Abstract - The new concept of cloud computing with on-demand provisioning of a shared pool of configurable computing resources of unprecedented size, elastic scalability and ubiquitous network service accessibility has gained huge momentum in the IT industry. However, there are many issues faced in migrating existing systems into the cloud that have led to its slow adoption. This paper focuses on the lack of user-centric architecture, and identifies the traditional system design, implementation and testing issues prevalent in engineering the cloud services for migration, in particular interoperability and extensibility issues with the SaaS. Further, the paper aims to propose a software engineering research roadmap to address these issues.

Software systems and infrastructure services based on traditional software engineering principles and service-oriented architecture (SOA) require suitable adaptations due to multi-tenancy concept of the cloud as compared to single ownership of existing systems and services. There is paucity of literature focussed on this topic, which forms the motivation for this paper. With a modest step forward, we identify two major contemporary issues of cloud computing, namely architecture-based and interoperability-based issues that affect cloud migration, and relate these issues to the roots of software engineering with specific research topics. In a nutshell, we identify that developing cloud architecture based on user-roles, and formulating new software engineering theory in the design, testing and process management of SaaS for cloud migration require prime attention. We provide a roadmap consisting of four main steps that provide focussed research opportunities for software engineering researchers and experts to engineer the cloud services towards successful and sustainable adoption for the future.

Keywords: Cloud computing, Software engineering, Service oriented architecture (SOA), Research roadmap.

1 INTRODUCTION

Cloud has become the recent information technology buzzword today. However, cloud computing is not a technology, rather is essentially a new concept of distributed computing that enables convenient on-demand access to a pool of shared computing resources, which can be rapidly configured and provisioned [1]. Since the quick growth of web services, IT companies have started to focus on integrating their web services that composites a set of powerful computing resources. These powerful computing resources present the “cloud” of today. However, there exists various security and privacy issues surrounding both technological and non-technological aspects of the cloud that have resulted in their slow adoption [2-3]. While some of these challenges are being tackled from technical and management perspectives, the fundamental problem lies in the lack of suitable software engineering adaptations in the cloud computing paradigm [4-5]. This paper identifies such challenges and issues involved in bringing existing software systems into the cloud, since such systems are predominantly developed based on service-oriented architecture (SOA), which have to be adapted and extended suitably for the cloud. Our premise is that these challenges lead to opportunities for research and development in software engineering that require attention and a research roadmap for realising the benefits of the cloud computing concept.

The recent developments in Web technologies, including Web 2.0 and SOA protocols and standards should be considered in re-engineering cloud computing [4-5]. While Internet and Grid networks have achieved the maturity in high-end anywhere and anytime distributed computing, cloud computing has the capability of providing quality user-centric services in a cost-effective manner. This allows small and large organisations to now host their data and business application centres virtually in the cloud without having to invest largely on IT infrastructure and resources. However, for the existing data, workflow and business applications to be migrated into the cloud, there are software engineering challenges that need to be addressed first [5].
Due to the unprecedented scale and heterogeneity of the required infrastructure, cloud computing poses a variety of challenges to conventional software engineering principles adopted in the existing systems [6]. Some challenges that are typical of any new information system paradigm would cover the technological issues, while others are non-technological in nature [7]. In this context, there is a need to rethink the software engineering principles that have traditionally been focused on behind Grid, SOA and Web technologies for a better realisation of services in cloud systems [8]. This demands setting up a research roadmap to address both technological and non-technological issues that relate to topics for research in software engineering, which could encompass the elastic scalability, programming models, data management, systems life cycle management, privacy and security, economic models, governance and legislation of the cloud systems.

Lack of proper knowledge of the cloud architecture, types, models and user-roles indicating that problems in deep understanding of the challenges involved have impacted very much on the cloud adoption [9-10]. Hence, the aim of this paper is to identify the technological and non-technological issues present in the current cloud models that relate to the software engineering design and implementation principles and practices, which require to be extended suitably for the cloud. In addition, we describe the research topics for the identified software engineering issues that could be addressed suitably using our proposed roadmap.

Literature review of the research and practice taking place around cloud computing provides insights into the various challenges that place cloud services at an early adoption stage [2-3, 7]. We have identified a number of questions related to cloud services that play an important role in their adoption and still remain unanswered. Some of these unanswered questions are listed below:

- What are the cloud models offered in the market?
- What are the software engineering issues related to the technological and non-technological concerns present in the cloud agenda worldwide?
- How can each service provider’s research perspectives and cloud agenda influence the migration to cloud services?
- How can each country’s strategic cloud agenda influence the global cloud adoption?
- What are the research initiatives that could be undertaken by software engineering experts to address the issues present in the cloud agenda?

- What are the open software engineering research challenges in migrating to the cloud?
- How can we establish a software engineering research roadmap for the future of the cloud?

The aim of this paper is to address the above questions to some extent as a first step towards understanding the underlying cloud concepts, trends and standards from software engineering perspective of migration to the cloud. The main contribution would be to set a research roadmap to address the software engineering challenges of migrating existing systems to the cloud. To achieve our aim, we first present in section 2, a typical cloud architecture and various types of clouds, user-roles, cloud deployment models, and cloud service levels along with the potential benefits and risks. The main motivation for this section is to set the platform for understanding the various user-centric data and service engineering issues prevalent around the cloud. The paper then discusses in section 3, the various countries’ cloud agenda, and the strategies and perspectives of cloud providers, who form the big players for cloud research and development. In section 4, we identify the various open research challenges pertaining to software engineering that have an influence in the migration of the existing systems into the cloud. To address these challenges, we provide in section 5 recommendations in the form of a software engineering research roadmap for the future of cloud computing that would tap on new software engineering models and practices. Finally, section 6 provides conclusions and future research that highlights new information-centric research capabilities in software engineering which simplify the combination of data, instruments, computing and analysis of various applications that can enable research workflows accessing multiple resources.

