Using the Cloud to Facilitate Global Software Development Challenges

Sajid Ibrahim Hashmi[1], Viktor Clerc[2], Maryam Razavian[2], Christina Manteli[2], Damian Andrew Tamburri[2], Patricia Lago[2], Elisabetta Di Nitto[3], and Ita Richardson[1]
Lero – the Irish Software Engineering Research Centre University of Limerick, Ireland[1]
Department of Computer Science, Faculty of Sciences, VU University Amsterdam, the Netherlands[2]
Department of Electronics and Information at Politecnico di Milano, Italy[3]
{Sajid.hashmi, ita.richardson}@lero.ie[1]
{v.clerc, m.razavian, c.manteli, d.a.tamburri, p.lago}@vu.nl[2]
dinitto@elet.polimi.it [3]

Abstract— **With the expansion of national markets beyond geographical limits, success of any business often depends on using software for competitive advantage. Furthermore, as technological boundaries are expanding, projects distributed across different geographical locations have become a norm for the software solution providers. Nevertheless, when implementing Global Software Development (GSD), organizations continue to face challenges in adhering to the development life cycle. The advent of the internet has supported GSD by bringing new concepts and opportunities resulting in benefits such as scalability, flexibility, independence, reduced cost, resource pools, and usage tracking. It has also caused the emergence of new challenges in the way software is being delivered to stakeholders. Application software and data on the cloud is accessed through services which follow SOA (Service Oriented Architecture) principles. In this paper, we present the challenges encountered in globally dispersed software projects. Based on goals mutually shared between GSD and the cloud computing paradigm, we propose to exploit cloud computing characteristics and privileges both as a product and as a process to improve GSD.**

Keywords- Service Oriented Architecture (SOA), Global Software Development (GSD), Challenges, SaaS (Software as a Service)

I. INTRODUCTION

Advances in technology and communication channels has had a positive impact on business growth as the exchange of information has become more timely, accurate and available. Because of this, business organizations are no longer reluctant to outsource software development and to have development operations in multiple geographical locations. They strive to make use of customized business models to maximize their benefits. In addition, from the marketing perspective, the goals of globally sourced development [10] include making use of international physical and material resources, reducing time to market, and taking advantage of marketing business opportunities.

In the remainder of this introduction section, we highlight the context of this research, the research question, the objective of the research, and the research methodology. Also, we present a synopsis of the cloud computing, challenges faced by GSD, and our motive for using the cloud paradigm to support GSD.

A. Context

In the global environment, outsourcing software development projects to low cost economies is becoming increasingly popular, especially as there is the expectation that companies who embark on GSD strategies will gain and maintain economic advantage through numerous technical and commercial factors [1][2]. This increase in GSD implementation is supported by the availability and accessibility of communication tools as they enhance the options to use a remotely located workforce [3]. The business models in low cost countries have provided capable and willing workers who undertake outsourced and offshore software development [4]. This in turn provides cost reduction in software development projects [5]. However, outsourcing software development to organizations at various outsourcing destinations is not an easy and straightforward task [8][9][10][11] and organizations very often face difficulties due to global distance and the involvement of the development teams which are geographically distributed.

B. Research Question

GSD is software development incorporating teams spread across the globe in different locations, countries, and even continents. We are motivated by the fact that conducting software projects in multiple geographical locations is likely to result in benefits such as cost reduction and reduced time-to-market [14][19], access to a larger skill pool, proximity to customer, and twenty-four hour development by following the sun [60]. But, at the same time, GSD brings challenges to distributed software development activities due to geographic, cultural, linguistic, and temporal distance between the project development teams.

In order to meet the different challenges posed by GSD, we suggest making use of the cloud computing paradigm and illustrate that it has potential to enhance the usefulness of GSD. We argue that different types of geographic and cultural issues can be addressed by making use of different cloud computing realizations such as PaaS (Platform as a Service), IaaS (Infrastructure as a Service), and SaaS (Software as a Service). Since data in the cloud is accessed through services [38], we study its characteristics in the light of Service-Oriented Architecture (SOA). Furthermore, we argue that the cloud can facilitate GSD both as a process and as a product. The former one could have implications for the GSD business model in which service providers are organizations and services are parts of a GSD process, for example, requirements, design, coding, and testing. SOA as a product is developed, run, and distributed globally. The idea is to identify different types and domains of GSD issues and investigate the potential of the cloud to address those.

