Literature Review

In a cloud computing system, there's a significant workload shift. Local computers no longer have to do all the heavy lifting when it comes to running applications. The network of computers that make up the cloud handles them instead. Hardware and software demands on the user's side decrease. The only thing the user's computer needs to be able to run is the cloud computing systems interface software, which can be as simple as a Web browser, and the cloud's network takes care of the rest (Mathur P (2010)). The authors discuss the relationship between grid and cloud computing, identifies gaps and overlaps in existing standards and identifies how grid and cloud technology could be exploited to improve the efficiency of NGN resources and to offer new “data” services to consumers. This will enable telecom operators to manage their resources in a dynamic and optimal way by a single platform. This paper describes the approach taken by the European Telecommunications Standards Institute (ETSI) Technical Committee for grid computing (TC GRID) to identify gaps and overlaps in grid/cloud computing standards and to support the integration of grid/cloud computing with the NGN architecture (Caryer G et al (2009)).

Component selection is not an easy task in Component Based Software Engineering and it is very difficult to select component for CBSE. Component Based Software Engineering (CBSE) is a concerned with the assembly of pre-existing software components that leads to a software system that responds to client-specific requirements. This paper presents an approach for defining evaluation criteria for reusable software components. We introduce taxonomy of factors that influence selection, describe each of them, and present a hierarchical decomposition method for deriving reuse goals from factors and formulating the goals into an evaluation criteria hierarchy. It also presents a summary of the common problems in reusable off-the-shelf software selection, describes the method. It also indicates that the evaluated aspects of the method are feasible; improve the quality and efficiency of reusable software selection. In this paper the selection of component is done on the basis of the cost of the component which is calculated on the basis of the quality attributes of the component. The approach used for selecting the component is a part of OTSO method that has been developed for reusable component selection process (Kaur A et al (2010)). We need the computing platform that is based on the policy of “Pay as you use” and also leads to fuller CPU Utilization with more security. So the new concept or computing platform emerges into the market that is known as the cloud computing. The strong degree of equivalency between "function points" and "SLOC" shown in the paper suggests a two-step work-effort validation procedure, first using "function points" to estimate "SLOC," and then using "SLOC" to estimate the work-effort. This approach would provide validation of application development work plans and work-effort estimates early in the development cycle. The approach would also more effectively use the existing base of knowledge on producing "SLOC" until a similar base is developed for function points (Allan J. Albrecht (1983)). Of the many existing and proposed techniques for software development, it seems clear that component-based software development will be at the forefront of new approaches to the production of software systems and holds the promise of substantially enhancing the software
production and maintenance process. Attempts to rationalize component-based development have to recognize that the construction of a software system is a complex multifaceted activity that involves domain engineering, frame working, assembling, archiving and design of software components. These activities, among others, are encompassed by a software life cycle, named the Y model, put forward in this study. The Y model provides guidance for the major phases to be followed under its umbrella ([Fernando L Capretz (2005)]). Software Verification and Validation (V & V) activities check the software against its specifications. In traditional software, V & V could be done in close cooperation with the customer to meet the specific requirements from the customer. When developing software components, or Component-Based Software Systems (CBSS), V & V becomes a bit more complicated. Components developed for reuse, and especially components developed for the open market, have to be more thoroughly specified and verified than most custom software for many reasons. The authors have presented V & V when developing software components which are helpful in improving the functionality and quality of component and Component-Based System (CBS) by using a new X Component-Based Model ([Tomar P et al (2010)]). Program slicing, introduced by Weiser, is known to help programmers in understanding foreign code and in debugging. Using the decomposition, we provide a set of principles to prohibit changes which will interfere with unmodified components. These semantically consistent changes can then be merged back into the original program in linear time. Moreover, the maintainer can test the changes in the component with the assurance that there are no linkages into other components. Thus decomposition slicing induces a new software maintenance process model which eliminates the need for regression testing ([Gallagher K Brian et al (1991)]). The essence of software re-engineering is to improve or transform existing software so that it can be understand, controlled, and used anew. The need for software re-engineering has increased greatly, as heritage software systems have become obsolescent in terms of their architecture, the platforms on which they run, and their suitability and stability to support evolution to support changing needs ([Linda H. Rosenberg (2009)]). This