CHAPTER 2

RESEARCH METHODOLOGY

**Keywords: Automation Tools, Analysing Tools, Cloud Computing, Heuristic Search, A\*, Fuzzy Logic,** Cloud Computing, Cloud Services, Cloud Network, Cloud Database, Cloud Management, Storage Devices, Data Storage, Adaptive Automation, Adaptive Automation, Real Time Systems, Run Time, Dynamic,

# 2.1 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.

# 2.2 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.

## 2.21 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.

# 2.3 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.

## 2.3.1 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

# 2.1 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 tooluses 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](http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf) 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.
* **De-fuzzification 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 μA: 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 de-fuzzification**

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.

# 2.6 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

## 2.6.1 Neuro-ergonomics, computerized adaptive test and Level of Automation

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.

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.

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.

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).

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)**.

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.

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.

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.

## 2.6.2 Dynamic (Run Time) Storage and Data (Center) Virtualization

**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.

**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

Insoon Jo, Im Y. Jung, Heon Y. Yeoma (2006) worked on Workload Aware Vm Scheduling On Multicore Systemsanalysed 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 .

Ramona A, Razvan R. (2006) published paper on Perspective on the Benefits of Data Virtualization Technology 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.

Udeze Chidiebele. C, Okafor Kennedy .C (2008) worked on Effective Security Architecture for Virtualized Data Center Networks and 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.

Soran Philip E. (Eden Prairie, MN), Guider; John P. (North Oaks, MN) ( 2009 ) worked on Virtual disk drive system and method 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.

Patil S.V and Honwadkar K.N (2009)Studies Unified Virtual Storage 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.

Hiteshi Atif (2010) presented paper on Virtualization as the new and key concept in the field of information technology 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.

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.

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.

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.

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.

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

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.

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

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.

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.

## 2.6.3 Automated (Adaptive Control) System Testing

Study And Analysis Of Automation Testing Techniques, [**Sachin Sharma**](http://www.oalib.com/search?kw=Sachin%20Sharma&searchField=authors)**,** [**Mrs. VISHAWJYOTI**](http://www.oalib.com/search?kw=Mrs.%20VISHAWJYOTI&searchField=authors) 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.

Automated Testing in Development Phase, [**SUNIL L. BANGARE**](http://www.oalib.com/search?kw=SUNIL%20L.%20BANGARE&searchField=authors)**, SACHIN M. KAMBLE,** [**PALLAVI S. BANGARE**](http://www.oalib.com/search?kw=PALLAVI%20S.%20BANGARE&searchField=authors)**,** [**ABHIJIT V. NAIK**](http://www.oalib.com/search?kw=ABHIJIT%20V.%20NAIK&searchField=authors)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.

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.

An Integrated Self-Testing Framework for Autonomic Computing Systems [**Tariq M. King**](http://www.oalib.com/search?kw=Tariq%20M.%20King&searchField=authors)**,** [**Alain E. Ramirez**](http://www.oalib.com/search?kw=Alain%20E.%20Ramirez&searchField=authors)**, 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.

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.

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.

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.

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.

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.

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

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

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.

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.

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.

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.

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.

## 2.6.4 Cloud (Dynamic) Storage Network

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.

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.

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.

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 an 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.

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

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.

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.

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.

Cloud versus On-Premise Computing, Cameron Fisherpublished 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.

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.

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.

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.

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.

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.

## 2.6.5 Storage Area Networks (Networked Storage)

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.

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.

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.

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.

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.

Key Management for Encrypted Storage in Storage Area Network, [**Hai Xin LU**](http://www.oalib.com/search?kw=Hai%20Xin%20LU&searchField=authors) 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.

Semi Symmetric Method of SAN Storage Virtualization, Dhanamma Jagli, Ramesh Solanki, Rohini Temkar, Laxmi Veshapogu presented a paper onVirtualization, 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.

## 2.6.6 Software (Engineering) Models, Framework and Test Strategies

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.

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.

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.

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

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.

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.

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.

Optimization in Software Testing Using Metaheuristics, **FREITAS, F. G., MAIA, C. L. B., CAMPOS, G. A. L., SOUZA, J. T. (2010), mentioned that t**here 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.

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.

Software Development Methodologies, Trends and Implications: A Testing Centric View, [**Xihui Zhang**](http://www.oalib.com/search?kw=Xihui%20Zhang&searchField=authors)**,** [**Tao Hu**](http://www.oalib.com/search?kw=Tao%20Hu&searchField=authors)**,** [**Hua Dai**](http://www.oalib.com/search?kw=Hua%20Dai&searchField=authors)**,** [**Xiang Li**](http://www.oalib.com/search?kw=Xiang%20Li&searchField=authors) **(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.