2 CLOUD MODELS AND SOFTWARE ENGINEERING ISSUES

A cloud is a dynamic provision of computing services/ resource pools in a co-ordinated fashion. In this section, we describe the basic background information about cloud architecture, types and models to understand the user interface and their roles associated with a typical cloud. The cloud interfaces through which users interact with the specific elements in the cloud architecture and the user-roles associated to manage and control them are based on the type of cloud and service level to which they are authorised and assigned [11-12]. We lay a foundation for a deep understanding of these important facets that facilitate our aims to identify software engineering issues and challenges for suitable adaptation towards cloud computing.

2.1 Cloud Architecture

Currently there are various organisations, called Cloud Service Providers, including brokers offering cloud services. The typical cloud architecture involves multiple cloud components communicating with each other over application programming interfaces that are usually available as web services. This cloud architecture expands to the clients, which are typically web browsers and/or software applications that could have access to the cloud applications. Figure 1 show a typical cloud architecture, where end-users access the cloud from any remote computing portable devices via the Internet. We call such end-users, which includes individual users as well as business adopters as Cloud Consumers. For the cloud consumers, the cloud is considered as on-demand computing without any knowledge of the location of the resources or where the applications are run. The hardware in the cloud (and the operating system that manages the ‘physical connection’ is invisible). However, the cloud architecture allows application developers as well as regulators and auditors to develop, deploy and run applications standards and policies that can easily grow in capacity (scalability) work in real-time (performance), and offer good reliability and policy mechanisms. We term such users of the cloud as Cloud Architects.

The cloud architecture also allows dynamically updates of the infrastructure elements without affecting the business. All the three types of users, namely Cloud Service Providers, Cloud Architects and Cloud Consumers have different purposes for interacting and interfacing with the cloud, and these are determined by the cloud type, deployment model and service levels used [13-14]. In general, the proposed user roles associated with the cloud architecture results in identifying certain software engineering challenges for cloud adoption. We identify the first issue in current cloud architecture as given below.

**Issuel: Lack of Non-Functional User Service Architecture -** The main concern found in current cloud computing framework (Figure 1) is that it is primarily resource-centric and does not cater to the non-technological service-centric aspects such as legalistic issues, economic impacts and green IT needs of the cloud users. This has a great impact from software engineering viewpoint since in traditional information systems, non-technological or ‘non-functional’ aspects of any system design plays an important role in user acceptance. Our premise is that a cloud computing reference architecture should define the key players responsible for the cloud evolution, and more importantly, identify within their roles the non-technological service-centric aspects as well. These attribute to challenges in adapting traditional software engineering system architecture and process management that relate to different cloud users.
Due to the unprecedented scale and heterogeneity of the required infrastructure, cloud computing poses a variety of challenges to conventional software engineering principles adopted in the existing systems [6]. Some challenges that are typical of any new information system paradigm would cover the technological issues, while others are non-technological in nature [7]. In this context, there requires a rethinking to exploit the software engineering principles behind Grid, SOA and Web technologies for a better realisation of services in cloud systems [8]. This demands setting up a research roadmap to address both technological and non-technological issues that relate to topics for research in software engineering, which could encompass the elastic scalability, programming models, data management, systems life cycle management, privacy and security, economic models, governance and legislation of the cloud systems.

Lack of proper knowledge of the cloud architecture, types, models and user-roles indicating that problems in deep understanding of the challenges involved have impacted very much on the cloud adoption [9-10]. Hence, the aim of this paper is to identify the technological and non-technological issues present in the current cloud models that relate to the software engineering design and implementation principles and practices, which require to be adapted and extended suitably for the cloud. In addition, we describe the research topics for the identified software engineering issues that could be addressed suitably using our proposed roadmap.

Literature review of the research and practice taking place around cloud computing provides insights into the various challenges that place cloud services at an early adoption stage [2-3, 7]. We have identified a number of questions related to cloud services that play an important role in their adoption and still remain unanswered. Some of these unanswered questions are listed below:

- What are the cloud models offered in the market?
- What are the software engineering issues related to the technological and non-technological concerns present in the cloud agenda worldwide?
- How can each service provider’s research perspectives and cloud agenda influence the migration to cloud services?
- How can each country’s strategic cloud agenda influence the global cloud adoption?
- What are the research initiatives that could be undertaken by software engineering experts to address the issues present in the cloud agenda?
- What are the open software engineering research challenges in migrating to the cloud?
- How can we establish a software engineering research roadmap for the future of the cloud?

The aim of this paper is to address the above questions to some extent as a first step towards understanding the underlying cloud concepts, trends and standards from software engineering perspective of migration to the cloud. The main contribution would be to set a research roadmap to address the software engineering challenges of migrating existing systems to the cloud. To achieve our aim, we first present in section 2, a typical cloud architecture and various types of clouds, user-roles, cloud deployment models, and cloud service levels along with the various user-centric data and service engineering issues prevalent around the cloud. The paper then discusses in section 3, the various countries’ cloud agenda, and the strategies and perspectives of cloud providers, who form the big players for cloud research and development. In section 4, we identify the various open research challenges pertaining to software engineering that have an influence in the migration of the existing systems into the cloud. To address these challenges, we provide in section 5 recommendations in the form of a software engineering research roadmap for the future of cloud computing that would tap on new software engineering models and practices. Finally, section 6 provides conclusions and future research that highlights new information-centric research capabilities in software engineering which simplify the combination of data, instruments, computing and analysis of various applications that can enable research workflows accessing multiple resources.

2 CLOUD MODELS AND SOFTWARE ENGINEERING ISSUES

A cloud is a dynamic provision of computing services/resource pools in a co-ordinated fashion. In this section, we describe the basic background information about cloud architecture, types and models to understand the user interface and their roles associated with a typical cloud. The cloud interfaces through which users interact with the specific elements in the cloud architecture and the user-roles associated to manage and control them are based on the type of cloud and service level to which they are authorised and assigned [11-12]. We lay a foundation for a deep understanding of these important facets that facilitate our aims to identify software engineering issues and challenges for suitable adaptation towards cloud computing.

