a lot of out of box features. It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, Java objects, and database. It also include Test IDE, dynamic reporting, command line mode, portability, multithreading, caching of test results and highly extensible core. It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, Java objects, and database.

8.3.14 BLAZEMETER

You can easily create load and performance tests. It is compatible with JMeter tool. JMeter tests compatible with BlazeMeter as well. With BlazeMeter, API tests can be setup easily, to do user interactive website testing, perform scalable load testing using virtual user traffic and do a lot more. This tool supports native as well as mobile web apps.

8.3.15 APPIUM

This Test automation framework is intended for mobile applications, and automation of native, hybrid and mobile web applications built for iOS and Android. This tool uses vendor-provided automation frameworks and is based on client/server architecture. Appium is easy to install and use.

8.3.16 ROBOTIUM

It is an open-source test automation framework primarily meant for Android UI testing? It supports both native and hybrid applications. Using Robotium, time-saving, readable and easy to use automated grey box UI tests intended for android apps can be written. System testing, functional testing, and user acceptance testing over Android-based apps with the help of Robotium can be performed.

8.3.17 KATALON STUDIO

It is a powerful test automation solution for mobile, Web, and API testing it provides a comprehensive set of features for test automation, including recording actions, creating test cases, generating test scripts, running tests, reporting generated results, and integrating with other tools in the software development lifecycle.

Katalon Studio runs on **Windows** and **MacOS**, also supporting automated testing of **iOS** and **Android apps, web applications** on all modern browsers, and API services. This tools can be integrated with tools such as JIRA, qTest, Kobiton, Git, and Slack.

8.3.18 WATIR

It is an abbreviation for Web Application Testing in Ruby. It is a light-weight tool for automating web application testing. It supports web application regardless of considering over which technology, app is designed. With this, simple, flexible, readable and easily maintainable automated tests can be designed. Companies that use Watir includes SAP, Oracle, Facebook, etc.

8.3.19 WATIN

Name of this tool signifies Web Application Testing in .NET. This test automation framework supports IE & FF browsers, and for UI & functional testing of Web apps.

8.3.20 SOAPUI

This is a functional testing tool which provides complete API Test Automation Framework for SOAP and REST.

8.4 COMMERCIAL TOOLS

8.4.1 PERFECTO MOBILE

Perfecto Mobile is a third party cloud provider that gives "run time dynamic adaptive automation " users access to hundreds of real mobile devices to test against.

Any Organization has access to the Perfecto public cloud as well as a secure private cloud. The private cloud can be updated/customized as needed, giving us flexibility to test against specific device/os combinations from physical locations in any country like India, Australia and China.

Additionally, Perfecto supports a private repository where apps can be stored as ".apk" or ".ipa" files and dynamically loaded onto a device at runtime and cleared after execution.

8.4.2 BROWSERSTACK

BrowserStack is a third party cloud provider that gives RTDAA users access to thousands of OS/Browser/Version combinations.

When a user makes a request to BrowserStack they provide an OS (Windows XP - 10, OSX Snow Leopard - El Capitan), browser (Chrome, Firefox, Safari, Opera, Yandex, IE, Edge) and a browser version.

BrowserStack spins up a Virtual Machine (VM) in their private cloud with these specifications and the automation scripts are executed against them as if they were the users local desktop.

This allows Mac users to test on IE, Windows 10 users to test old versions of Chrome, etc.

8.4.3 APPLITOOLS EYES

AppliTools Eyes is a visual automation tool allowing users to test the look and feel of an application in a way that Selenium by itself is not capable of. A user takes a screenshot of an application at various points in the test and creates "gold standard" images of what they expect it to look like.

In subsequent runs engineers will put "checkpoints" into their code where they take another screenshot and send it to AppliTools for comparison against the "gold standard" image. Using the Eyes, AppliTools can tell you whether a logo is out of place or missing, if the layout of a page is messed up or anything in between. AppliTools Eyes allows for an incredible amount of flexibility in the comparison and allows the user to tweak the algorithm for each image. Users can choose to ignore certain areas of the image (for example, if Google Ads are present and always changing) and set the comparison type (pixel to pixel, layout, etc.).

8.4.4 PAGE OBJECT MODEL

The Page Object Model is an industry standard best practice for writing test automation scripts that allows the user to represent a page's functionality in a highly intuitive, readable format.

Page Objects allow engineers to define an application's behaviour in single place and reuse it throughout their tests. The model allows for easy and efficient updating of automation scripts as the application UI changes. RTDAA provides utilities that assist in writing these objects and take care of many of their commonly pain points users experience with them.