978-0-7695-4558-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ICGSE-W.2011.19

C. *Objective of the Research*

This paper proposes the development of Global Software Development (GSD) using the cloud computing paradigm, based on our understanding of current GSD and SOA methods from literature, and our overall project aim is to propose the re-construction and improvement of the GSD process. This is done through the use of cloud computing and SOA. We discuss how the GSD process can be aligned with SOA, and how GSD products can be implemented using services. We have established that, for example, some web tools such as Wikis support GSD communication processes. However, we question whether these can be streamlined and re-organized by defining how exactly GSD can work better by making use of a service based environment.

Initially, we identify problem areas in GSD and subsequently, propose the support of GSD development activities through services. The emphasis is on facilitating collaboration activities among GSD teams by structuring those activities. Our rationale is that we can parallel the GSD situation with manufacturing supply-chain management where systems used are composed of ready-to use service-oriented systems. The reason services are widely adopted in industry is because they can be integrated seamlessly. This has resulted in benefits to industry such as increased return on investment and reduced information technology costs [5]. We argue that services to support GSD activities could be developed in the form of service based systems and that what we need are heterogeneous services which could support different development activities. Moreover, output from one service could be taken as input to the next, in cases, where those services supported interrelated activities. In this article, terms like SOA (Service Oriented Architecture) and the cloud have been used interchangeably as different representations of the cloud are being accessed using services.

D. *Research Methodology*

In order to conduct this research, our literature review studied characteristics of services (both SOA and the cloud). We also identified challenges faced by GSD. Following this step, we held a workshop, attended by all of the authors of this paper, each of whom has research and/or industrial expertise in GSD and/or SOA. During this workshop, through interactive discussion and brainstorming, we developed the concepts presented in this paper. To do this, we summarized the GSD challenges and requirements and investigated the potential of SOA based cloud services [47] to address these. We are embarking on further research to understand whether these indeed can be of value to both the industrial and research communities.

E. *Cloud Computing*

Cloud computing is an internet based computing paradigm in which shared resources like software, hardware, and information are provided to the subscribers on demand [17][18][26]. NIST [55] defines cloud computing as a model for enabling convenient and on demand network access to shared computing resources that can be managed and provided rapidly with minimal effort. The aim is to construct a low cost computing system by using certain entities without compromising on computing capabilities. Depending on the type of shared resources, the cloud paradigm can have different implementations like IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service), to dispense computing capacity to end users.

Infrastructure as a Service (IaaS) includes the delivery of hardware such as processors and storage as a service, e.g., Amazon Elastic Cloud (EC2) and Simple Storage Service (S3). In other words we can say that it delivers a platform utilization environment as a service. Instead of physically purchasing hardware and software infrastructure, clients buy such resources as a fully outsourced service.

In addition to the infrastructure, Platform as a Service (PasS) occurs when a software platform is provided on which systems can be run. This includes the delivery of programming platforms and tools as a service. This kind of cloud computing provides a development environment and the infrastructure provider's equipment can be used to develop programs which are delivered to end users through internet and servers.

Software as a Service (SaaS) occurs when applications are delivered as services using IaaS and PaaS. This implementation of the cloud focuses on separating the ownership and possession of software from its use [18]. It is based on the idea that software functionality could be provided as set of distributed services that could be configured and bound at delivery time, to avoid the current limitations with software use, deployment, and evolution [18]. Since cloud computing stimulates the provision of online services via the World Wide Web, software can be hosted on web servers as services [18]. Thus, the advent of SaaS within the cloud computing paradigm has created new opportunities for organizations to communicate and coordinate among themselves.

F. *GSD Challenges*

With the emergence of technologies in a world which has become increasingly globalized, the relationship between culture and management of remote work has become an unavoidable issue which needs to be addressed [15]. Because of distance among the software development teams, GSD encounters certain challenges in terms of collaboration [61], communication [62], coordination [63], culture [64], management [65], organizational [66], outsourcing [35][67], development process [68], development teams [16][69], and tools [29][70].

