Cloud-Based Support for Global Software Engineering: Potentials, Risks, and Gaps

Mohammed Al-qadhi
Department of Computer Science
City University of Hong Kong
Hong Kong SAR, China
+852 3442 2306
moalqadhi2-c@my.cityu.edu.hk

Jacky Keung
Department of Computer Science
City University of Hong Kong
Hong Kong SAR, China
+852 3442 2591
jacky.keung@cityu.edu.hk

ABSTRACT
Global Software Engineering (GSE) aims at utilizing global expertise and reducing time-to-market and costs for software development. However, GSE faces serious challenges related to the communication, coordination and provision of development environments for geographically distributed GSE sites, resulting in increased costs. Nowadays, cloud computing offers a cost-effective and a handy solution to provide computing support for global business. In particularly, it provides potential and viable solutions to address concerns and challenges in GSE. We have reviewed the relevant literature on cloud-based support for GSE and identified a set of pros and cons for supporting GSE with cloud services. We also present some recommendations to increase the trustworthiness and productivity with cloud-based support for GSE. This paper outlines the potential benefits and risks of cloud computing to GSE and identifies some gaps that need to be addressed by cloud-based support towards more trustworthy and productive work environment for GSE. This will open windows of opportunities for tailoring the solution utilizing cloud-based support for GSE.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management—Programming teams; K.6.4 [**Management of Computing and Information Systems**]: System Management—Centralization/decentralization.

General Terms
Management, Standardization.

Keywords
Cloud Computing, Global Software Engineering (GSE).

1. INTRODUCTION
The notion of globalism is penetrating almost all today's world facets. Software Engineering is not an exception where the trend of Global Software Engineering (GSE) became more wide spreading. GSE means that software is developed across borders by teams scattered across different geographical areas (see Figure 1). Adoption of GSE model contributes effectively to software business by capitalizing on global resource pools, reducing development costs and time-to-market through round-the-clock development [5, 17].

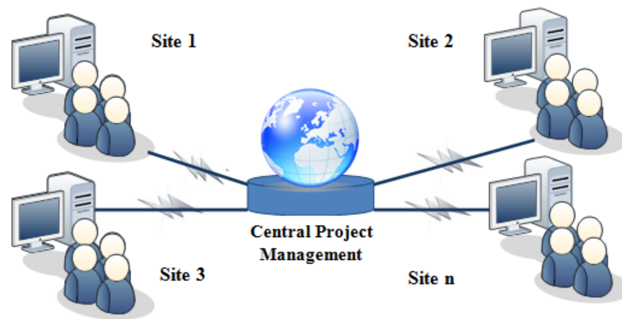

Figure 1. An overview of GSE process.

However, GSE faces some challenges including: geographic, cultural, linguistic, and temporal (i.e. difference in term of time zones) dimension forming what is called *global distance* [6-8, 15]. Another big challenge of GSE is associated with the effort needed for building on-premises IT infrastructure needed to run a project whose development team is dispersed across multiple geographical areas which is costly. These costs include the effort needed for installation, licensing, monitoring the performance and maintaining all software and hardware within all development sites.

Advent of cloud computing model has provided the enterprises with an opportunity to reduce the costs of establishing IT infrastructure by offering computing resources as utilities and converting computing expenses from capital expenses into operational ones [25]. Also, cloud computing can provide an extensive support to the spectrum of project engineering and management tasks of GSE [16]. Research has shown that cloud computing can make GSE activities cost-effective and easier despite of the difficulties posed by global distance between geographically distributed teams.

The main objective of this paper is to highlight the main potential benefits and inevitable risks that GSE will have as a result of supporting it with cloud services. Benefits are enhancements of the following aspects: productivity, testing, overall GSE process, IT operations and content management. In addition, it helps reducing negative effect of global distance and encouraging standards adoption. On the other hand, risks include: existence of task dependencies, unavailability of network connectivity, effect of cloud-based platform on code commitment and integration,

InnoSWDev'14, November 16, 2014, Hong Kong, China
Copyright 2014 ACM 978-1-4503-3226-2/14/11...$15.00
http://dx.doi.org/10.1145/2666581.2666583

lack of policy to propagate changes for a code to its linked codes within the codebase, variety of cloud configurations for different cloud service providers, security of stored test data, and varying governmental regulations across different countries.