article categorizes and examines a number of methods for describing or modeling how software systems are developed. It begins with background and definitions of traditional software life cycle models that dominate most textbook discussions and current software development practices. This is followed by a more comprehensive review of the iterative models of software evolution that are of current use as the basis for organizing software engineering projects and technologies ([Scacchi W (2001)]). The author in this paper presents a conceptual foundation for software re-engineering. The foundation is composed of properties and principles that underlie reengineering methods, and assumptions about reengineering. A general model of software re-engineering is established, based on this foundation. This model, along with its underlying foundation, proves useful for examining re-engineering issues such as the reengineering process and re-engineering strategies ([Eric J. Byrne (1992)]). The authors characterize the problems and their impact on adoption. In addition, and equally importantly, he describes how the combination of existing research thrusts has the potential to alleviate many of the concerns impeding adoption ([Chow R et al (2009)]). Using the Amazon EC2 service as a case study, the author has shown that it is
possible to map the internal cloud infrastructure, identify where a particular target VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target. He explores how such placement can then be used to mount cross-VM side-channel attacks to extract information from a target VM on the same machine (Risten T. Part(2009)). The author explains the new risks that face administrators and users (both image publishers and image retrievers) of a cloud's image repository. To address those risks, he proposes an image management system that controls access to images, tracks the provenance of images, and provides users and administrators with efficient image filters and scanners that detect and repair security violations (Wei J (2009)). The authors present the seven architectural principles and derive ten interconnected architectural modules to form a reusable and customizable Cloud Computing Open Architecture (CCOA). Two case studies on Infrastructure and Business Cloud are used to deliver business and practical value of infrastructure and business process provisioning services over the Internet. We also present some potential value added services of the proposed CCOA to guide strategic planning and other consulting practices of Cloud Computing (Jie L. Zhang et al (2009)). What differences are there for different cloud computing platforms and what characteristics and advantages each has? To answer these problems, the characteristics, architectures and applications of several popular cloud computing platforms are analyzed and discussed in detail. From the comparison of these platforms, users can better understand the different cloud platforms and more reasonably choose what they want (Peng J et al (2009)). Our goal in this article is to reduce that confusion by clarifying terms, providing simple figures to quantify comparisons between of cloud and conventional computing, and identifying the top technical and non-technical obstacles and opportunities of cloud computing (Armbrust M(2010)). Our idea of Cloud proposes a new dimension of computing, in which everyone, from single users to communities and enterprises, can, on one hand, share resources and services in a transparent way and, on the other hand, have access to and use such resources and services adaptively to their requirements. Such an enhanced concept of Cloud, enriching the original one with volunteer computing and interoperability challenges, has been proposed and synthesized in Cloud@Home (Cunsolo V. D et al (2010)). The author propose a novel approach to reengineering enterprise software for cloud computing by building ontology for enterprise software and then partitioning the enterprise software ontology to decompose enterprise software into potential service candidates. Ontology development process includes three steps, namely, building ontologies for source code, data, and application framework respectively, integrating captured ontologies and deploying the final produced ontology. Firstly, the ontology development process is supported by the reverse engineering and model transformation techniques. Secondly, the ontology integration is based on ontology engineering research. Thirdly, the deployment of enterprise software ontology is done through the software reengineering activities. Once the ontology is built, there will be a link between ontology and enterprise software. By analyzing the concepts and relations in ontology, the enterprise software will be understood and decomposed as different service candidates (Zhou H et al (2010)). Author aim is to pinpoint the Challenges and issues of Cloud computing. We first discuss two
related computing paradigms - Service-Oriented Computing and Grid computing, and their relationships with Cloud computing. We then identify several challenges from the Cloud computing adoption perspective. Last, we will highlight the Cloud interoperability issue that deserves substantial further research and development (Dillon T et al(2010)). This article introduces the background and principle of cloud computing, the character, style and actuality. This article also introduces the application field the merit of cloud computing, such as, it do not need user’s high level equipment, so it reduces the user’s cost. It provides secure and dependable data storage center, so user needn’t do the awful things such storing data and killing virus, this kind of task can be done by professionals. It can realize data share through different equipments. It analyses some questions and hidden troubles, and puts forward some solutions, and discusses the future of cloud