Global distance comprises of four elements: geographic, cultural, linguistic, and temporal distance [57][58]. Geographic distance occurs as the teams are dispersed across countries. Cultural distance occurs due to teams being made up of members from different cultures, and the additional expectation that each member will understand and support each other's culture. When team members speak in different languages, there needs to one chosen language for work purposes, and as this is everyone's first language, linguistic distance occurs. As teams are geographically dispersed, there is the additional difficulty of temporal distance – members working across different time zones [49][50]. Each of these differences individually causes problems within GSD teams, and the culmination of these differences into global distance can and do impede global software development projects [12][13]. Thus, the management of globally outsourced software development has been accepted as a difficult and complex task [14]. These four types of GSD challenges are addressed using the SOA based cloud services (Table 1).

G. *Motive for Using the Cloud for Supporting GSD*

One of the missions [59] of the cloud architecture is to provide services to customers by not only managing them but optimizing them by taking into consideration economies of scale. The cloud model is composed of three service models (IaaS, PaaS, and SaaS), five essential characteristics, and four deployment models [59]. The cloud deployment models - Private, Community, Public, and Hybrid - define the scope of the cloud solution. The cloud model is discussed in terms of creation and provision of services [20] which means that

TABLE I. GSD CHALLENGES POTENTIALLY FACILITATED BY THE USE OF SERVICES

Collaboration Challenges	Issues	Negative Impact on Software Project	Facilitating GSD Using Services (SOA/Cloud)
Geographic	Distance Time Knowledge transfer Tools	Communication gaps Project Delays Ambiguity on technical aspects Unequal quality levels across the software development sites	Dynamic binding, runtime adaptation, and timely availability of required services could help dealing with geographic issues. Also, availability of SaaS could diminish installation overheads at each development location.
Cultural	Unequal distribution of work Lack of Trust, Fear	Increase in cost Poor skill management Reporting problems	Service could maintain a fair distribution of work between the teams. Only a specific person will be responsible for the task assigned to thus skill management would be easier too.
Linguistics	Frequency of communication Knowledge transfer	Loss in project quality Invisibility on project development Ineffective project management	Run time evolution of services can meet with the linguistic issues. Also, isolation of each task and related information as a service can ensure right level of knowledge transfer.
Temporal	Lack of Motivation Less visibility Risk	Loss in project quality Poor management of configuration Chances of project artifact loss	The cloud service models imply that the data resides on a centralized location where inventory of services is maintained. Services maintain a registry where all of them are stored. This attribute could be used to store and retrieve configurations.

it supports services. Since SOA runs a mechanism for development and management of distributed dynamic systems and it evolved from the distributed component based approach [21], we argue that it has potential to cater the challenges of GSD where a project is developed across different geographical locations. Our thesis is that GSD challenges can be overcome through Service Oriented Architecture (SOA) support. This will contribute increased interoperability, diversification, and business and technology alignment. Moreover, the vision behind this architectural paradigm is to set up common goals and objectives to improve the collective effectiveness of the enterprises participating in globally distributed projects. Since software processes are software too [22], we argue that the cloud has potential to reinforce GSD as a process. Initially, we considered the use of standard procedures to meet the quality challenges posed by GSD. But, since organizations have to interact dynamically in global environments, these standard procedures cannot scale up to support dynamism (which is a main feature of SOA). Moreover, the ideology posed by both SOA and GSD is somehow similar [1, 23], for example, coordination, transaction, context, execution monitoring, and infrastructure. In addition, SOA is one of the main technical foundations of the cloud [51].

For GSD, the use of collaboration tools among teams is not new. Existing research has already proposed further work in this regard [7][24][25]. We adopt the idea of SaaS for GSD to make use of properties of both cloud and SaaS, such as reusability, reliability, extendibility and inexpensiveness [27][28]. Teams with frequent communications among their members are likely to collaborate better. Thus, this frequent communication is important to make full use of GSD advantages, e.g. improved productivity, reduced time to the market, and reduced cost. However, oral communication is prone to confusion and misunderstanding. One way could be to minimize the need for communication but such strategy would emphasize on the involvement of more dedicated personnel from each development site [48] which could not be feasible either. At the same time it is important for the communication media to be formal, flexible, and evolvable to ensure the collaboration mechanisms work effectively.