2.1 Cloud Architecture

Currently there are various organisations, called Cloud Service Providers, including brokers offering cloud services. The typical cloud architecture involves multiple cloud components communicating with each other over application programming interfaces that are usually available as web services. This cloud architecture extends to the clients, which are typically web browsers and/or software applications that could have access to the cloud applications. Figure 1 show a typical cloud architecture, where end-users access the cloud from any remote computer or portable devices via the Internet. We call such end-users, which includes individual users as well as business adopters as Cloud Consumers. For the cloud consumers, the cloud is considered as on-demand computing without any knowledge of the location of the resources or where the applications are run. The hardware in the cloud (and the operating system that manages the hardware) is invisible. However, the cloud architecture allows application developers as well as regulators and auditors to develop, deploy and run applications standards and policies that can easily grow in capacity (scalability) work in real-time (performance), and offer good reliability and policy mechanisms. We term such users of the cloud as Cloud Architects.

The cloud architecture also allows dynamically updates of the infrastructure elements without affecting the business. All the three types of users, namely Cloud Service Providers, Cloud Architects and Cloud Consumers have different purposes for interacting and interfacing with the cloud, and these are determined by the cloud type, deployment model and service levels used [13-14]. In general, the proposed user roles associated with the cloud architecture results in identifying certain software engineering challenges for cloud adoption. We identify the first issue in current cloud architecture as given below.

Issue 1: Lack of Non-Functional User Service Architecture - The main concern found in current cloud computing framework (Figure 1) is that it is primarily resource-centric and does not cater to the non-technological-service-centric aspects such as legalistic issues, economic impacts and green IT needs of the cloud users. This has a great impact from software engineering viewpoint since in traditional information systems, non-technological or ‘non-functional’ aspects of any system design plays an important role in user acceptance. Our premise is that a cloud computing reference architecture should define the key players responsible for the cloud evolution, and more importantly, identify within their roles the non-technological service-centric aspects as well. These attribute to challenges in adapting traditional software engineering system architecture and process management that relate to different cloud users.

The cloud architects would include financial and system auditors, government bodies, intermediaries, law enforcement agencies or standards regulators ensuring information control and governance through policies, negotiations and standards pertaining to information format, access, privacy and other issues. They would be responsible in applying the software engineering principles during the system analysis and design in addressing the legal issues that would cover detailed location handling of the cloud, data protection and governance policies.

Currently, the cloud architecture lacks appropriate software engineering approach to address the non-functional process and management issues that are user-centric. To some extent National Institute of Standards and Technology (NIST) has formulated a cloud architecture defining five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor, and cloud broker [1]. However, the role of a cloud developer and policy architect is not very clear here. There are other consortiums that combine the terms of cloud actors along with their roles into a common term as ‘user-roles’. With this view, IBM has recently developed a set of three user-roles, namely cloud service creator, provider, and consumer [15]. The main drawback in such user models is that governance policies and other non-technical aspects are not being given the deserving and explicit importance. Our proposed user model (Figure 2) consists of three key players, namely Cloud Service Providers.
Cloud Architects and Cloud Consumers who play a pivotal role in addressing the software engineering challenges that are both technological as well as non-technological in achieving cloud adoption. Our main aim to address this issue is to give importance for the key player to include the technological roles of the developers of a cloud service along with the non-technological roles of regulatory parties and auditors. We have termed such a category of key players as 'Cloud Architects'.

There are four main deployment models of the cloud architecture, namely public, private, hybrid and community clouds. When a cloud is made available to the general sector's requirements. Programming models need to be revisited to include legalistic aspects, sound security and non-technological pressures that surround cloud adoption could be addressed. Cloud consumers, be it individual users, business users, government users or research users, would require different types of services based on their dynamically changing requirements. Cloud services should be able to deliver their SaaS, PaaS and IaaS to any type of cloud consumer environment, catering to consumer needs. We identify technological challenges as given below.

3 provides a list of typical cloud services that could be offered in any of the cloud deployment models, namely public, private, hybrid and community clouds. The cloud services are grouped under three main service levels, namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Currently, we find that not all big providers provide all these three levels of service offerings to an extent that is meeting the demand of all types of users. These three service levels are described below.

Cloud Infrastructure as a Service (IaaS). IaaS is the capability provided to the consumer for providing computer processing, storage, networks, and other fundamental computing resources, along with software, which can include operating systems and applications. The consumer may be allowed with limited rights to manage or control the provisioned cloud infrastructure. A typical IaaS level of cloud service offers various server, storage, virtualization and networking components and the associated operational software such as, operating systems, file systems, virtualization technologies, etc. It allows users to operate as if it were their own data centres with resource provisioning made on demand. These pose further privacy issues due to multi-tenancy and multi-vendors as compared to traditional infrastructure services.

Cloud Software as a Service (SaaS). SaaS is a capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. While consumers could manage the user-specific application configuration settings, they have no control on the underlying cloud infrastructure (network, servers, operating systems, storage, and applications). Typically, the SaaS level of cloud service offers various software applications to meet the needs of different users over the Web. These software could typically have different ownership which is different from traditional software engineering theory of single ownership.

Cloud Platform as a Service (PaaS). PaaS is a capability provided to the consumer to deploy applications created or acquired onto the cloud infrastructure. While the consumers have control over the deployed applications and their hosting environment configurations, they do not manage the underlying cloud infrastructure. The PaaS is a typical infrastructure platform for different software developers to develop applications and deploy them over the Web. It provides typical software tools such as virtualisation, database, middleware and programming compilers that are required for an application development life cycle, without the need to buy and maintain them. Suitable integration and interfacing of multi-systems with existing systems could be a challenge.

Issue 2: Lack of Trusted Interoperability and Migration of Service Technologies

As seen in Figure 2, both technological and non-technological issues are identified as cloud migration issues. The software engineering challenges for the migration of existing systems to the cloud exist. There are software engineering challenges to cope with inter-operable cloud deployment market place due to dynamic changes in the cloud service providers and their service provisions. We need to next explore the impact on the cloud service levels offered by them.