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.

# 2.7 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.

# References

1. Siva Rama Krishnan Somayaji and Ch. A. S Murty, Securing Internet Protocol (IP) Storage: A Case Study, International Journal of Next Generation Network (IJNGN), Vol.2, No.1, March 2010
2. 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
3. Insoon Jo et al. / International Journal on Computer Science and Engineering (IJCSE) Workload-aware VM Scheduling on Multicore Systems
4. Ana-Ramona BOLOGA, Razvan BOLOGA , Informatica Economică A Perspective on the Benefits of Data Virtualization Technology, Academy of Economic Studies, Bucharest, Romania, vol. 15, no. 4/2011
5. Udeze Chidiebele, Okafor Kennedy .C, (IJACSA) International Journal of Advanced Computer Science and Applications Effective Security Architecture for Virtualized Data Center Networks, , Vol. 3, No. 1, 2012
6. Federico Calzolari, High availability using virtualization, [Computer Science](http://www.oalib.com/journal/12544/1)  2009
7. Ardalan Amiri Sani, Sreekumar Nair, Lin Zhong, Quinn Jacobson, Making I/O Virtualization Easy with Device Files, , Technical Report 2013-04-13, Rice University
8. Maurice Askinazi, David Free, Bruce Gibbard, Thomas Throwe, Building and Operating a Large Scale Storage Facility Brookhaven National Laboratory’s RHIC, [Physics](http://www.oalib.com/journal/12542/1)  2003, Computing Facility
9. [Thandar THEIN](http://www.oalib.com/search?kw=Thandar%20THEIN&searchField=authors), Sung-dochi, jongsou PARK, Annals of Dun rea de Jos University. Fascicle I: Economics and Applied Informatics 2007, Availability Analysis and Improvement of Software Rejuvenation Using Virtualization
10. Mrs. Dhanamma Jagli, Mr. Ramesh Solanki, Mrs. Rohini Temkar, Mrs. Laxmi Veshapogu, SEMI SYMMETRIC METHOD OF SAN STORAGE VIRTUALIZATION ,International Journal of Information Technology Convergence and Services (IJITCS) Vol.2, No.6, December 2012
11. Hai Xin LU, 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
12. [Sachin Sharma](http://www.oalib.com/search?kw=Sachin%20Sharma&searchField=authors), [Mrs. Vishawjyoti](http://www.oalib.com/search?kw=Mrs.%20VISHAWJYOTI&searchField=authors), Research on study and analysis of automation testing techniques, Journal of Global Research in Computer Science, RESEARCH PAPER,. Volume 3, No. 12, December 2012
13. SUNIL L. BANGARE, SACHIN M. KAMBLE, PALLAVI S. BANGARE, ABHIJIT V. NAIK ,International Journal of Engineering Science and Technology (IJEST), AUTOMATED TESTING IN DEVELOPMENT PHASE
14. Challenges for Software Engineering in Automation

Journal of Software Engineering and Applications, 2014, 7, 440-451 Published Online May 2014 in scires. [Http://www.scirp.org/journal/jsea](http://www.scirp.org/journal/jsea) <http://dx.doi.org/10.4236/jsea.2014.75041>