GSD teams also need to collaborate effectively and the attributes of the cloud paradigm, especially SaaS, can be used to facilitate efficient collaboration between geographically distributed teams during software development phases such as requirements, design, coding, and testing. The characteristics and the architecture of the cloud model itself has the potential to fulfill the GSD task requirements. For example, cloud deployment models allow certain trusted partners (which could be GSD team members) to share resources among themselves. Service models may not only provide access to collaboration and productivity tools but also allow network access to computing resources, and the "use as you go" feature is likely to reduce the overall project costs across multiple development sites as computing resources and infrastructure is not required up-front.

We investigate the impacts of the aforementioned collaboration challenges and suggest the likelihood of using the cloud to address them. We expect to achieve efficiency in collaboration through using the cloud in different implementations. The essence of using this paradigm to facilitate GSD is that instead of acquiring and owning the software and project data, GSD team members can access and subscribe to some of the software at a time (according to the need) in the form of services. In addition, we want to take advantage of the SOA characteristics [43] like loose coupling, service composition and negotiation to facilitate a similar level of development practices across multiple sites. Moreover, the service provider and user are important to the technical and economic changes made possible by cloud computing. In our model, this concept of provider and consumer is similar to the SOA paradigm.

II. GSD AS A SERVICE ON THE CLOUD

In this section, we describe how GSD as a service can facilitate and improve how GSD is carried out. We discuss certain GSD challenges and provide a rationale as to how the cloud service models can address them respectively.

A. *GSD Services Concept*

Figure 1 illustrates the concept of using the cloud paradigm to support GSD. Service standards and policies are defined by engineering and project management personnel. Different GSD development sites (represented as GSD1, GSD2,…,GSD5) are deployed on a private cloud which covers all geographically distributed development teams. We propose that this concept can support the reduction of difficulties caused by global distance. For example, the use of services itself reduces the distance factor to meet

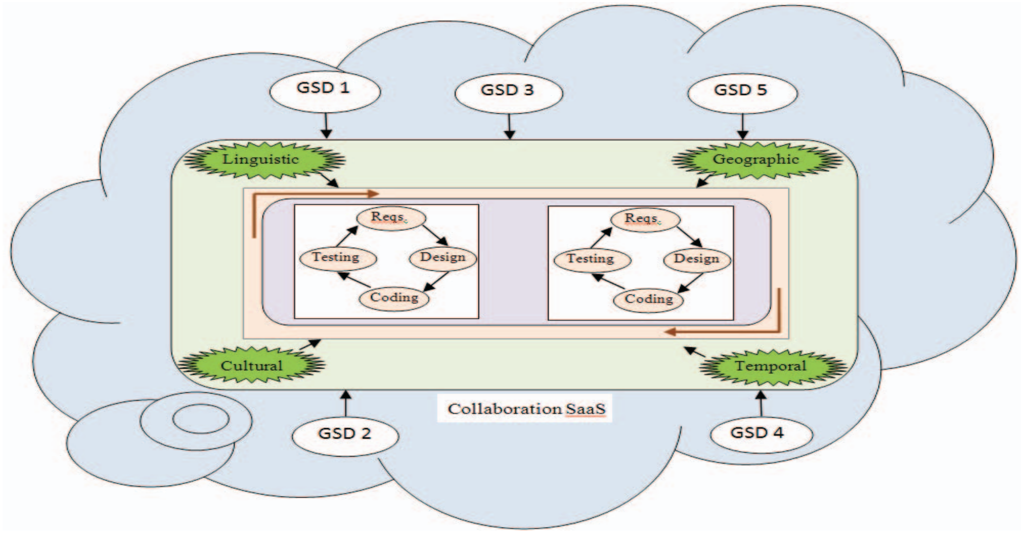

Figure 1. Using the cloud to support GSD product and process activities

geographic and temporal challenges. As far as cultural and linguistic challenges are concerned, the provision of multilingual services based on the location of GSD teams could improve the problem. We consider an example scenario to understand the GSD collaboration challenges that could be minimized using the cloud paradigm. Suppose that an organization in Ireland (GSD1) outsources a software component development to a company in Germany (GSD3).