2.3 Cloud Service Levels

Cloud services refer to not only software applications (predominantly called as Web services) that are available over the Internet, but also encompass the hardware and systems services available in remote data centres. Figure

FIGURE 2. PROPOSED USER MODEL FOR CLOUD ADOPTION

2.2 Cloud Deployment Models

There are four main deployment models of the cloud architecture, namely public, private, hybrid and community clouds. When a cloud is made available to the general public or organisations with services rented in a pay-as-you-go manner such a cloud is called a Public Cloud or external cloud. The best known example of such a public cloud service is by Amazon Elastic Computing Cloud EC2 [16] that offer several different in-the-cloud services. EC2 has key features that include: elasticity, control, and flexibility. Other Amazon services include Amazon Simple Storage Service (S3), Simple DB, Cloudfront, Simple Queue Service (SQS), and Elastic MapReduce. EC2 supports both pure cloud-delivered offerings and hybrid, on-premises plus cloud scenarios which will be common options for the cloud consumers.

In order to cater to data privacy and certain internal local data operations, organisations may operate on their own existing data centres with heavily invested infrastructure. Such services do not cater to the general public but are offered only to their business branches or business partners in the form of a cloud and it forms a Private Cloud or internal cloud. The cloud infrastructure is operated solely for an organization. Since the cloud is managed by the organization, there is fine grained control over the resources and services in an extent that is meeting the demands of all types of users.

EC2 supports both pure cloud-delivered offerings and hybrid, on-premises plus cloud scenarios which will be common options for the cloud consumers. We have termed such a category of key players as 'Cloud Architects'.
Cloud Architects and Cloud Consumers who play a pivotal role in addressing the software engineering challenges that are both technological as well as non-technological in achieving cloud adoption. Our main aim to address this issue is to give importance for the key player to include the technological roles of the developers of a cloud service along with the non-technological roles of regulatory parties and auditors. We have termed such a category of key players as ‘Cloud Architects’.

2.2 Cloud Deployment Models

There are four main deployment models of the cloud architecture, namely public, private, hybrid and community clouds. When a cloud is made available to the general public or organisations with services rented in a pay-as-you-go manner such a cloud is called a Public Cloud or external cloud. The best known example of such a public cloud service is by Amazon Elastic Computing Cloud EC2 [16] that offers different in-the-cloud services. EC2 has key features that include: elasticity, control, and flexibility. Other Amazon services include Amazon Simple Storage Service (S3), Simple DB, Cloudfront, Simple Queue Service (SQS), and Elastic MapReduce. EC2 supports both pure cloud-delivered offerings and hybrid, on-premises plus cloud scenarios which will enable common options for the cloud consumers.

In order to cater to data privacy and certain internal local data operations, organisations may operate on their own existing data centres with heavily invested infrastructure. Such services do not cater to the general public but are offered only to their business branches or business partners in the form of a cloud and it forms a Private Cloud or internal cloud. The cloud infrastructure is operated solely for an organization. Since the cloud is managed by the organization, there is fine grained control over the resources of the service and any issues that could arise. They are more secure since operated internal to the organisation. Some large organisations serve as internal service providers by hosting such private cloud services on their private networks so that their corporate partners and in-house staff are able to make use of hardware and software resources without having to purchase these for their own departments.

A private cloud does not satisfy the true objectives of cloud services and typically organisations may wish to have a combination of both external as well as internal service providers, which leads to a Hybrid Cloud. In such cases, the cloud infrastructure is a composition of two or more clouds (private or public) that operate using standardized or proprietary technology enabling data and application portability.

Lastly, there are instances where a group of organisations having similar interests and requirements may form a cloud so as to share the infrastructure, resources and services. Such a cloud is called a Community Cloud. Here, the cloud infrastructure is shared by several organizations and supports a specific community that has common concerns and mission. It may be managed by the organisations or a third party. Google’s Gov Cloud [17] is a typical example of a community cloud, which is tailored to meet public sector’s requirements.

Whether the cloud is deployed as private cloud, public cloud, hybrid cloud or community cloud, software engineering challenges for the migration of existing systems to the cloud exist. There are software engineering challenges to cope with inter-operable cloud deployment market place due to dynamic changes in the cloud service providers and their service provisions. We need to next explore the impact on the cloud service levels offered by them.

2.3 Cloud Service Levels

Cloud services refer to not only software applications (predominantly called as Web services) that are available over the Internet, but also encompass the hardware and systems services available in remote data centres. Figure 3 provides a list of typical cloud services that could be offered in any of the cloud deployment models, namely public, private, hybrid and community clouds. The cloud services are grouped under three main service levels, namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Currently, we find that not all big providers provide all these three levels of service offerings to an extent that is meeting the demand of all types of users. These three service levels are described below.

Cloud Infrastructure as a Service (IaaS). IaaS is the capability provided to the consumer for providing computer processing, storage, networks, and other fundamental computing resources, along with software, which can include operating systems and applications. The consumer may be allowed with limited rights to manage or control the provisioned cloud infrastructure. A typical IaaS level of cloud service offers various servers, storage devices and networking components and the associated operational software such as, operating systems, file systems, virtualization technologies, etc. It allows users to operate as it were their own data centres with resource provisioning made on demand. These pose further privacy issues due to multi-tenancy and multi-vendors as compared to traditional infrastructure services.

Cloud Software as a Service (SaaS). SaaS is a capability provided to the consumer to use the provider’s applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. While consumers could manage the user-specific application configuration settings, they have no control on the underlying cloud infrastructure (network, servers, operating systems, storage, and applications). Typically, the SaaS level of cloud service offers various software applications to meet the needs of different users over the Web. These software could typically have different ownership which is different from traditional software engineering theory of single ownership.

Cloud Platform as a Service (PaaS). PaaS is a capability provided to the consumer to deploy applications created or acquired onto the cloud infrastructure. While the consumers have control over the deployed applications and their hosting environment configurations, they do not manage the underlying cloud infrastructure. The PaaS is a typical infrastructure platform for different software developers to develop applications and deploy them over the Web. It provides typical software tools such as virtualisation, database, middleware and programming compilers that are required for an application development life cycle, without the need to buy and maintain them. Suitable integration and interfacing of multi-systems with existing systems could be a challenge.

Cloud as a Service (CAS). CAS is the capability provided to the consumer for consuming services from various public, private, hybrid and community clouds. These services could be consumed via a standardised API interface, including software, infrastructure, and associated operational software such as, operating systems, file systems, virtualization technologies, etc. It allows users to operate as it were their own data centres with resource provisioning made on demand. These pose further privacy issues due to multi-tenancy and multi-vendors as compared to traditional infrastructure services.
achieve substantial efficiency that would result in improved cloud adoption. These attribute to challenges in adapting traditional software engineering design, implementation and testing methodologies into the cloud paradigm.