Furthermore, we presented two recommendations to enhance trustworthiness and productivity for cloud-supported GSE. They are to make rigorous collaboration rules and enhancing software reuse by supporting it with cloud services.

Organization of the paper is as follows. Section 2 provides a brief background of cloud computing and Section 3 provides an overview of the potentials that can be achieved by making GSE supported with cloud computing, as well as, inevitable risks associated with such support. In section 4, based on the literature we discuss we discuss some recommendations to be considered in future cloud-based GSE research. In section 5, we present some future work directions. The last section concludes and summarizes the paper.

2. CLOUD COMPUTING

2.1 Key Motivation

Cloud computing provides enterprises with access to tremendous pools of internet-based IT infrastructure including both hardware and software. This will relieve enterprise from the costs of building on-premises IT infrastructure. Enterprises will gain the following basic advantages when adopting cloud computing paradigm [13]:

(a) Capacity planning – it is always necessary to estimate the fluctuated usage load for computing infrastructure. It requires balancing peak usage requirements without causing unnecessary over-expenditure on infrastructure. There are three capacity planning strategies applied by cloud computing [13, 27]:

- Lead strategy – provision of more resources proactively by adding or subtracting capacity in anticipation of future market conditions and demand.
- Lag strategy – provision of more resources reactively when currently assigned resources are fully utilized by adjusting its capacity in response to demand. In favorable market conditions, the lag strategy generally does not add capacity until the firm is operating at full capacity. The lag capacity strategy is a conservative approach that may result in lost opportunity when demand increases rapidly.
- Match strategy – a moderate strategy for provision of small resources increments in proportion to demand and changing market conditions.

(b) Cost reduction – provisioning computing infrastructure includes two main costs: the cost of acquiring new infrastructure, and the cost of its ongoing ownership. Operational overhead represents a considerable share of IT budgets, often exceeding up-front investment costs. Cloud computing provides a direct alignment between IT costs and business performance through on-demand-request of the computing facilities rather than purchasing and deploying them on-premises.

(c) Organizational agility – businesses should be able to adapt and evolve to successfully face change caused by both internal and external factors. Organizational agility is the measure of an organization's responsiveness to changes related to market demands, competitive pressures, and its own strategic business goals. Cloud computing making it easier for enterprise to successfully and seamlessly adapt to unexpected changes in business needs that require more or less computational resources and, consequently, more or less IT resources provision.

2.2 Enabling Technologies

Contemporary clouds are underpinned by a set of primary matured technologies which preexisted before the advent of cloud computing as shown in table 1 below [13]:

Table 1. Enabling technologies for cloud computing

Enabling Technology	Description
Contemporary Broadband Networking and Internet Technologies	Accessing cloud prerequires internet connectivity. Internet allows anytime anywhere provisioning of IT resources. Despite most clouds are internet-enabled, cloud service customer can still access the cloud using only private and dedicated network links in LANs.
High Performance Data Centers	Helping bringing together IT resources instead of being geographically dispersed. This allows power sharing, higher efficiency in shared IT resource usage and improved accessibility for IT personnel.
Modern Virtualization Technologies	Converting a physical IT resource (e.g. servers, storage and network) into a virtual resource (e.g. virtual server, virtual disk and VLAN) that runs under management of virtualization software (aka hypervisor). Virtualization supports hardware independence of the underlying platform and architecture, as well as, replication of the same IT resource even on the same physical server.
Web Technologies	Cloud computing uses web technology as being both the implementation medium and the management interface. Web technologies include: URL, HTTP, HTML, XML and web applications.
Multitenancy Technology	Providing multiple instances of a system to different clients (aka tenants). In fact, virtualization is a cheap and easy alternative to multitenancy.
Service-Oriented Architecture (SOA) Technology	A software engineering technique that enables software vendors to offer their software products as services to clients who want to save their money, effort, and time by adopting pay-as-you-go computing model.