As part of the project requirement, people in both countries have to communicate to exchange information on different development phases and tasks. The project manager from GSD1 sends on some important instructions regarding requirements and architecture of the potential system. Not only should this information be conveyed to the concerned team member but there should be some assurance that it has indeed reached them, without the risk of being lost or disclosed to other GSD locations. But, the concerned team member is a novice in the language and also needs to have those instructions translated into his local language. Thus, a translation service is required on both sides to facilitate the task.

One would argue that the translation task could be facilitated by a simple word parser, and the use of cloud and services seemed to be irrelevant, but the situation is not as simple as it appears to be. Communications between the teams could involve some other artifacts such as design documents, code snippets, and legal and financial negotiations. All of this could be made available on the cloud in the form of services which could be accessed by the authorized team members. Also, cloud services can evolve with changes in the associated business [18], for example, such a change might occur in the form of a financial or money transfer service after both companies agree on terms and conditions. Using our proposed system would result in the elimination of GSD3 overhead, i.e. storage of project artifacts and information, as everything would be stored on cloud infrastructure and would be accessible from there in the form of services. Table 2 list down the characteristics [46][55] of the Cloud which can be potentially beneficial for GSD.

TABLE II. SUPPORTING CHARACTERISTICS OF CLOUD COMPUTING

Virtualization	Courtesy of this privilege, cloud providers can enhance their infrastructure to accommodate in case there is growing demand for services. Usually, a combination of hardware and software are used on the provider side to meet with the scaling requirements.
Reduced Cost	Costs in the cloud do not include server side infrastructure and equipment costs. Moreover, pay as you go model ensures that subscribers are bound to pay for only those resources which they use. In short, the distribution costs of software are reduced.
Scalability	On-demand provision of application software provides scalability, which results in greater efficiency. Whereas cloud based application development platforms provide with high level of scalability thus making the developed application to coup with the fluctuation demands.
Infrastructure	Providers' applications are run on a cloud infrastructure from where a consumer can access those. Similarly, consumer-modified information or application can be deployed on the same infrastructure as well. The privilege is that the consumer does not have to deal with the underlying infrastructure.
Performance	The cloud paradigm can support various levels of performance requirements like service scaling, response time, and availability of the application based on the needs of the consumers. In addition indirect performance measures may also be achieved by eliminating the overheads involved with installation procedures and reduction in unnecessary reduction among the applications running on the cloud.
Multi Tenancy Support	Public clouds are elastic in nature as their consumers are not limited. More importantly, consumers' workloads are isolated to provide privacy. However, the number of consumers can be restricted by opting out a specific deployment model.

A. GSD Challenges and Requirements

During our workshop, we identified GSD challenges and requirements which could be, in our view, solved through using cloud architecture.

1) Coordination

Coordination among distributed teams is important to GSD but geographic distance negatively affects the ability to collaborate [23]. For building complex systems, coordination requires interaction over sequences of operations. However, often, due to collaboration within different time zones, employees have less time to coordinate their work.

As a product, cloud services ensure interactions among different activities. For example, interaction between the service consumer and provider on finding and binding of services is independent of the geographical distance. SOA puts an emphasis on adding transactional guarantees to facilitate the interaction in the coordination framework [31]. For example, standards have been proposed by IBM [32][33] and suggested by Sun [34].

Since cloud computing is the key service delivery platform in the field of service computing [44], *as a process*, it could allow resource sharing not only for infrastructure and application resources, but also for software resources and business processes [45][46]. These advantages are likely to support different disciplines, for example, Infrastructure as a Service (IaaS) could help provide different GSD teams with resources such as computing power or storage provisioning to store project related data. Software resources may consist of middleware and development resources like application systems, database servers, and operating systems. The advantage of using fist two types of resources as a service is that they are never wasted after the project is over - instead, they can be unsubscribed. Application resources could assist in providing SaaS with necessary interfaces that could facilitate collaboration and sharing of information among the teams.

B. Support of Technical Development

A variety of special purpose services can be used for *process* related software development activities e.g. requirements, design, and testing. Services which support different process activities, can be combined together to facilitate the whole process. As shown in Figure 2, supporting development process activities in the form of services can help alleviate geographic and distance challenges.