Overall, the lack of functional migration of systems (Issue 2) and the lack of non-functional user-centric architecture (Issue 1) have posed technological and non-technical pressures in the cloud adoption. Figure 4 provides our proposed cloud service framework, in which best practices of SOA, Web 2.0 and Grid computing could be leveraged to address both these issues faced by SaaS, PaaS and IaaS of cloud computing. This way the cloud architects (that includes cloud developers as well as policy regulators), and cloud service providers (that includes network providers as well as data centre service providers and brokers) would be able to address the software engineering challenges faced by cloud computing. It also includes clearly defined roles of cloud consumers in alleviating the technological and non-technical pressures for their seamless cloud migration.

Having identified the two major issues in the current cloud paradigm, it is also important to review the strategies and perspectives of cloud implementations from different countries’ cloud agenda. This section describes the cloud strategies and perspectives of major countries and cloud providers that would play an influential role in this context.

3.1 Countries’ Cloud Agenda

Cloud computing is being debated as the next technological disruption to transform enterprise service providers and brokers would be able to address the software engineering challenges faced by cloud computing. It also includes clearly defined roles of cloud consumers in alleviating the technological and non-technical pressures for their seamless cloud migration.

The table 1 summarizes a country-wise adoption status and their current strategies in implementing a cloud agenda for some of the leading countries in this domain.
achieve substantial efficiency that would result in improved cloud adoption. These attribute to challenges in adapting traditional software engineering design, implementation and testing methodologies into the cloud paradigm.

Overall, the lack of functional migration of systems (Issue 2) and the lack of non-functional user-centric architecture (Issue 1) have posed technological and non-technological pressures in the cloud adoption. Figure 4 provides our proposed cloud service framework, in which best practices of SOA, Web 2.0 and Grid computing could be leveraged to address both these issues faced by SaaS, Paas and IaaS of cloud computing. This way the cloud architects (that includes cloud developers as well as policy regulators), and cloud service providers (that includes network providers as well as data centre service providers and brokers) would be able to address the software engineering challenges faced by cloud computing. It also includes clearly defined roles of cloud consumers in alleviating the technological and non-technological pressures for their seamless cloud migration.

Having identified the two major issues in the current cloud paradigm, it is also important to review the strategies and perspectives of cloud implementations from different countries’ cloud agenda as well as service providers’ perspectives of cloud implementations from different paradigm, it is also important to review the strategies and perspectives of major countries and cloud providers that would play an influential role in this context.

3.1 Countries’ Cloud Agenda

Cloud computing is being debated as the next technological disruption to transform enterprise service providers and brokers to address the software engineering challenges faced by cloud computing. It is also includes clearly defined roles of cloud consumers in alleviating the technological and non-technological pressures for their seamless cloud migration.

The table 1 summarizes a country-wise adoption status and their current strategies in implementing a cloud agenda for some of the leading countries in this domain.

FIGURE 4: PROPOSED CLOUD SERVICE FRAMEWORK.

<table>
<thead>
<tr>
<th>Country/Region</th>
<th>Adoption Status &amp; Strategy</th>
<th>Regulatory Authority / Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td>European strategy’s three broad areas: 1. Legal framework which concerns data protection and privacy, including the international dimension. It also concerns laws and other rules that have a bearing on the deployment of cloud computing in public and private organisations. And it concerns users’ rights (insofar as they are provided for by law). It aims to bring together the different legislative and regulatory areas in the EU and make a coherent policy area for Cloud Computing. 2. Technical and commercial fundamentals which aims to extend EU’s research support and focus on critical issues such as security and availability of cloud services. As a mediator, the Commission can also play a stronger role in the technical standardization of APIs and data formats, as well as in the development of template contracts and service level agreements &amp; use stimulation measures to support training and awareness, research and standardisation. 3. The market i.e. To really harness the power of public procurement; to engage with the public sector partners on Member States and regional levels to work on common approaches to cloud computing. The European Cloud Partnership will align current public sector initiatives, pulling together efforts which are fragmented and inadequate and create a substantial coherent market that could assist EU-based industry to improve on current commercial offers.</td>
<td>European Commission Information Society <a href="http://ec.europa.eu/information_society">http://ec.europa.eu/information_society</a></td>
</tr>
<tr>
<td>USA</td>
<td>All IT systems in the Federal Government should move to the cloud (25 % of all Federal IT spent moving to cloud computing). NIST is leading the development of a United States Government (USG) Cloud Computing Technology Roadmap. This roadmap will define and prioritize USG requirements for interoperability, portability, and security for cloud computing in order to support secure and effective USG adoption of Cloud Computing. Current Strategy includes: 1. Establish Cloud Service Metrics, including Standardized Units of Measurement for Cloud Resources. 2. Develop frameworks to support seamless implementation of federated community cloud environments. 3. Define and develop technical security solutions which support the decisions required by diverse sovereign, legal, business, or other authoritative policy rules. 4. Establish Cloud Service Metrics, including Standardized Units of Measurement for Cloud Resources. 5. Establish Cloud Service Metrics, including Standardized Units of Measurement for Cloud Resources.</td>
<td>National Institute of Standards and Technology, NIST <a href="http://www.nist.gov">www.nist.gov</a></td>
</tr>
<tr>
<td>Australia</td>
<td>A projected 7.1 percent of total ICT spending in Australia in 2015 will be directly Cloud related, up from 2.8 percent in 2011 Main strategies are: 1. Preparing agencies to adopt the cloud. 2. Development of a Cloud framework that includes policy, principles, contract guidance and knowledge sharing. 3. Adoption of public cloud by the Government as offerings mature. 4. Transitioning to private and government community clouds. 5. Australia to be as the “Safe, Secure, Green Cloud” destination.</td>
<td>Australian Government Information Management Policy AGIM. <a href="http://www.agimo.gov.au">www.agimo.gov.au</a> National ICT Australia, NICTA <a href="http://www.nicta.com.au">www.nicta.com.au</a></td>
</tr>
<tr>
<td>Korea</td>
<td>Reinforcing national competition policy by sharing information, development of application services based on cloud and promoting cloud services based on next-generation internet</td>
<td><a href="http://kosa.or.kr">http://kosa.or.kr</a></td>
</tr>
<tr>
<td>India</td>
<td>Yet to announce a formal cloud strategy</td>
<td>-</td>
</tr>
</tbody>
</table>
4 CURRENT OPEN SOFTWARE ENGINEERING RESEARCH CHALLENGES