2.3 Basic Concepts and Issues

Cloud computing model is characterized by a set of concepts that include but, not limited to: resource scaling, Service-Level Agreement (SLA), deployment models, delivery models. On the other hand, there is a set of issues that needs to be addressed by future cloud computing research. These concepts and issues are shown in table 2 below.

Table 2. Some Concepts and issues of cloud computing

Concept/Issue	Description
Resource scaling	Cloud service provides a rapid adaptation according to business requirements by means of two types of resource scaling: (1) Horizontal scaling – scaling out (i.e. allocating) or scaling in (i.e. releasing) resources. (2) Vertical scaling – scaling up by replacing an IT resource with another of higher capacity while, scaling down is done by replacing an IT resource with another of a lower capacity.
Service-Level Agreement (SLA)	SLA is a critical document specifying: QoS features, behaviors, limitations of cloud-based service or other provisions. It also provides a detailed specification of measurable characteristics related to IT service outcomes such as: uptime, security, and some QoS features.
Cloud security	Cloud security became a critical issue due to the existence of multiple clients on the same computing environment there is a serious issue of "Overlapping Trust Boundaries" threatening the security of cloud-based services.
Unregulated marketplace	There is a need for establishing well-defined standard regulations and laws to help establishing the trust between cloud service providers and customers.
Cloud delivery models	**Infrastructure-as-a-Service (IaaS) model** It provides computing infrastructure, physical or (quite often) virtual machines and other resources such as virtual machine disk image library, block and file-based storage…etc. Examples include Amazon EC2, Windows Azure …etc. **Platform-as-a-Service (PaaS) model** It provides computing platforms which typically include operating system, programming language execution environment, database, web server…etc. Examples include AWS Elastic Beanstalk, Google App Engine …etc. **Software-as-a-Service (SaaS) model** It provides access to an internet-based application. Examples include Google Apps and Microsoft Office 365.
Cloud deployment models	**Public cloud model** Here, services, such as applications and storage, are available for general use over the Internet. An example of a public cloud is IBM's Blue Cloud. **Private cloud model** It is a virtualized data center that operates within a firewall. Private clouds are highly virtualized, joined together by mass quantities of IT infrastructure into resource pools, and privately owned and managed. **Hybrid cloud model** A hybrid cloud is a mix of public and private clouds. **Community cloud model** It is an infrastructure shared by several organizations which supports a specific community.

3. CLOUD-BASED SUPPORT FOR GSE

Cloud computing has shown a promising level of support for Global Software Engineering (GSE). This support is exemplified in the provision of all development-related facilities in the form of "pay-as-you-go" utilities. These services include: computing infrastructures, platforms, development environments and tools.

In the following underlying subsections, we are going to distill the potentials and risks of applying cloud-based support to GSE process as demonstrated by related research work.

3.1 Potentials

Based on the literature, we can categorize the research on the potential benefits of applying cloud-based support to GSE based on different development aspects into the following:

3.1.1 Productivity

Cloud computing support for development can bring high availability of development environments that supports quick software prototyping. An experiment by [28] used a formed computing cloud for compiling and building up Eclipse IDE from its Java source code. The result was promising as it took about 80 minutes with 5 standardized desktop workstations while, it will take 150 minutes if it performed as a standalone job.

A model for GSE with cloud platforms in [12] claims that cloud computing support can save the time and effort for distributed agile development if some introduced factors holds true. The simulation of the model has shown that development rate with cloud-based support for the agile GSE process was higher than the development rate of agile GSE without cloud-based support.

In a qualitative case study [23], it is found that cloud-based support for GSE resulted in allowing development teams to contribute to the development immediately after gaining access to shared cloud-based development tools resulting in rapid development. It also helps in faster ramp-up by speeding up the overall development process.

As an industrial case, Fujitsu company reported enhanced productivity as one of the advantages of supporting its middleware development with cloud services [4].

3.1.2 Testing

Potential improvement for testing and quality assurance can be made through [28]:

- Performing cloud-based infrastructure software testing across platforms with various combinations.
- Continuous code integration by providing "compile-test-change" software cycle which will support continuous builds

and integration to meet strict guidelines of quality software development.