8.4.5 UFT

Unified Functional Testing (UFT) is a commercial testing tool for functional testing. It provides a feature set for API, web services, and GUI testing of desktop, web, and mobile applications across platforms. This tool has advanced image-based object recognition feature, reusable test components, and automated documentation. UFT uses Visual Basic Scripting for testing processes and object control. UFT can be integrated with Mercury Business Process Testing and Mercury Quality Center. This supports CI with Jenkins. It was previously known as QuickTest Professional (QTP).

It brings developers & engineers coming together under one umbrella and provides high-quality automation testing solutions. It makes functional testing less complex and cost-friendly. Its features include Cross browser & multi-platform compatibility, Optimized distributed testing, multiple testing solutions, image-based object recognition and canvas, visual test flows.

8.4.6 IBM RATIONAL FUNCTIONAL TESTER

IBM RFT is a data-driven testing platform. It supports applications such as .Net, Java, SAP, Flex, and Ajax. RFT uses VB, .Net and Java as programming languages. RFT has a feature called as Storyboard testing in which users' actions on AUT are recorded and visualized in a storyboard format through application screenshots. It can be integrated with IBM Jazz application lifecycle management systems such as IBM Rational Team Concert and Rational Quality Manager.

This tool is intended for automated functional testing & regression testing. It allows to perform data-driven and GUI testing. The automated testing is based upon script assure technology which highly improves the efficiency of testing and provides easy script maintenance. This tool does automated performance testing over web and server based apps. It has RCA capabilities to remove performance bottleneck. It provides real-time reporting and test data customizations. It also offers load and scalability testing.

8.4.7 TESTCOMPLETE

Environment Supported are web, mobile, and desktop testing. Programming/scripting languages support: JavaScript, VBScript, Python, and C++Script. Testing performed: keyword-driven and data-driven testing with Test Complete offers easy-to-use record and playback feature. Like UTF, TestComplete's GUI object recognition capability can automatically change with UI objects which helps reduce the effort to maintain test scripts when the AUT is changed. It can be integrates with Jenkins in a CI process. This tool works for desktop, mobile and web applications. With TetComplete, Functional UI tests can be built and run via robust record & replay capabilities or by scripting in your favorite languages, including Python, JavaScript, VBScript With support for applications, such as .Net, and native and hybrid iOS and Android apps, along with regression, parallel, and cross-browser testing capabilities.

8.4.8 TESTPLANT

This is an image-based automated functional testing tool, enables engineers to interact with application under test. Here User's point of view is put rather of the test scripts view often seen by engineers. This allows engineers with less programming skills to learn and apply test automation intuitively. This tool supports platforms like Web, mobile, and POS systems. Lab management and CI integration can be done with this tool.

8.4.9 TRICENTIS TOSCA

This is a model-based test automation tool which provides a broad feature set for continuous testing including dashboards, analytics, and integrations to support agile and DevOps methodologies. This helps users to optimize the reusability of test assets. It supports a range of technologies and applications such as web, mobile, and API. This has features for integration management, risk analysis, and distributed execution.

8.4.10 RANOREX

Ranorex is a quite automation tool for web, mobile, and desktop testing. The tool features for GUI recognition, reusable test scripts, and record/playback. Codeless test creation is a feature that allows new automation engineers to learn and apply test automation to their projects. The tool does Selenium integration for web application testing. Engineers can distribute the execution of their tests across platforms and browsers using Selenium grid.

8.4.11 DOMO

Domo is an analytics dashboard allowing teams to aggregate data such as test results, sonar scans, test coverage, etc. into a single, easily readable and digestible dashboard that can give an instantaneous overview of a product's health.

8.4.12 QMETRY AUTOMATION STUDIO

This is a leading software automation tool built on Eclipse IDE and leading open source frameworks, Selenium and Appium. It brings structure, efficiency, and reusability to automation efforts. The studio supports advanced automation strategy with coded automation and enables manual teams to transition into automation seamlessly with script less automation methods. In addition to test authoring, QAS provides a unified solution for an Omni channel, multi-device, and multi-locale scenario by supporting the web, mobile native, mobile web, web services, and micro-services components. This helps the enterprise to scale automation thereby eliminating the need for special purpose tools. This is a part of the Artificial Intelligence enabled QMetry Digital Quality Platform, one of the most comprehensive software quality platforms offering test management, test automation, quality analytics in a single suite.