Cloud computing has not only changed the way businesses could rethink about their cost-effective operations, but it has also changed the way software engineering concepts are applied in the cloud. It has caused opportunities on the hand but has resulted in some major impacts on the design, implementation, testing and maintenance phases of a typical systems development life cycle. Most researchers and industry are focussing on solving technology issues, publishing articles that discuss cloud architectures, technologies, and models. However, they have failed to consider the issues reported in this paper that are of interest to the software engineering community. Our arguments in this section would lead towards research initiatives with a focus on addressing these issues from the cloud lifecycle perspective rather than merely technology. Hence, this paper would result in a Pandora’s box of open research challenges for software engineering researchers to explore. We have listed these current open research challenges in software engineering (SE) under two main categories as listed below:

- **SE Research Challenges for Cloud Issue 1: Cloud architecture requires clear user-roles that require adaptation of SE processes and management principles as they play a pivotal role in addressing the software engineering challenges that are both technological as well as non-technological in achieving cloud adoption. The user-roles in a cloud environment are quite different from the traditional software engineering roles of systems analysts, programmers, systems evaluators, systems engineers, etc. This is because the system composition would typically consist of inter-operable components with different servers and different organisations. In other words, such a multilateral system development and deployment that is suited for cloud computing poses many software engineering challenges. Transparency in user-roles and linking them with cloud interfaces open up important research avenues in this context.

- **SE Research Challenges for Cloud Issue 2: Successful cloud migration requires suitable adaptation of the traditional SE design, implementation, testing and maintenance models. Present cloud services pose many software engineering challenges of interoperability and incompatibility of legacy file systems and applications while migrating existing legacy systems to the cloud, or migrating from one cloud model to another. Similarly, integration with existing procedures and standardisation of data control practices and policies are also posing major software engineering research challenges. Cloud services such as SaaS, PaaS and IaaS need to manage change in these technologies at the same time render the quality of service. In particular, with recent vulnerabilities found in the cloud [20], management of added security issues due to the migration to the cloud should be given prime attention. Traditional measurement metrics for fault proneness [21] should be adapted suitably for the cloud. Similar to the existence of software engineering challenges in traditional information services due to IT evolution [22], the major software engineering challenges would be in the sustainable lifecycle management of cloud technologies and at the same time providing contemporary solutions to personalised security and privacy concerns, in meeting the demands and expectations of different cloud consumers. Some leading organizations such as Microsoft and Oracle have now started to look at enterprise cloud computing strategies [23-24]. However, the strategies are yet to incorporate theories that would optimise on energy consumption for achieving green IT environment in the cloud [25-26].

5 SOFTWARE ENGINEERING RESEARCH ROADMAP

There are too many recent developments in cloud computing that have resulted in a cloudy situation. While there are new concepts of mega-exchanges and on-net one-stop service provisioning, there are technological and non-technological issues that still require attention. Overall, the abovementioned two main cloud issues have resulted in major research challenges that need to be given prime attention in order to address the differences between cloud computing theory and practice. To achieve this, we describe a research roadmap.

Cloud computing is a concept, and hence to facilitate this concept to be applied from theory into practice, it is imperative to apply a systematic, disciplined, quantifiable approach to the design, implementation, testing, operation, and maintenance of the cloud. In other words, we need to adapt software engineering principles for a sustainable adoption of the cloud. However, traditional software engineering practices are incapable of being adopted in their current form within the cloud paradigm. There are
4 CURRENT OPEN SOFTWARE ENGINEERING RESEARCH CHALLENGES

Cloud computing has not only changed the way businesses could rethink about their cost-effective operations, but it has also changed the way software engineering concepts are applied in the cloud. It has caused opportunities but on the hand has resulted in some major impacts on the design, implementation, testing and maintenance phases of a typical systems development life cycle. Most researchers and industry are focussing on solving technology issues, publishing articles that discuss cloud architectures, technologies, and models. However, they have failed to consider the issues reported in this paper that are of interest to the software engineering community. Our arguments in this section would lead towards research initiatives with a focus on addressing these issues from the cloud life cycle perspective rather than mere technology perspective. Hence, this paper would result in a Pandora's box of open research challenges for software engineering researchers to explore. We have listed these current open research challenges in software engineering (SE) under two main categories as listed below:

• SE Research Challenges for Cloud Issue 1: Cloud architecture requires clear user-roles that require adaptation of SE processes and management principles as they play a pivotal role in addressing the software engineering challenges that are both technological as well as non-technological in achieving cloud adoption. The user-roles in a cloud environment are quite different from the traditional software engineering roles of systems analyst, programmer, systems evaluator, systems client, consumer, etc. This is because the system composition would typically consist of inter-operable third party components with no source codes for some of the components and could be distributed among different servers and different organisations. In other words, such a multilateral system development and deployment that is suited for cloud computing poses many software engineering challenges. Transparency in user-roles and linking them with cloud interfaces open up important research avenues in this context.

• SE Research Challenges for Cloud Issue 2: Successful cloud migration requires suitable adaptation of the traditional SE design, implementation, testing and maintenance models. Present cloud services pose many software engineering challenges of interoperability and incompatibility of legacy file systems and applications while migrating existing legacy systems to the cloud, or migrating from one cloud model to another. Similarly, integration with existing procedures and standardisation of data control practices and policies are also posing major software engineering research challenges. Cloud services, such as SaaS, PaaS and IaaS need to manage change in these technologies at the same time render the quality of service. In particular, with recent vulnerabilities found in the cloud [20], management of added security issues due to the migration to the cloud should be given prime attention. Traditional measurement metrics for fault proneness [21] should be adapted suitably for the cloud. Similar to the existence of software engineering challenges in traditional information services due to IT evolution [22], the major software engineering challenges would be in the sustainable life cycle management of cloud technologies and at the same time providing contemporary solutions to personalised security and privacy concerns, in meeting the demands and expectations of different cloud consumers. Some leading organisations such as Microsoft and Oracle have now started to look at enterprise cloud computing strategies [23-24]. However, the strategies are yet to incorporate theories that would optimise on energy consumption for achieving green IT environment in the cloud [25-26].