As a product, shifting the provision model from Software as an Application to SaaS removes the dependencies and challenges in terms of architecture and task dependencies that traditional software development and reuse models impose. Moreover, it can reduce cost by facilitating reuse of services which provide similar operability for software application development. Hence the development is reduced on building similar business applications as the only challenge which remains is the identification of suitable services which can serve the purpose required.

C. Geographical Distance

Physical distance removes the opportunity for face to face communication. *As a product*, the Platform as a Service representation can provide a development platform with set of services to assist application development and hosting on the cloud. It does not require any kind of software downloads and installations [2], and because of its characteristics, has the capacity to support geographically distributed teams. Moreover, the philosophy of the cloud paradigm is to facilitate a pool of shared hardware and software resources.

Facilitating global software development activities *as a process* in the form of services, can overcome many software limitations involving software evolution, reuse, and deployment. Such a model is likely to open not only new opportunities for the business but also the way software is being developed, i.e. services become part of GSD processes being provided by the outsourcing organizations.

D. Global Project Optimization

In GSD, it is important to share the information in terms of work performed by distributed teams. Communication and awareness capabilities should be provided by integrating this information not only into a collaborative environment [1], but also to maintain a rich “project memory” [34].

Provision of this information exchange on software development activities *as a process* is likely to reduce the software installation costs across different development sites. In addition, it can make collaboration more instant and flexible because of the customization and scalability attributes of the cloud.

SOA not only manages service execution and output information, but also keep track of the new information without any changes to the underlying infrastructure [23]. This unique feature can ensure team management and coordination by means of its use *as a product*, by scaling on to the existing project information.

E. Optimizing Globally Distributed Software Development

It is true that geographic distance affects the ability to collaborate [1]. Moreover, it has been reported that communication and collaboration declines as the distance between the two working location increases [28]. *As a process,* cloud based collaboration among GSD teams is likely to diminish the deficiency caused by distance as services are free from geographical boundaries. Yet another type of resource in the cloud could be business process [45], which may facilitate the optimization of the overall technical software development.

As a product, it can serve as an intermediary to facilitate users to access and communicate with the cloud. The services involved in a system can change with the change in the associated business in terms of requirements, and can perform this change dynamically. The reason for change in GSD could be the availability of yet another programming task or a need to collaborate on a task which is already underway.

F. Eliminating the Strategic Issues

In GSD, ownership is often lacking [1]. Service ownership is a concept which allows the service users to focus on their core activities; it also helps the service provider with an opportunity to take advantage of economies of scale [56]. *As a product*, well defined ownership exists for each service, and GSD users benefitting from such services can enjoy this privilege. This ensures that services are used in a way to give the most to a business.

As a process, SOA addresses ownership by collocating service provisioning by service development. The wrapping of a GSD task into an independent service can promote ownership, as it can be used exclusively. In this case, the outsourcing company (the provider) could be the one to convey the project requirements or architecture knowledge to a specific GSD team or to a single member without notifying others. Thus, it can incorporate privacy by increasing the feeling of ownership.

G. Enhancing Communication Among Teams

The structure of multi-site software development mimics the team structure [37]. The main distinctive feature of the cloud is that it allows rapid elasticity, making it straightforward for the service provider to dimension the resources necessary to support a service dynamically depending on the service demands [39]. Thus, investigating the potential interactions among the stakeholders would enable getting insight into the service creation process for collaboration among GSD teams. These interactions are likely to be among the outsourcing organizations and the teams jointly working on the same project.

As a product, appropriate service definitions (including their descriptions) may act as proxies for communication and hence may reduce the need for cross site communication. This privilege could be useful when teams from different time zones find it difficult to collaborate. On a technical level, the SOA paradigm provides an appropriate mechanism for cross platform data exchange and sharing by message passing, service search and collaboration [30]. In addition, the SOA and the XML-described data, information and knowledge can combine the different loosely coupled subsystems.

H. Managing Project Knowledge Transfer

The transfer of requirements and architecture knowledge across development sites is an issue in GSD. Services are likely to wrap this knowledge using the correct abstraction level. Using services *as a product* diminishes the need for sharing knowledge as the constituent services have sufficient description about themselves, and because this knowledge is developed locally. Two knowledge transfer issues in GSD which exist are requirements and architectural knowledge. With SOA, a sufficient description and transfer of the requirements knowledge diminishes the need for transfer of other forms of knowledge. Moreover, coordination aspects are hardly needed in services as they are isolated.