8.4.13 TESTIM.IO

This tool leverages machine learning for the authoring, execution, and maintenance of automated test cases. Uses dynamic locators and learn with every execution/repetition of Steps**.** This produces super-fast authoring and stable tests that learn, thus eliminating the need to continually maintain tests with every code change. Companies like Netapp, Verizon Wireless, and Wix.com run over 300,000 tests using Testim.io every month. Testim, a Heavy bit portfolio company, has dual offices in San Francisco and Israel (R&D) and is backed by Spider Capital (Appurify, PagerDuty), Foundation Capital and other U.S. based investors.

8.4.14 HP QUALITY CENTER

HP Quality Center software standardizes testing. It is an integrated IT quality management software application. Automated testing is one of its key features which constantly allows to test earlier and quicker. Sharing with reusability allows HP QC to have bug-free and reliable applications.

8.4.15 TELERIK TEST STUDIO

Is a comprehensive test automation solution**.** It is for GUI, performance, and API testing. It allows to test desktop, mobile and web applications. Its features include Point-and-click test recorder, support for real coding language, central object repository and continuous integration with source control.

8.4.16 RANOREX

It is flexible, all in one GUI testing tool where you can execute automated tests flawlessly throughout all environments and devices. It has super smart object recognition feature that automatically detects any change in the UI and keeps the test going. Other significant features of Ranorex include reusable code modules, early bug finding, and seamless integration with other tools, simple test recording and easy to use the editor.

8.4.17 EGGPLANT

EggPlant is (By TestPlant) primarily aimed for application testing and GUI testing. For engineers, Eggplant offers a variety of test automation tools using which different types of testing can be performed. This has two components, Eggplant functional for functional testing and eggplant performance for performance, load and stress testing.

EggPlant works on image-based approach instead object-based approach. Using a single script, testing on multiple platforms like Windows, Mac, Linux, Solaris, can be performed.

8.4.18 SILK TEST

It is a licensed product of Micro focus aims at automated functional and regression testing. It does cross-browser support and unified test automation for a variety of applications including desktop apps, mobile apps, web apps, rich-client applications and enterprise applications. It enables efficient, speedy and high-quality automation testing.

8.4.19 SAUCE LABS

This is a selenium cloud-based concepts that offers automated testing over cross-browsers and multiple platforms. It works with mobile and desktop apps. It is known for significantly accelerating test cycles. Companies includes Yahoo, Zillow, and OpenDNS have certified that they have reduced testing time by a huge extent with the help of SauceLabs.

8.4.20 SAHI

It is a tester centric web automation tool. This cross-browser/platform tool comes with features, such as smart accessory identification, record and playback on any browser, no ajax timeout issues, end to end reporting, powerful scripting and inbuilt excel framework.

8.4.21 HP LOAD RUNNER

This is again an automated load and performance testing tool provided by Hewlett Packard. It works in different environments and over different types of applications. It supports mobile and cloud testing as well. HP Load Runner measures system performance, allows to do the RCA and fix the bugs before the application is released to live environment.

8.4.22 NEOLOAD

It is also very popular and automated performance testing tool. It replicates user activities and finds out the system bottlenecks. It supports both mobile and web apps. This supports both mobile and web apps.

8.4.23 PERFECTO

This automated application testing over cross browsers and mobile devices. It can be plugged with other test automation frameworks.

8.4.24 WEBLOAD

This is a load, performance, and stress testing tool for mobile and web applications. It integrates with testing tools like Selenium, Perfecto mobile, etc. It provides dashboards to give clear picture of tests run.

8.4.25 TEST ANYWHERE

It is a tool for frontend automated testing tool that replicates the real user actions and do not need to write any code.

8.4.26 VISUAL STUDIO TEST PROFESSIONAL

This tool provides exploratory browser-based testing**.** This tool for streamlining quality and continuous delivery. It has the free trial available as well.

8.4.27 TESTINGWHIX

It is a licensed tool which offers automation solutions for regression testing, web testing, mobile testing, cross-browser testing, web services testing and database testing. It has architecture which supports continuous integration very well.

8.4.28 TOSCA TESTSUITE

It is an automated tool for performing functional testing, regression testing. And business dynamic steering.

8.5 TOOLS USED FOR CROSS PLATFORM TESTING

1) Browserstack: this is based on cloud and tests websites on multiple browsers and OS combinations. Trial version may be tested and thereafter fee need to be paid as number of users licenses.

2) Testize: This tool works on standards and analyze websites identifying issues in performance, site rendering and compatibility.
3) Test Plant: This is cross browser testing automation tool to test Websites in different Operating system environments and configurations.
4) Browser shots: this is a free Open source tool which creates screenshots of website to run on different browsers. These screenshots are tests n real browser on different operating systems.