5 SOFTWARE ENGINEERING RESEARCH ROADMAP

There are too many recent developments in cloud computing that have resulted in a cloudy situation. While there are new concepts of mega-exchanges and on-net one-stop service provisioning, there are existing technological and non-technological issues that still require attention. Overall, the abovementioned two main cloud issues have resulted in major research challenges that need to be given prime attention in order to address the current differences between cloud computing theory and practice. To achieve this, we describe a research roadmap:

Cloud computing is a concept, and hence to facilitate this concept to be applied from theory into practice, it is imperative to apply a systematic, disciplined, quantifiable approach to the design, implementation, testing, operation, and maintenance of the cloud. In other words, we need to adapt software engineering principles for a sustainable adoption of the cloud. However, traditional software engineering practices are incapable of being adopted in their current form within the cloud paradigm. There are
current open research challenges and we have enumerated some of the prominent software engineering challenges associated with each identified cloud issue.

Cloud computing has opened up new opportunities for software engineering researchers to adopt innovative modifications to software engineering theories, such as multilateral software development within SOA, and adaptations to practices such as Web 2.0. Theories should facilitate in the intelligent automation of the migration process into the cloud, similar to some attempts made by Cisco [27]. In addition, test beds in the cloud [28-32] that would facilitate the evaluation of a possible cloud migration seamlessly would alleviate the fear associated with the cloud adoption. Various testing in the cloud, on the cloud and over the cloud could therefore be explored with innovative and intelligent cloud testing strategies [30].

We propose a SE research roadmap that comprises of four main steps, as described below:

Step 1: Create new Software Engineering theories, approaches, and techniques for Cloud computing

Good practices from SOA, Web 2.0 and Grid computing that could be adapted to the cloud should be created.

Step 2: Identify techno-social cloud user-roles that clearly define the cloud user responsibilities in dealing with both technological and non-technological issues of the cloud. Such a framework would integrate the cloud user-roles and user interfaces facilitating in transparency and a better understanding of each other's responsibilities in the cloud value chain.

Step 3: Formulate self-organised service-oriented framework that enables automatic and seamless next generation scaling and optimisation of the effectiveness and efficiency of cloud computing environments. Such a self-organisation framework would aid in seamless migration of existing systems into cloud. It defines service level objectives and measure the delivery of those objectives in maintaining quality of service (QoS) in the cloud.

Step 4: Establish trusted cloud computing governance that mitigates any security, identity, privacy and manageability risks in cloud-based environments.

This would require new advanced technologies and solutions such as, innovative encryption algorithms, data anonymisation, energy saving procedures, efficient cost-bounding and resource utilisation protocols, flexible and user-sensitive data governance, federated services, dynamically adaptive and transparent costing models, as well as integrating user-roles with cloud interfaces.

Clearly, software engineering research directions for the future cloud are towards multilateral software development, trusted cloud infrastructures, automatic migration and scalability models, and integrating cloud user-roles with service-oriented architectures to enhance QoS in the cloud environments. At present, many types of cloud services provider are flooding the market place [31-32], and setting up different battlefields and value chains in the cloud arena [33]. Our next step would be to evaluate the state-of-the-art exhibited by various user-roles such as Cloud Architects, Cloud service Providers and Cloud Consumers with reference to our proposed software engineering research roadmap for cloud computing. We would also evaluate the effectiveness of our proposed roadmap in terms of achieving interoperability, flexibility and scalability, which are the fundamental principles of cloud computing [34-36] that are warranted for a successful user adoption of the cloud.

6 CONCLUSIONS

In this paper, we have identified that software engineering principles in the current form have attributed to the present cloud computing issues that arise from both technological and non-technological pressures. Non-functional user requirement issues such as privacy, and system's functional migration bottlenecks due to interoperability and other migration issues play a major role in cloud adoption. We have argued that present cloud computing issues require migration issues play a major role in cloud adoption. We require migration and scalability models, and integrating cloud user-roles and functional architectures for a sustained cloud adoption. Cloud user-roles and traditional software engineering principles and models require re-thinking of the system architecture, design, implementation, testing and maintenance in engineering the cloud services, in particular the SaaS.

We have recommended four main research steps that should alleviate the current situation. In future, we would also evaluate the research and industry developments as against the state-of-the-art accomplishments of these four research steps in the software engineering research roadmap and their impact on cloud adoption. This would lead towards refining the roadmap further. We believe that research roadmap presented in this paper has provided only an initial framework that would instill more research towards evolving into an established roadmap in future years to come.

REFERENCES

current open research challenges and we have enumerated some of the prominent software engineering challenges associated with each identified cloud issue.

Cloud computing has opened up new opportunities for software engineering researchers to adopt innovative methodologies to software engineering theories, such as multilayer software development within SOA, and adaptations to practices such as Web 2.0. Theories should facilitate in the intelligent automation of the migration process into the cloud, similar to some attempts made by Cisco [27]. In addition, test beds in the cloud [28-29] that would facilitate the evaluation of a possible cloud migration seamlessly would alleviate the fear associated with the cloud adoption. Various testing in the cloud, on the cloud and over the cloud could therefore be explored with innovative and intelligent cloud testing strategies [30].

We propose a SE research roadmap that comprises of four main steps, as described below:

Step 1: Create new Software Engineering theories, approaches, and techniques for Cloud computing. Good practices from SOA, Web 2.0 and Grid computing that could be adapted to the cloud should be considered.

Step 2: Identify techno-social cloud user-roles that clearly defines the cloud user responsibilities in dealing with both technological and non-technological issues of the cloud. Such a framework will articulate the cloud user-roles and user interfaces facilitating in transparency and a better understanding of each user’s responsibilities in the cloud value chain.