- Generating fully functional test cases with industry standard frameworks by performing tests under cloudified virtual machines. This will enhance testing with the following advantages: flexible and quick tests, standardized test procedures, and simulating the real-world operation of the software under a large volume of users.

In agile GSE, cloud-based support can facilitate testing by rapid and easily accessible prototypes and demos to reduce the time needed to get customer feedback [12]. This will save the time and effort for distributed agile development.

A study of 15 software organizations to get their viewpoints regarding cloud-based testing found that cloud-based testing helps attaining the following advantages: efficient performance testing, faster testing, more realistic test results, availability of testing tools, better developer-tester communication, and helping vendors delivering quality service through timed addressing of customers' needs [24].

In [19], a cloud-based testing platform called AGARIC has exhibited high efficiency and effectiveness of load testing for online web applications with diversely distributed nodes.

Cloud-based testing experiments on "D-cloud" testing environment allowed setting up the testing environment easily, as well as, ability to test software to be deployed on distributed systems [14].

Cloud-based unit testing using Testing-as-a-Service (TaaS) has shown scalability and helped automating the entire unit testing process which saved the tester's effort [21].

3.1.3 Overall GSE Process

Hashmi et al. [16] have elaborately shown that utilizing cloud computing will facilitate the entire process of GSE against many of its challenges. This support manifests as both a process and a product as shown in table 3 below.

Table 3. Potential areas of contribution by cloud-based as a process and a product to support GSE by Hashmi et al. [16]

GSE Challenge	**Potential Contribution of Cloud Support**	
	As a product	**As a process**
Coordination and Collaboration	Eliminates temporal and geographical distance through SOA.	Resource sharing.
Support of Technical Development	SaaS helps removing tasks and architecture dependencies. Also, it makes cost reduction through reuse of services.	-
Geographical Distance	PaaS provides a pool of shared hardware and software resources that comply with the needs of geographically distributed teams.	Turning GSE activities into services can overcome software evolution, reuse, and deployment.
GSE Process Optimization	Adaptation to business requirements changes of the GSE process.	Eliminates temporal and geographical distance through cloud-based collaboration.
Service Ownership	Resolves ownership issue of GSE through well-defined ownership for each cloud service.	Resolves ownership issue of GSE through SOA which collocates services by service development.
Teams Communication	Reduces the need for cross-site communication through making appropriate service definitions act as proxies.	-
Project Knowledge Transfer	Eliminates the need for sharing the knowledge as the services contributing to GSE process are self-described, as well as, the knowledge is developed locally.	Eliminates the need for sharing the knowledge because making tasks as services helps transferring tasks information under service model.
Execution Monitoring	Ensuring correct services execution order by central services schedulers.	Ensuring correct services execution order through sharing their execution context among services. This will help in collaboration and eliminating geographic distance.
Project Configuration Management	Facilitating project configuration management using service registries.	-
Technical Issues	Minimizing inter-task dependencies as services are loosely-coupled to help seamless recovery/replacement of failed services at runtime.	Assuring high quality software by performing cloud-based testing services and standardized tests.

3.1.4 IT Operations

The two concepts of: (1) Platform-as-a-Service model to support and facilitate global software development and (2) delivery practices and standards for internet-based Software-as-a-Service solutions are compatible. Designing a platform that theoretically supports all service delivery management processes – even with the current technologies – will have a great positive impact on global software development making developers only concentrating on the quality of their products and getting them rid of problems of platform delivery options [26].

Cloud-based support to GSE will reduce GSE costs through: making coordination and collaboration through pay-per-use model instead of on-premises installation of collaboration tools, provisioning of work information exchange on software development activities as a process, and flexible and instant collaboration through customizable scalability [16].

IT operations have two areas that can be improved through cloud-based support. First, it helps reducing deployment costs by relieving the organization from the burden of building totally new on-premises infrastructures and instead adopting pay-per-use computing model. Second, it makes easy and effective use of backup and restore to support powerful software configuration management and business continuity, archiving…etc. [28].

Also, time and effort can be saved through using GSE with cloud platforms as at the beginning of the development, there will be no need for installation of software tools, hardware, and software licenses. Moreover, during the development, there will be no need for installation and upgrading for development tools [12, 23].