8.6 HOW THE RIGHT TOOL BE CHOSEN

1) Market Research: Search tool available whether Free Open Source, Community versions or Paid one fit in the requirements.
2) Experts View: Get feedback from the users and experts or from Forum of experts to get experiences on the features of tools.
3) Personal Experience: Do some research on your personal experience and shortlist some best tools that best suits requirements and affordability.
4) Prepare comparison chart and do SWOT Analysis to select the best tool.
5) Have at least 2 tools for the requirements we have to get experience and backup.

CHAPTER 9

CONCLUSION

CHAPTER 9

CONCLUSION

9.1 SUMMARY & RECOMMENDATIONS

Adaptive Lifecycle Management is a structured methodology geared toward ensuring successful implementation of automated testing. The ALM approach mirrors the benefits of modern rapid application development efforts, where such efforts engage the user early in the development cycle. The user of the software product is actively involved throughout analysis, design, development, and test of each software build, which is delivered in an incremental fashion may be called as beta versions.

Many organizations have adopted the ALM. Many companies have adopted the book and ALM as their company standard for automated software testing. Others believe that industry automated tool vendors will soon be incorporating the book's structured methodology within their tools. Instead of performing the entire test lifecycle haphazardly, software managers will use an ALM-compliant test tool that automatically supports (and possibly enforces) the book's sound building-block approach to the test effort.

Almost all of the modern cloud or API applications are developed in the new micro-services architecture, there are certain factors or requirements the application should closely adhere to make sure it is cloud ready or can be used to easily expand and change services/behaviours as per market demand. These standards help us achieve applications to be in sync with the cloud computing definition from NIST as closely as possible to be used and deployed on a cloud infrastructure.

A Practical implementation of Adaptive automation system by setting up Automated Build System with Continuous Integration and Continuous delivery is demonstrated using setting up (deploying) Free/Open source

Software (Jenkins) over cloud for accomplishing multiple tasks such as triggering software build as soon as check-ins made in repo, or then running unit tests, Code Coverage and Static Code Analysis and parsing code for bugs, running integration tests etc. As there are many plugins freely available, if configured properly it encompasses (approaches to) adaptive learning as per environment and schedule.

This research continue demonstrating patterns used in adaptive automation testing. It includes continuously Learning, Speculate and Collaborate, Spiral model have Plan, Build and Revise continuously. It can be understood with Complex Adaptive Systems Theory. It includes Adaptive Software Development with conceptual perspective and particle perspective. CAs have concepts of Emergence, Complexity and Quality.

Rapid Application Development can be used for adaptive development because it’s iterative, time boxed and change tolerant.

Adaptive S/W development management has polices namely Passive and Active Management and Leadership-Collaboration Management. Adaptive Automation Testing have some patterns named as Data Patterns, Technical Patterns, Proxy Patterns, Business Patterns, Page Object Patterns, Façade Patterns, Factory Patterns, and Singleton Patterns.

There are some tools and technologies available to achieve results on Run Time Dynamic Adaptive Automation Testing (RTDDA). These tools are mainly divided in two categories Free/Open Source Tools and Commercial Tools. There are some tools used for cross platform testing, Hence in the list of many tools there should be strategy to choose right tools on the basis of Market research, Experts View and Personal Experience. Sometime if required tools may be developed in-house.

A study of Adaptive Algorithm is done used to develop adaptive software algorithms and tools mainly categorized in Heuristic Strategy and Fuzzy Logic.

9.2 CUSTOMIZED ADAPTIVE ALGORITHM FOR TAKING DECISION IN RUN TIME DYNAMIC NETWORKED CLOUD STORAGE ENVIRONMENT.

Step 1: Collect maximum possible parameters or variables (System Environment) need to run/execute the system.

Stop 2: Define set for variables, based on components and functionalities.

Step 3: Collect the variables in Set or arrange them in group.

Step 4: As there will be large number of set, and different states of variables so apply Heuristic Evaluation Functions to calculate the cost of optimal path between two states.

Step 5: Every set will become a node so apply pure heuristic search, in which nodes will be expanded as per their heuristic values. This step will result a list of unique shortest paths between nodes and dispose long paths.

Step 6: Now apply Best First search to get the cost of path with cost of goal. By using priority queue, it will return optimized weighted cost.

Step 7: As nodes in networked dynamic cloud storage environment has various sizes and capabilities; hence next fuzzy logic will be applied which works on the levels of possibilities of input to achieve the definite output.