computing (Zhang S et al (2010)). In order to make clear the essential of cloud computing, we propose the characteristics of this area which make cloud computing being cloud computing and distinguish it from other research areas (Gong C et al (2010)). The current cloud implementations are often isolated from other cloud implementations. The author gives an overview survey of current cloud computing architectures, discusses issues that current cloud computing implementations have and proposes a Service-Oriented Cloud Computing Architecture (SOCCA) so that clouds can interoperate with each other. Furthermore, the SOCCA also proposes high level designs to better support multi-tenancy feature of cloud computing (Wei-Tek Tsai*, Xin Sun (2010)). Architectures must also encompass the software and devices that users utilize in order to invoke functions in the cloud, and intermediary functions. A further problem with analyses to date is inadequate reflection of the risks that users are subject to when they use cloud services. Author proposes a comprehensive model that reflects user needs, and identifies implications of the model for computer scientists working in the area (Clarke R (2010)). Service Developers want the Service Providers to ensure or provide the capability to dynamically allocate and manage resources in response to changing demand patterns in real-time. Ultimately, Service Providers are under pressure to architect their infrastructure to enable real-time end to end visibility and dynamic resource management with fine grained control to reduce total cost of ownership while also improving agility (Sarathy V et al (2010)). The author proposes P2P Cloud architecture to improve on the cloud computing infrastructure which contains Center P2P Network and Side P2P Network. 1) Center P2P which replace Master of centralized architecture can avoid bottleneck of the centralized architecture, provide computing service to client using Center P2P Server and increase the number of customers to access the system. 2) Side P2P Network which replace Slave of the centralized architecture would provide computing and storage capability, which can improve the transmission of data and the performance of the system effectively. Through the analysis and evaluation, this system has better availability and scalability (Peng Z (2010)). The authors in this article introduces the application field the merit of cloud computing, such as, it do not need user’s high level equipment, so it reduces the user’s cost. It provides secure and dependable data storage center, so user needn’t do the awful things such storing data and killing virus, this kind of task can be done by professionals (Zhang S et Al (2010)). The author present a scheduling strategy on load
balancing of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed VM resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling. Experimental results prove that this method is able to realize load balancing and reasonable resources utilization both when system load is stable and variant (Jinhua Hu, Jianhua Gu et Al (2010)). The author deals with the various methodologies adopted to handle all the processes and jobs concurrently executing and waiting into the web application and web server housed into the same system or different systems. Also, these different methods will be compared taking into account the same number of jobs, but varied environmental conditions and hence, the result would be formulated. Various issues like virtual resources, queuing strategies, resource managers etc. has been discussed here apart from the main coverage points. All these aspects will be closely studied, observed and proved with proper explanations (Gupta P K. et Al (2010)).

Cloud computing is the product of the fusion of traditional computing technology and network technology like grid computing, distributed computing parallel computing and so on. It aims to construct a perfect system with powerful computing capability through a large number of relatively low-cost computing entity, and using the advanced business models like SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service) to distribute the powerful computing capacity to end users’ hands. This article introduces the background and service model of cloud computing. Author introduces the existing issues in cloud computing such as security, privacy, reliability and so on. Proposition of solution for these issues has been provided also (Yang J (2010)). The authors discuss security issues for cloud computing and present a layered framework for secure clouds and then focus on two of the layers, i.e., the storage layer and the data layer. In particular, the authors discuss a scheme for secure third party publications of documents in a cloud. Next, the paper will converse secure federated query processing with map Reduce and Hadoop, and discuss the use of secure co-processors for cloud computing. Finally, the authors discuss XACML implementation for Hadoop and discuss their beliefs that building trusted applications from untrusted components will be a major aspect of secure cloud computing (Hamlen K (2010)). The authors have explored eight multidisciplinary issues affecting security management from a business perspective: 1) who is responsible for security, 2) Cloud services provider transparency, 3) how penetration tests are conducted, 4) what happens when a cloud computing service provider goes bankrupt or is acquired by another company, 5) how we gather forensic evidence in the case of a breach, 6) what are the hypervisor vulnerabilities, 7) what problems derive from a layered cloud architecture, and 8) lack of direct experience. We will examine the many security issues involved with cloud computing and offer insight into how to deal with these issues (Lumley R A (2010)). Currently, Google is the largest search engine and Facebook is the largest social network in the Software as a Service (SaaS). But how them can support the huge requests from world thought each personal computer, mobile device, and smart phone. The