Step 3: Formulate self-organised service-oriented framework that enables automatic and seamless next generation scaling and optimisation of the effectiveness and efficiency of cloud computing environments. Such a self-organisation framework would aid in seamless migration of existing systems into cloud. It defines service level objectives and measure the delivery of those objectives. It maintains quality of service (QoS) in the cloud.

Step 4: Establish trusted cloud computing governance that mitigates any security, identity, privacy and manageability risks in cloud-based environments. This would require new advanced technologies and solutions such as, innovative encryption algorithms, data anonymisation, energy saving procedures, efficient load-balancing and resource utilisation protocols, flexible and user-sensitive data governance, federated services, dynamically adaptive and transparent costing models, as well as integrating user-roles with cloud interfaces.

Clearly, software engineering research directions for the future cloud are towards multilayer software development, trusted cloud infrastructures, automating migration and scalability models, and integrating cloud user-roles with service-oriented architectures to enhance QoS in the cloud environments. At present, many types of cloud service providers are flooding the market place [31-32], and setting up different battlefields and value chains in the cloud arena [33]. Our next step would be to evaluate the state-of-the-art exhibited by various user-roles such as Cloud Architects, Cloud Service Providers and Cloud Consumers with reference to our proposed software engineering research roadmap for cloud computing. We would also evaluate the effectiveness of our proposed roadmap in terms of achieving interoperability, flexibility and scalability, which are the fundamental principles of cloud computing [34-36] that are warranted for a successful user adoption of the cloud.

6 CONCLUSIONS

In this paper, we have identified that software engineering principles in the current form have attributed to the present cloud computing issues that arise from both technological and non-technological pressures. Non-functional user requirement issues as privacy, and system’s functional migration bottlenecks due to interoperability and other migration issues play a major role in cloud adoption. We have argued that present cloud computing issues require prime attention from software engineering theory and architecture perspectives for a sustained cloud adoption. Cloud user-roles and traditional software engineering principles and models require re-thinking of the system architecture, design, implementation, testing and maintenance in engineering the cloud services, in particular the Saas.

We have recommended four main research steps to alleviate the current situation. In future, we will also evaluate the research and industry developments as against the state-of-the-art accomplishments of these four research steps in the software engineering research roadmap and their impact on cloud adoption. This would lead towards refining the roadmap further. We believe that research roadmap presented in this paper has provided only an initial framework that would instill more research towards evolving into an established roadmap in future years to come.
Software Maintainability Prediction using Machine Learning Algorithms

Ruchika Malhotra¹ and Anuradha Chug²

¹Department of Software Engineering, Delhi Technological University, Delhi 110042, India
²University School of Information and Communication Technology, GGSIP University, Dwarka, New Delhi 110077

ruchikamalhotra2004@yahoo.com, a_chug@yahoo.co.in

Abstract - Software maintainability is one of the most important aspects while evaluating quality of the software product. It is defined as the ease with which a software system or component can be modified to correct faults, improve performance or other attributes or adapt to a changed environment. Tracking the maintenance behaviour of the software product is very complex. This is precisely the reason that predicting the cost and risk associated with maintenance after delivery is extremely difficult which is widely acknowledged by the researchers and practitioners. In an attempt to address this issue quantitatively, the main purpose of this paper is to propose use of few machine learning algorithms with an objective to predict software maintainability and evaluate them. The proposed models are Group Method of Data Handling (GMDH), Genetic Algorithms (GA) and Probabilistic Neural Network (PNN) with Gaussian activation function. The prediction model is constructed using the above said machine learning techniques. In order to study and evaluate its performance, two commercial datasets UIMS (User Interface Management System) and QUES (Quality Evaluation System) are used. The code for these two systems was written in Classical Ada. The UIMS contains 39 classes and QUES datasets contains 71 classes. To measure the maintainability, number of "CHANGE" is observed over a period of three years. We can define CHANGE as the number of lines of code which were added, deleted or modified during a three year maintenance period. After conducting empirical study, performance of these three proposed machine learning algorithms was compared with prevailing models such as GRNN (General Regression Neural Network) Model, ANN (Artificial Neural Network) Model, Bayesian Model, RT (Regression Tree) Model, Backward Elimination Model, Stepwise Selection Model, MARS (Multiple Adaptive Regression Splines) Model, TreeNets Model, GN (Generalized Regression) Model, ANFIS (Adaptive Neuro Fuzzy Inference System) Model, SVM (Support Vector Machine) Model and MLR (Multiple Linear Regressions) Model which were taken from the literature. Based on experiments conducted, it was found that GMDH can be applied as a sound alternative to the existing techniques used for software maintainability prediction since it assists in predicting the maintainability more accurately and precisely than prevailing models.

Keywords: GMDH (Group Method of Data Handling), Genetic Algorithms, Probabilistic Neural Network (PNN), Software Maintainability, Software Maintainability Prediction Metrics and Modeling.

I. INTRODUCTION

Software maintainability means the ease with which a software system or component can be modified to correct faults, improve performance or other attributes or adapt to a changed environment [1]. The change in the software is required to meet the changing requirements of customers which may arise due to many reasons such as change in the technology, introduction of new hardware or enhancement of the features provided etc. Producing software which does not need to be changed is not only impractical but also very uneconomical. This process of changing the software which has been delivered is called software maintenance. The amount of resource, effort and time spent on software maintenance is much more than what is being spent on its development. Thus, producing software that is easy to maintain may potentially save large costs and efforts. One of the main approaches in controlling maintenance cost is to monitor software metrics during the development phase. It is a matter of interest for researchers to measure various attributes of software design in terms of inheritance, coupling, cohesion etc and predict its maintenance behaviour on the basis of their values. The problem of predicting the maintainability of software is widely acknowledged in the industry and much has been written on how maintainability can be predicted by using various tools and processes at the time of designing with the help of software design metrics [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Studies have been conducted and found the strong link between Object Oriented software metrics and its maintainability. They have also found that these metrics can be used as predictors of maintenance effort. Accurate prediction of software maintainability can be useful because of the following reasons:

(a). It helps project managers in comparing the productivity and costs among different projects.
(b). It provides managers with information for more effectively planning the use of valuable resources.