Step 8: To get acceptable reasoning and overcome the uncertainties, covert node values in fuzzy sets called a fuzzification.

Step 9: Convert/arrange fuzzy sets into logical constructs, decision making statements.

Step 10: Evaluate and combine results with the help of (Inference Engine).

Step 11: Then output data need to be converted back into non-fuzzy values. (Called as de-fuzzification)

Step 12: Repeat Step 7 to 11 for all data sets.

Step 13: Repeat Step 4 and 5 if required and there is a larger set of data.

Step 14: Go to step 2 to check if any data/functionality is remaining. Follow step 3 onwards if necessary.

Step 15: End of execution of Adaptive Algorithm in dynamic environment.

Bibliography

Bibliography

1) A.E. Eiben (Author), J.E. Smith (Author), Introduction to Evolutionary Computing (Natural Computing Series) Paperback – 17 Oct 2016

2) Elfriede Dustin and Jeff Rashka(Author), Automated Software Testing: Introduction, Management, and Performance: Introduction, Management, and Performance

3) Marc Farley(Author), Storage Networking Fundamentals: An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems

4) Robert W Kembel(Author), Fibre Channel a Comprehensive Introduction Paperback, 6

5) Robert W Kembel (Author), Fibre Channel Over Ethernet (FCoE) Paperback, 28 Jan 2010

6) Judith Hurwitz and Robin Bloor (Author), Cloud Computing For Dummies, 13 November 2009

7) Paul Massiglia Richard Barker (Author), Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs Paperback 2008

8) Jez Humble (Author), David Farley (Author), Martin Fowler (Foreword), Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation Textbook Binding – 2016

9) Learning Robotic Process Automation Paperback – Import, 1 Mar 2018, By Alok Mani Tripathi (Author)

10) Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature

Series) Hardcover – 27 Jul 2010, By Jez Humble (Author), David Farley (Author)

11) Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin) Paperback – 25 Sep 2017, By Robert C. Martin (Author)

12) Software Test Automation: Effective Use of Test Execution Tools. By Mark Fewster and Dorothy Graham

13) Head First Design Patterns: A Brain-Friendly Guide - 10th Anniversary Edition (Covers Java 8), 2016, By Eric Freeman and Elisabeth Robson

14) Towards Adaptive Flexibility in Automation Systems, 29 May 2015, By Van Der Linden Dirk

15) Advances in Human Factors, Software, and Systems Engineering: Proceedings of the AHFE 2016 International Conference on Human Factors, Software, and Systems ... Intelligent Systems and Computing Book 492) 26 July 2016 by Ben Amaba

16) Evolutionary Learning Algorithms for Neural Adaptive Control (Perspectives in Neural Computing), 15 August 1997, By Dimitris C. Dracopoulos

17) Anticipatory Behavior in Adaptive Learning Systems: From Brains to Individual and Social Behavior (Lecture Notes in Artificial Intelligence), 22 August 2007, By Martin V. Butz and Olivier Sigaud

18) Intelligent Control: A Stochastic Optimization Based Adaptive Fuzzy Approach (Cognitive Intelligence and Robotics), 28 August 2018, By Kaushik Das Sharma and Amitava Chatterjee

19) Model-Based Design of Adaptive Embedded Systems, 8 February 2015, By Twan Basten and Roelof Hamberg

20) Cooperative Control of Multi-Agent Systems: A Consensus Region Approach (Automation and Control Engineering Book 57), By Zhongkui Li and Zhisheng Duan

21) Evolutionary Learning Algorithms for Neural Adaptive Control (Perspectives in Neural Computing), 15 August 1997, By Dimitris C. Dracopoulos

22) Software Engineering: A Practitioner's Approach, By Roger S Pressman

23) Cloud Computing: Concepts, Technology & Architecture, 1e–2014, By Erl (Author)

24) Moving to the Cloud: Developing Apps in the New World of Cloud Computing Paperback – 2012, By Sitaram (Author)

25) Internet of Things: A Hands-On Approach Paperback – 2015, By Arsheep Bahga (Author), Vijay Madisetti (Author)

26) Cloud Security: A Comprehensive Guide to Secure Cloud Computing, 2010, By Ronald L. Krutz and Russell Dean Vines

27) Cloud Computing and Virtualization, 15 April 2011, By Abhay Bhadani

28) PC Hardware: The Complete Reference, 1 July 2017, By Craig Zacker and John Rourke