author in this paper, we will try to analysis their backend cloud computing architecture to support future SaaS especially in large social network (Yang B W et Al (2011)). Interoperability is central to enabling sharing of resources from a pool of cloud-service providers in a seamless fashion. In this paper the author describes some of the challenges in achieving interoperability for cloud computing and recommend an adaptation of the U.S. Department of Defense’s LISI Maturity Model to address cloud-to-cloud interoperability (Barreto A et al (2011)). The author in this paper outlines what cloud computing is, the various cloud models and the main security risks and issues that are currently present within the cloud computing industry. This research paper also analyzes the key research and challenges that presents in cloud computing and offers best practices to service providers as well as enterprises hoping to leverage cloud service to improve their bottom line in this severe economic climate (Padhy R P (2011)). The author presents the findings from the points of view of a cloud service provider, cloud consumer, and third-party authorities such as Govt. We also discuss important research directions in cloud security in areas such as Trusted Computing, Information Centric Security and Privacy Preserving Models. Finally, we sketch a set of steps that can be used, at a high level, to assess security preparedness for a business application to be migrated to cloud (Sengupta S (2011)). The author summarizes reliability, availability, and security issues for cloud computing (RAS issues), and propose feasible and available solutions for some of them (Sabahi F (2011)). The computer hardware of today’s world was designed and architected for the purpose of running a single operating system and its applications and therefore most of the hardware of such computer would be left underutilized. Virtualization enables the option of running multiple virtual computers on a single physical system. The authors have shown the various methods, through which virtualization is supported in computer systems. The impact of this solution on computer hardware architecture is also investigated (Semnanian A A et al (2011)). Cloud based development is a challenging task for several software engineering projects, especially for those which needs development with reusability. Present time of cloud computing is allowing new professional models for using the software development. The expected upcoming trend of computing is assumed to be this cloud computing because of speed of application deployment, shorter time to market, and lower cost of operation. Until Cloud Computing Reusability Model is considered a fundamental capability, the speed of developing services is very slow. The author in this paper spreads cloud computing with component based development named Cloud Computing Reusability Model (CCR) and enable reusability in cloud computing. In this paper Cloud Computing Reusability Model has been proposed. The model has been validated by Cloudsim and experimental result shows that reusability based cloud computing approach is effective in minimizing cost and time to market (Singh S P et al (2012)). The author in this paper briefly describes traditional re-engineering then discusses the emerging process of hybrid-reengineering which is often used as means to simplify the cumbersome tasks. The paper represents how maintenance is going to be effect with the help of given software engineering approaches. An analysis of various possible risks, their impact and mapping with various attributes is correspondingly depicted. This paper is also presenting the way to reduce the impact
of most of these risks by using hybrid re-engineering (Tarar S et al (2012)). The software architect who is well-known in markup language i.e. HTML can create the design and further convert into implementation phase. Here the lack of design time will be reduced by a special technique called as Re-Engineering. After converting the application, the Organization has to maintain the architecture of their work process. At that situation this Cloud Computing is used to integrate into our components. The cloud computer has many factors that contribute to the success and survival of the company during transition, one of them is to assess learning curves of many different individuals. The author in this paper aims to shed light on the realities of screen scraping and discuss some of the possibilities and limitations of automated language converters (Kirubakaran E. et Al (2012)). Cloud computing has become an important platform for companies to build their infrastructures upon. If companies are thinking to take advantage of cloud based systems, they will have to seriously reassess their current security strategies as well as the cloud-specific aspects to be a successful solution provider. The author’s focus of this study, based on existing literature, is to define a methodology for cloud providers that will protect users’ data, information which is of high importance (Malik A et al (2012)). Accurate estimation of project costs is an essential prerequisite to making a reengineering project. Existing systems are usually reengineered because it is cheaper to reengineer them than to redevelop or to replace them. However, to make this decision, management must know what the reengineering will cost. This contribution describes an eight step tool supported process for calculating the time and the costs required to reengineer an existing system. The process is derived from the author’s 20 year experience in estimating reengineering projects and has been validated by several real life field experiments in which it has been refined and calibrated (Sneed H M (2005)).