Fujitsu company has reported that cloud-based middleware development resulted in: reduction of server operation costs, balanced server use, reduction of working hours to set up construct development environment, better resource access management, and improved information management [4].

There are seven advantages of providing Tool-as-a-Service (TaaS) in GSE that include: no fixed license fees and avoiding replication of software and hardware across sites, alignment of tools with processes and organization size, artifacts traceability, working with sensitive and massive data, working with expensive and technology sensitive hardware, support for awareness and collaboration and inter-organization knowledge ecosystem [11].

3.1.5 Global Distance

Geographically distributed GSE teams face the issue of global distance as a result of geographic, cultural, linguistic, and temporal distances [6-8, 15].

Cloud services can provide support of technical development by eliminating temporal and geographical distance through supporting development process with services. It facilitates team management and coordination through SOA by keeping track of information without changing the underlying infrastructure [16].

Provisioning development tools as cloud-based service will eliminate the need for local IDEs which will help resolving geographic and temporal distance (i.e. difference in time zones) challenges [28].

Moreover, cloud-based support helps resolving the problem of miscommunication between clients and venders within GSE process resulting from the linguistic gap between clients and venders especially, during application requirements definition where language difference may cause severe errors in the development of the final system. For example, to resolve the miscommunication problem, an agent-based requirement definition support system for a client and offshore vender where cloud-based services can be used to bridge the linguistic gap [20].

3.1.6 Content Management

Cloud computing support helps offering content collaboration spaces for managing the project-related information like: Software Requirements Documents (SRDs), Request for Proposal (RFP), datasets, images, invoices…etc. It will help collecting this information from different resources scattered across various development sites. Also, it provides a scalable data management platform to support various activities of GSE environments. This helped providing a location-independent content collaboration space for storing code repositories, digital content, file archives, streaming media [4, 28].

3.1.7 Standardization

Cloud computing offers an opportunity for enforcing development standards adoption. For example, Service-Oriented Architecture (SOA), which is one of the main enabling technologies for cloud computing, is posing its underlying standard structure. Therefore, developers should adopt SOA interfacing standards if the software under development is to be deployed on the cloud.

Experiments by [28] have shown that using of cloud services implies provisioning of Application Programming Interfaces (APIs) frameworks that enforce the adoption of standard programming model APIs, styles, and conventions. Also, it will force developers to put into account constraints for multicore, parallel programming, and virtualization. Moreover, cloud-based continuous code integration will support continuous builds and integration to meet strict guidelines of quality software development.

In testing, cloud-based support will help adopting industry standard testing frameworks when performing tests under cloudified virtual machines which leads to more standardized test procedures [28].

3.2 Risks

There are five main risks of a cloud-based GSE which include [23]:

(1) Dependencies – can be either technical or operational like waiting a team to deliver a task. Sometimes, it is hard to determine interconnected components due to lack of communication between concerned teams.

(2) Unavailability of network connectivity.

(3) Code commitment and integration – committing code to the proper code repository needs staff well-aware of the underlying cloud-based platform. Moreover, integrating a code to the overall product requires extra testing on the cloud-based platform.

(4) Technical debt – consequent commitment of changes to a code in the codebase may not get visible to other code linked to code whose changes been committed.

(5) Additional managerial and operational support costs.

Moreover, challenges of cloud-based testing include: different cloud service providers are having different service configurations, security of the stored test data, varying governmental regulations across geographical regions imposed a better test data management [24].

4. DISCUSSION

After distilling the potentials and risks of cloud-supported GSE, we here deduce and present some gaps that need to be addressed in order to encourage GSE industry migration to the cloud.

In section 3, we could see that almost all GSE engineering and managerial activities can gain betterment by making them cloud-based. However, there are some factors that may delay GSE industry from migrating to the cloud. Some of these factors are generic and applies to any enterprise such as: (1) lack of regulations for governing cloud computing marketplace (2) security-related threats. On the other hand, cloud-based GSE still faces some complexities associated with its processes [11].