29) Storage Networks: The Complete Reference, 1 March 2003, By Robert Spalding

30) iSCSI: The Universal Storage Connection, 9 January 2015, By John L. Hufferd

31) VMware Software-Defined Storage: A Design Guide to the Policy-Driven, Software-Defined Storage Era, 11 August 2016, By Martin Hosken

32) Using SANs and NAS: Help for Storage Administrators, 9 February 2009, By W. Curtis Preston

33) Storage Networks, 7 June 2004, By Daniel J. Worden

34) On the Design of High Performance Data Center Networks, January 15, By Guo Zhiyang

35) Cloud Computing: Automating the Virtualized Data Center, 1e 2012, By Josyula

36) Mastering Jenkins, 27 October 2015, By Jonathan McAllister

37) Continuous Delivery with Docker and Jenkins, 24 August 2017, By Rafal Leszko

38) Learning Continuous Integration with Jenkins: A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins 2, 2nd Edition, 20 December 2017

39) Storage Management in Data Centers: Understanding, Exploiting, Tuning, and Troubleshooting Veritas Storage Foundation, By Volker Herminghaus and Albrecht Scriba

40) Jenkins 2: Up and Running- Evolve Your Deployment Pipeline for Next Generation Automation, 15 June 2018, By Brent Laster

41) Jenkins: The Definitive Guide: Continuous Integration for the Masses, By John Ferguson Smart

42) Hybrid Cloud for Dummies, 2013, By Judith Hurwitz and Marcia Kaufman

43) 29 September 2018, By Rahul Shetty

44) Android Continuous Integration: Build-Deploy-Test Automation for Android Mobile Apps, 26 August 2017, By Pradeep Macharla

45) Build Automation: A Successful Design Process, 5 November 2017, By Gerard Blokdyk

46) Robotic Building (Springer Series in Adaptive Environments),11 May 2015, By Feroz Pearl Louis and Gaurav Gupta

47) Software Automation Testing Tools for Beginners, 26 April 2012, By Rahul Shende

48) Test Automation Using HP Unified Functional Testing: Explore Latest Version of QTP, 1 August 2013, By MR Navneesh Garg

49) Test Complete Cookbook,16 December 2013, By Gennadiy Alpaev

50) Foundations of Software Testing ISTQB Certification, February 2015, By Rex Black and Erik van Veenendaal

51) Stop Coding, Learn to test automate without coding and get that automation testing job, By Ajamo Adams

52) Securing Internet Protocol (IP) Storage: A Case Study Siva Rama Krishnan Somayaji1 and Ch. A. S Murty2, International Journal of Next Generation Network (IJNGN), Vol.2, No.1, March 2010

53) Bindu Trikha / A Journey from floppy disk to cloud storage (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 1449-1452

54) Insoon Jo et al. / International Journal on Computer Science and Engineering (IJCSE) Workload-aware VM Scheduling on Multicore Systems

55) Informatica Economică vol. 15, no. 4/2011, A Perspective on the Benefits of Data Virtualization Technology, Ana-Ramona BOLOGA, Razvan BOLOGA Academy of Economic Studies, Bucharest, Romania

56)(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 1, 2012, Effective Security Architecture for Virtualized Data Center Networks, Udeze Chidiebele, Okafor Kennedy .C

57)Computer Science 2009, High availability using virtualization, Federico Calzolari

58)Technical Report 2013-04-13, Rice University, Making I/O Virtualization Easy with Device Files, Ardalan Amiri Sani, Sreekumar Nair, Lin Zhong, Quinn Jacobson

59)Building and Operating a Large Scale Storage Facility Brookhaven National Laboratory's RHIC, Physics 2003, Computing Facility Maurice Askinazi, David Free, Bruce Gibbard, Thomas Throwe

60)Annals of Dun rea de Jos University. Fascicle I: Economics and Applied Informatics 2007, Availability Analysis and Improvement of Software Rejuvenation Using Virtualization, Thandar THEIN, Sung-DoCHI, JongSou PARK

61)International Journal of Information Technology Convergence and Services (IJITCS) Vol.2, No.6, December 2012, SEMI SYMMETRIC METHOD OF SAN STORAGE VIRTUALIZATION, Mrs. Dhanamma Jagli, Mr. Ramesh Solanki, Mrs. Rohini Temkar, Mrs. Laxmi Veshapogu

62)The Annals of "Dunarea de Jos" University of Galati, Fascicle I – 2009. Economics and Applied Informatics. Years XV - ISSN 1584-0409, a Study of Key Management for Encrypted Storage in Storage Area Network, Hai Xin LU

63)Volume 3, No. 12, December 2012, Journal of Global Research in Computer Science, RESEARCH PAPER, Sachin Sharma, Mrs. VISHAWJYOTI, Research on study and analysis of automation testing techniques.