1. [Tariq M. King](http://www.oalib.com/search?kw=Tariq%20M.%20King&searchField=authors), [Alain E. Ramirez](http://www.oalib.com/search?kw=Alain%20E.%20Ramirez&searchField=authors), [Rodolfo Cruz](http://www.oalib.com/search?kw=Rodolfo%20Cruz&searchField=authors), [Peter J. Clarke](http://www.oalib.com/search?kw=Peter%20J.%20Clarke&searchField=authors), [Journal of Computers](http://www.oalib.com/journal/10268/1)  2007, An Integrated Self-Testing Framework for Autonomic Computing Systems, DOI: 10.4304/jcp.2.9.37-49
2. Rajesh Mathur, [Scott Miles](http://www.oalib.com/search?kw=Scott%20Miles&searchField=authors), [Miao Du](http://www.oalib.com/search?kw=Miao%20Du&searchField=authors), Adaptive Automation: Leveraging Machine Learning to Support Uninterrupted Automated Testing of Software Applications, Computer Science  2015
3. John S. BARNETT U.S. Army Research Institute for the Behavioral and Social Sciences Orlando, FL 32826 USA, Training People to Use Automation: Strategies and Methods, ISSN: 1690-4524 SYSTEMICS, CYBERNETICS AND INFORMATICS VOLUME 3 - NUMBER 5
4. Rohan R. Kachewar Automation Architect, K model for designing Data Driven Test Automation Frameworks and its Design Architecture “Snow Leopard”, International Journal of Computer Applications (0975 – 8887) Volume 31– No.7, October 2011
5. [Tanya Jha](http://www.oalib.com/search?kw=Tanya%20Jha&searchField=authors), [Rashmi Shetty](http://www.oalib.com/search?kw=Rashmi%20Shetty&searchField=authors), Automation of Smartphone Traffic Generation in a Virtualized Environment, [Computer Science](http://www.oalib.com/journal/12544/1)  2015
6. M. Thillaikarasi, K. Seetharaman, Regression Testing in Developer Environment for Absence of Code Coverage, Journal of Software Engineering and Applications, 2014, 7, 617-625 Published Online July 2014 in scires. [Http://www.scirp.org/journal/jsea](http://www.scirp.org/journal/jsea), <http://dx.doi.org/10.4236/jsea.2014.78057>
7. Ms. Latika Kharb, Dr. Vijay Singh Rathore, Reliable Software Development with Proposed Quality Oriented Software Testing Metrics, Int. J. Comp. Tech. Appl., Vol 2 (4), 798-802 IJCTA | JULY-AUGUST 2011 Available online@www.ijcta.com 798 ISSN:2229-6093
8. Maneela Tuteja, Gaurav Dubey, A Research Study on importance of Testing and Quality Assurance in Software, Development Life Cycle (SDLC) Models, International Journal of Soft Computing & Engineering  2012
9. FREITAS,[F. G.](http://www.oalib.com/search?kw=%20F.%20G.&searchField=authors), [MAIA](http://www.oalib.com/search?kw=MAIA&searchField=authors), C. L., B., CAMPOS, G. A. L.,SOUZA,[J. T.](http://www.oalib.com/search?kw=%20J.%20T.&searchField=authors), Optimization in Software Testing Using Metaheuristics,

Salesian Journal on Information Systems 2010

1. Prakash.V Ravikumar, Ramadoss Gopalakrishnan.S, Software as a Service (saas) Testing Challenges- an In-depth Analysis, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue No 3, May 2012 ISSN (Online): 1694-0814 www.IJCSI.org
2. Xihui Zhang, [Tao Hu](http://www.oalib.com/search?kw=Tao%20Hu&searchField=authors), Hua Dai, Xiang Li, Software Development Methodologies, Trends and Implications: A Testing Centric View, Information Technology Journal  2010
3. Marcin Kawalerowicz, Classification of automatic software build methods, Computer Science  2013
4. International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.6, December 2012, CLOUD PENETRATION TESTING, Ralph labarge and Thomas mcguire
5. Alexey Markov, Software Testing Models against Information Security Requirements, Computer Science  2013
6. Sunil Kumar, Dr. P.K Yadav, Formal Methods of Software Testing and Terminology, Int. J. Comp. Tech. Appl., Vol 2 (4), 1082-1091 IJCTA | JULY-AUGUST 2011 Available online@www.ijcta.com 1082 ISSN:2229-6093
7. Deepti Gaur, Dr. Rajender Singh Chhillar, How Automated Testing Tools Are Showing Its Impact in the Field Of Software Testing, IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, Sept 2012 ISSN (Online): 2231 –5268
8. [Priyadharshini. V](http://www.oalib.com/search?kw=Priyadharshini.%20V&searchField=authors), [Malathi. A](http://www.oalib.com/search?kw=Malathi.%20A&searchField=authors) Survey on software testing techniques in cloud computing, [Computer Science](http://www.oalib.com/journal/12544/1)  2014
9. Mr. Premal B. Nirpal, Dr. K. V. Kale, A Brief Overview of Software Testing Metrics, International Journal on Computer Science and Engineering (IJCSE)
10. [Chitra Wasnik](http://www.oalib.com/search?kw=Chitra%20Wasnik&searchField=authors), SOFTWARE TESTING AND SOFTWARE DEVELOPMENT LIFECYCLES , [International Journal of Computer and Distributed System](http://www.oalib.com/journal/7653/1)  2013
11. [Jun Yao](http://www.oalib.com/search?kw=Jun%20Yao&searchField=authors), [Ji-Wu Shu](http://www.oalib.com/search?kw=Ji-Wu%20Shu&searchField=authors),Wei-minzheng, Distributed Storage Cluster Design for Remote Mirroring Based on Storage Area Network, 2007