Cloud-based support for GSE still needs more comprehensive research work that considers all aspects that affects directly or indirectly GSE process. Chauhan et al. [11] have indicated ten quality attributes that Tool-as-a-Service (TaaS) for GSE, as well as, hosting cloud infrastructure should support. The list includes: supporting multiple organization, maintaining different versions of the tools, combining multiple tools, compatibility with commercial tools, enabling tools to work with private data, support for multiple types of persistence methods accessible through tools, access by multiple types of devices, service level agreement compliance, specifying and modeling service level agreement (SLA) and tool bundling.

Yu et al. [22] proposed a detailed task management system for cloud-based testing. They have targeted the analysis of the dependencies and conflicts relationships between test tasks and considering these relationships throughout the entire testing process. They found that test tasks management system helps: (1) managing concurrent testing conflicts by adjusting execution order of tests (2) accommodating resource assignment dynamically (3) saving total testing time by recovering from testing errors resulting from dependencies between test tasks.

In the underlying subsections, we present two main gaps that need to be bridged and addressed by future research work related to cloud-based support for GSE.

4.1 Collaboration

4.1.1 Collaboration Challenges

GSE teams are in continuous need for collaboration and exchange of information and development artifacts. This need comes from the existence of unavoidable dependencies between the delegated tasks among the distributed GSE teams. Cataldo [10] has shown that dependencies within GSE process has a socio-technical duality where there are two dimensions: technical and socio-organizational for the development process.

Project team members need some assurance that cloud-based GSE will exhibit the same level of trustworthiness in the project-related information and artifacts as in the traditional software engineering despite of the dispersive nature of GSE. However, GSE process has some complexities that stem from its geographic, cultural, linguistic, temporal, and task dependencies challenges. Such challenges cause project artifacts shared or exchanged by distributed teams throughout development process to be subject to uncertainty. This uncertainty is due to the lack or incompleteness of information associated with the shared or exchanged artifacts.

The literature contains some radical solutions to resolve or at least reduce task dependencies. For instance, Jacobson et al. [18] proposed a use-case-driven development approach that employs aspect-orientation. They have introduced the concept of use-case modules which is a design unit to keep the specifics of a use-case from all models and artifacts in the entire software project. Hence, one possible solution for task dependencies in GSE is to assign each development site a use-case module which will have no dependencies on other use-case modules.

Also, the uncertainty associated with the project-related information and artifacts can be eliminated by introducing rigorous and standard rules to manage the collaboration and sharing across the cloud services which support GSE process.

4.1.2 Establishing Rigor for Collaboration

GSE needs unified standards to facilitate collaboration and sharing across different development sites. The unified standards should address all possible aspects that may affect the collaboration which mainly include:

- **Standard inter-site communication formats**. For example, a developer in a site may need to exchange some bug information with a colleague in another site. Usually, inter-site communication tends to be formal where the used words and sentences are technical terms and statements. Hence, the existence of a standard inter-site communication makes the communication between sites more efficient and effective. Regarding possible need for translation during inter-site communication due to lingual differences between communicators, a cloud-based language translation services such as the one proposed by [20] can be utilized with adding the ability to learn the new words and semantics.
- **Standard code integration policy.** It is a very critical issue as all the distributed sites will be sharing the work of one or more projects. So, a project team will be working on different pieces of a project in terms of system functions or software components (i.e. modules, subsystems...etc.). A team member may need to use some components that are developed by another development site whose code is already stored in the shared codebase. Hence, integration and testing require a well-defined policy that governs synchronizing and committing changes within the shared codebase to be included during integration and testing stages at a certain time. A code integration policy needs to have the following:

(a) Providing a powerful code version management which allows a participant at any development site to choose the version of software component to be involved in the integration or testing. Cloud services can aid in providing massive storage pool to keep history of changes made to a software component and therefore, keeping a various versions of the component within a certain time period. Hence, each version of software component should be associated with a stamp that shows date, time, a unique version ID, and the participant who have made the change and brief description about the change(s).

(b) Making change and version management to be based on a well-defined privileges scheme where different cloud-based GSE service users are having different authorization levels in terms of access, change, committing change and usage.

4.2 Cloud-Based Support for Software Reuse

There is a myriad of software industry companies and individual developers around the world. Many of these companies and individual developers are making software development business in alignment with the state-of-the-art software engineering research in terms of both methodologies and tools. Their work may reuse preexisting software components that are either open source or proprietary. Otherwise, they have to develop the software entirely from scratch.