64) Sunil L. Bangare et al. / International Journal of Engineering Science and Technology (IJEST), AUTOMATED TESTING IN DEVELOPMENT PHASE, SUNIL L. BANGARE, SACHIN M. KAMBLE, PALLAVI S. BANGARE, ABHIJIT V. NAIK

65) Journal of Software Engineering and Applications, 2014, 7, 440-451 Published Online May 2014 in SciRes. http://www.scirp.org/journal/jsea http://dx.doi.org/10.4236/jsea.2014.75041, Challenges for Software Engineering in Automation

66) Journal of Computers 2007, An Integrated Self-Testing Framework for Autonomic Computing Systems, DOI: 10.4304/jcp.2.9.37-49, Tariq M. King, Alain E. Ramirez, Rodolfo Cruz, Peter J. Clarke

67) Computer Science 2015, Adaptive Automation: Leveraging Machine Learning to Support Uninterrupted Automated Testing of Software Applications, Rajesh Mathur, Scott Miles, Miao Du

68) ISSN: 1690-4524 SYSTEMICS, CYBERNETICS AND INFORMATICS VOLUME 3 - NUMBER 5, Training People to Use Automation: Strategies and Methods, John S. BARNETT U.S. Army Research Institute for the Behavioural and Social Sciences Orlando, FL 32826 USA

69) International Journal of Computer Applications (0975 – 8887) Volume 31– No.7, October 2011, K model for designing Data Driven Test Automation Frameworks and its Design Architecture “Snow Leopard”, Rohan R. Kachewar Automation Architect

70) Computer Science 2015, Automation of Smartphone Traffic Generation in a Virtualized Environment, Tanya Jha, Rashmi Shetty

71) Journal of Software Engineering and Applications, 2014, 7, 617-625 Published Online July 2014 in SciRes. http://www.scirp.org/journal/jsea, http://dx.doi.org/10.4236/jsea.2014.78057, Regression Testing in Developer Environment for Absence of Code Coverage, M. Thillaikarasi, K. Seetharaman

72) Latika Kharb et al, Int. J. Comp. Tech. Appl., Vol 2 (4), 798-802 IJCTA | JULY-AUGUST 2011 Available online@www.ijcta.com 798 ISSN:2229-6093, Reliable Software Development with Proposed Quality Oriented Software Testing Metrics, Ms. Latika Kharb, Dr. Vijay Singh Rathore

73) Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software ISSN: 2248-7441 www.fundacioniai.org/raccis/index.htm raccis@fundacioniai.org, Tools and Behaviour Abstraction: A Future for Software Engineering Las Herramientas y la Abstraction del, Comportamiento: Un Futuro para la Ingeniería de Software

74) International Journal of Soft Computing & Engineering 2012, A Research Study on importance of Testing and Quality Assurance in Software, Development Life Cycle (SDLC) Models, Maneela Tuteja, Gaurav Dubey

75) Salesian Journal on Information Systems 2010, Optimization in Software Testing Using Metaheuristics, FREITAS, F. G.,MAIA, C. L., B.,CAMPOS, G. A. L.,SOUZA, J. T.

76) IJCSI International Journal of Computer Science Issues, Vol. 9, Issue No 3, May 2012 ISSN (Online): 1694-0814 www.IJCSI.org, Software as a Service (SaaS) Testing Challenges- an In-depth Analysis, Prakash.V Ravikumar Ramadoss Gopalakrishnan.S

77) Information Technology Journal 2010, Software Development Methodologies, Trends and Implications: A Testing Centric View, Xihui Zhang, Tao Hu, Hua Dai, Xiang Li

78) Computer Science 2013, Classification of automatic software build methods, Marcin Kawalerowicz

79) International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.6, December 2012, CLOUD PENETRATION TESTING, Ralph LaBarge and Thomas McGuire

80) Computer Science 2013, Software Testing Models against Information Security Requirements, Alexey Markov

81) Sunil Kumar et al, Int. J. Comp. Tech. Appl., Vol 2 (4), 1082-1091 IJCTA | JULY-AUGUST 2011 Available online@www.ijcta.com 1082 ISSN:2229-6093, Formal Methods of Software Testing and Terminology, Sunil Kumar, Dr. P.K Yadav

82) IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, Sept 2012 ISSN (Online): 2231 –5268, www.ijcsms.com, How Automated Testing Tools Are Showing Its Impact in the Field Of Software Testing, Deepti Gaur, Dr. Rajender Singh Chhillar