All software engineering paradigms appeared in the literature so far aim at giving the maximal support for software reuse. This list includes but, not limited to: object-oriented, aspect-oriented and component-based software engineering paradigms. There are a lot of crafted software products that can be reused but, they gained less fame. This is due to that software reuse is still unorganized. In other words, there are no well-established venues that can provide organized hosting and environment for reusing such software products. Even initiatives like GitHub [1], Google code [2] and SourceForge [3] offer just web-based software hosting where the program source code, packages and executables are stored.

We here suggest a cloud-supported software reuse as a service to provide more than just a hosting and storage space for reusable software components. It should provide both options of reuse: (1) offline component-based reuse (2) online SaaS-based reuse model. This service is expected to facilitate and enable an effective, efficient and smart software reuse through:

- Massive hosting space for reusable software components.
- Efficient and smart reusability checks by searching the software repository for candidate reusable components.
- Smart quality query and reports by using state-of-the-art software quality metrics along with machine learning techniques to predict quality level of software products.

5. FUTURE WORK

We intend to build a formal framework for identifying potentials, risks, and gaps (PRG) to support the decision of GSE industry of migrating to cloud. This PRG framework will target a set of enterprise-related elements in its evaluation. These enterprise-related elements have direct or indirect relationship with: enterprise size, organizational hierarchy, adopted software process model (e.g. incremental, spiral, agile…etc.), availability of internet connectivity in the development sites...etc.

The PRG framework will take the enterprise-related elements as inputs. Each input will be associated with a determined weight indicating the level of its contribution to the outputs of the framework.

There will be two outputs of this framework. First output will be a decision of whether it is feasible for GSE to migrate to cloud or not. The other output will be a set of mandatory and optional requirements to be fulfilled before migrating to the cloud such as modifying some parts in the current organizational hierarchy and other related recommendations.

6. CONCLUSION

Global Software Engineering (GSE) is a trend of software engineering and project management where software development team is distributed across multiple geographical sites around the globe. GSE promises software industry with huge cuts in development costs and increased productivity by taking advantage of low-cost manpower marketplaces and adopting follow-the-sun workflow [9].

Provision of hardware and software infrastructure needed for setting up a GSE development environment in the form of cloud services can give a huge push for GSE software development activities. In this study, we have reviewed the relevant literature on cloud-based support for GSE and identified a set of potential benefits and risks for supporting GSE with cloud services.

From the result, the list of benefits include: enhancements for productivity, testing, and overall GSE process. It also makes IT operations more cost-effective and reduces the effect of the global distance (i.e. distance results from geographic, cultural, linguistic, and temporal differences) on the GSE environment. It provides better content management and enforces standard adoption throughout the development life cycle.

On the other hand, software development risks include: dependencies between development tasks, unavailability of network connectivity, effects of cloud-based platform on code commitment and integration, lack of policy to propagate changes for a code fragment to its dependent code fragments within the codebase, a variety of cloud configurations for different cloud service providers, security of stored test data, and varying governmental regulations across different countries.

Furthermore, we presented two gaps that need to be tackled to increase the trustworthiness and productivity of cloud-supported GSE. The first gap is the lack rigor for collaboration. Rigorous rules for collaboration will increase the level of trustworthiness in both communication and artifacts exchange between development sites during the GSE life cycle. The second gap is to broaden the scale of software reuse by making cloud supported. This is accomplished by providing many software developers' sizes (i.e. from large-size software companies to individuals) with a cloud service that will make their software source code and executables globally accessible using either offline component-based reuse or online SaaS-based reuse models. This work is first of its kind to provide a platform to develop knowledge in this area.

7. ACKNOWLEDGEMENTS

This research is supported in part by the City University of Hong Kong research fund (Project No. 7200354, 7003032).

8. REFERENCES

[1] *GitHub: Social Coding*. Available from: https://github.com/.

[2] *Google Code*. Available from: https://code.google.com/.

[3] *SourceForge*. Available from: http://sourceforge.net/.