83) Computer Science 2014, Survey on software testing techniques in cloud computing, Priyadharshini. V, Malathi. A

84) Premal B. Nirpal et al. / International Journal on Computer Science and Engineering (IJCSE), A Brief Overview of Software Testing Metrics, Mr. Premal B. Nirpal, Dr. K. V. Kale

85) International Journal of Computer and Distributed System 2013, SOFTWARE TESTING AND SOFTWARE DEVELOPMENT LIFECYCLES, Chitra Wasnik

86) Hands-On Automation Testing with Java for Beginners: Build automation testing frameworks from scratch with Java,

87) https://en.wikipedia.org/wiki/Data_center

88) https://en.wikipedia.org/wiki/Cloud_computing

89) https://en.wikipedia.org/wiki/Service-oriented_architecture

90) https://en.wikipedia.org/wiki/Computer_data_storage

91) https://en.wikipedia.org/wiki/Fibre_Channel_over_Ethernet

92) https://en.wikipedia.org/wiki/ATA_over_Ethernet

93) https://en.wikipedia.org/wiki/InfiniBand

94) https://en.wikipedia.org/wiki/Fibre_Channel

95) https://www.snia.org/home

96) https://aws.amazon.com/compliance/data-center/data-centers/

97) https://docs.microsoft.com/en-us/azure/

98) https://jenkins.io/doc/

99) https://wiki.jenkins.io/display/JENKINS/Extend+Jenkins

100) https://en.wikipedia.org/wiki/Virtual_machine

101) https://cloud.vmware.com/

102) https://buildbot.net/

103) http://www.seleniumhq.org/

104) https://www.katalon.com/

105) http://www.robotframework.org/

106) http://watir.com/

107) https://software.microfocus.com/fr-ca/software/uft

108) https://www.ibm.com/

109) https://smartbear.com/

110) https://www.testplant.com/

111) https://www.tricentis.com/

112) https://www.ranorex.com/

113) https://en.wikipedia.org/wiki/Software_testing

114) https://en.wikipedia.org/wiki/Debugging

115) Mastering Mobile Test Automation,

116) https://en.wikipedia.org/wiki/Jenkins(software)

117) https://en.wikipedia.org/wiki/Continuous_delivery

118) https://en.wikipedia.org/wiki/Software_build

119) https://en.wikipedia.org/wiki/Build_automation

120) https://en.wikipedia.org/wiki/Acceptance_test%E2%80%93driven_development

121) https://en.wikipedia.org/wiki/Behavior-driven_development

122) https://en.wikipedia.org/wiki/Automatic_test_pattern_generation

123) https://en.wikipedia.org/wiki/Test-driven_development

124) https://en.wikipedia.org/wiki/Software_design_pattern

125) https://ieeexplore.ieee.org/document/5345873

126) http://web.cse.ohio-state.edu/~soundarajan.1/papers/testPatterns.pdf

127) https://en.wikipedia.org/wiki/Automation

128) https://en.wikipedia.org/wiki/Test_automation

129) https://en.wikipedia.org/wiki/Cloud_storage

130) https://en.wikipedia.org/wiki/Software_testing

131) https://en.wikipedia.org/wiki/Software_release_life_cycle

132) https://en.wikipedia.org/wiki/Glossary_of_artificial_intelligence

133) https://en.wikipedia.org/wiki/Automated_planning_and_scheduling

134) https://www.securedatarecovery.com/infographics/the-evolution-of-data-storage

135) https://en.wikipedia.org/wiki/Model-based_testing

136) https://www.linkedin.com/beat/intelligent-quality-automation-transforming-qa-using-power-ahmed/

Publications

List of Publications

[1] Vikas Kumar Choudhary & Dr. Sanjay Chaudhary “Data Centre or Cloud for Dynamic Storage: In Modern Perspective, **Naveen Shodh Sansar (An International Refereed/ Peer Review Research Journal) (U.G.C. Jr. No. 64728) ISSN 2320-8767, E- ISSN 2394-3793, Impact Factor - 4.710 (2016) July to September 2017 E-Journal 341**

[2] Vikas Kumar Choudhary & Dr. Sanjay Chaudhary “Practises for Building Quality Software with Automation: A Practical Approach” **Naveen Shodh Sansar (An International Refereed/ Peer Review Research Journal) (U.G.C. Jr. No. 64728) ISSN 2320-8767, E- ISSN 2394-3793, Impact Factor - 5.110 (2017) January to March 2018 E-Journal 466**

Plagiarism Report