[4] Arimura, Y. and M. Ito, *Cloud Computing for Software Developmnt Environment.* Fujitsu Sci. Tech. J, 2011. 47(3): p. 325-334.

[5] Aspray, W., F. Mayadas, and M.Y. Vardi, *Globalization and offshoring of software.* Report of the ACM Job Migration Task Force, Association for Computing Machinery, 2006.

[6] Avritzer, A., et al., *Coordination implications of software architecture in a global software development project.* Journal of Systems and Software, 2010. 83(10): p. 1881-1895.

[7] Begel, A. and N. Nagappan. *Global Software Development: Who Does It?* in *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*. 2008.

[8] Carmel, E., *Building Your Information Systems from the Other Side of the World: How Infosys Manages Time Zone Differences.* MIS Quarterly Executive, 2006. 5(1).

[9] Carmel, E., J.A. Espinosa, and Y. Dubinsky, *" Follow the Sun" Workflow in Global Software Development.* Journal of Management Information Systems, 2010. 27(1): p. 17-38.

[10] Cataldo, M., *Identifying and Managing Dependencies in Global Software Development*, in *Making Software: What Really Works, and Why we Believe it*. 2010, O'REILLY. p. 349-371.

[11] Chauhan, M.A. and M.A. Babar, *Cloud infrastructure for providing tools as a service: quality attributes and potential solutions*, in *Proceedings of the WICSA/ECSA 2012 Companion Volume*. 2012, ACM: Helsinki, Finland. p. 5-13.

[12] Cocco, L., K. Mannaro, and G. Concas. *A Model for Global Software Development with Cloud Platforms*. in *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. 2012.

[13] Erl, T., R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*. 2013: Pearson Education.

[14] Hanawa, T., et al. *Large-Scale Software Testing Environment Using Cloud Computing Technology for Dependable Parallel and Distributed Systems*. in *Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on*. 2010.

[15] Hartman, P. *ESB Enablement of an International Corporate Acquisition, an Experience Report*. in *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*. 2008.

[16] Hashmi, S.I., et al. *Using the Cloud to Facilitate Global Software Development Challenges*. in *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*. 2011.

[17] Herbsleb, J.D. and D. Moitra, *Global software development.* Software, IEEE, 2001. 18(2): p. 16-20.

[18] Jacobson, I. and P.-W. Ng, *Aspect-oriented software development with use cases (addison-wesley object technology series)*. 2004: Addison-Wesley Professional.

[19] Ji, W., et al. *Agaric - A hybrid cloud based testing platform*. in *Cloud and Service Computing (CSC), 2011 International Conference on*. 2011.

[20] Kaeri, Y., et al. *A Cloud-Based Support System for Offshore Software Development*. in *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*. 2010.

[21] Lian, Y., et al. *Testing as a Service over Cloud*. in *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*. 2010.

[22] Lian, Y., L. Xiaohu, and L. Zhongjie. *Testing Tasks Management in Testing Cloud Environment*. in *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*. 2011.

[23] Oza, N., et al., *Identifying Potential Risks and Benefits of Using Cloud in Distributed Software Development*, in *Product-Focused Software Process Improvement*, J. Heidrich, et al., Editors. 2013, Springer Berlin Heidelberg. p. 229-239.

[24] Riungu-Kalliosaari, L., O. Taipale, and K. Smolander, *Testing in the Cloud: Exploring the Practice.* Software, IEEE, 2012. 29(2): p. 46-51.

[25] Sodhi, B. and T.V. Prabhakar. *Assessing Suitability of Cloud Oriented Platforms for Application Development*. in *Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on*. 2011.

[26] Stankov, I. and R. Datsenka, *Platform-as-a-Service as an enabler for global software development and delivery.* Multikonferenz Wirtschaftsinformatik 2010, 2010: p. 113.

[27] Wisner, J.D., *Principles of supply chain management: a balanced approach*. 2011: Cengage Learning.

[28] Yara, P., et al., *Global Software Development with Cloud Platforms*, in *Software Engineering Approaches for Offshore and Outsourced Development*, O. Gotel, M. Joseph, and B. Meyer, Editors. 2009, Springer Berlin Heidelberg. p. 81-95.