In terms of *as a process*, isolating any task as a service helps to identify right abstraction level for the transfer of such information within the task *as a service.* This form of service provision could take care of how requirements should be provided, what the outcome will be, and what should a GSD team member expect from others. Hence it can help managing the knowledge of a distributed project.

I. Execution Monitoring

In GSD, the potential order in which components interact should dictate the decision on the interaction among the corresponding teams [1]. Moreover the use of Application Programming Interfaces (APIs) promotes isolation and reduced information sharing [36]. A crucial aspect is responding to the constant changes in the business requirements. *As a product*, SOA can ensure the correct order of service execution by a central scheduler, which controls the execution of the service and consequently the right order of communication between the project partners in GSD. *As a process*, these services may share their execution context among each other to guarantee the correct execution order. To support the collaboration activities among geographically distributed teams, the concept of execution monitoring can be used as the basis for designing the collaboration process.

J. Eliminating Project and Process Management Issues

In GSD, a centralized configuration management system should be made available to manage project artifacts produced out of components being developed across multiple locations. *As a product*, this task can be facilitated by service registries which serve as databases for services. Potential users can find and bind any service using the service description provided with these registries. Or, alternatively, since services are a black box, their use is likely to eliminate the need for centralized configuration management within their scope.

Cooperation and coordination is required to obtain trust between two or more parties [52]. Lack of trust is always likely to reduce the team cohesion. Teams with higher trust are coordinate better to achieve better performance [53] which could make management an easier task. A goal of cloud computing is that its users must be able to access its different implementations at any time [54].

K. Technical Issues

In GSD, a modular approach for software development has been suggested [1], but dependencies are likely to exist among components in the running version of the software. This nature of such a project would require evolvable software which could cope with the challenge of component and functional dependencies. *As a product*, services are loosely coupled and independent in nature, with minimum dependencies among them. This characteristic can not only minimize the task dependencies but ultimately eliminate the risk factor because, in case of a service failure, the failed component can be replaced with another one at run time. This dynamic replacement is one of the distinguishing features of services.

Testing software is usually the most costly phase in software development and it can be responsible for over 50 percent of development costs [41]. Therefore, this phase often becomes responsible for the ultimate profitability of the product [42]. Carrying out testing activities correctly is important as the quality assurance, financial incentives, and customer satisfaction of the end product often depend on the testing activity [40]. Making use of standard procedures to meet the quality challenges posed by GSD is important, but since organizations have to interact dynamically in global environment, these standard procedures often cannot scale up to support dynamism. On the other hand, dynamism is supported by SOA by means of run time evolution and on demand provision. Facilitating collaboration on testing in the form of the services *as a process* could ensure higher quality levels.

In normal circumstances, different software tools [29][70] are used to facilitate not only GSD development but also collaboration among the teams. Unavailability of tools at the right time or version misalignment can cause delays in global software projects. Use of collaboration tools *as a service* can reduce the overhead of tool installation. Using the SOA paradigm, tools, data, and workspace could be stored and accessed from the cloud, thus eliminating the need for tool installation. The purpose is not only to cater with the version issue, but also provide GSD teams with all tools required for the project.

III. DISCUSSION

In order to propose the use of cloud for GSD, it is important to have a comprehensive understanding of the GSD processes. We do not expect that all will be served by this technological paradigm, but we do believe, if designed correctly, GSD can be successfully supported by services. For example, in a context where different GSD locations are inter-connected and are using the cloud, all of them may not have the same level functional needs. Determining different level of needs for service provision could be one of the major concerns among different GSD locations.

The concept of SaaS itself continues to be subject to evolution and revision. In addition, the availability and subscription of these services because of different types of dependency relationships among cloud users (tenants) could be considered as a challenge for using the cloud to support GSD. Moreover, in terms of project knowledge transfer across global software development sites, the right level of abstraction of the useful codification as well as the reduction in tacit knowledge will remain an issue. Since the main usage in services comes in connecting pieces of information, sharing services across different domains and enterprises is also likely to result in further security issues.

IV. CONCLUSIONS

We have proposed using the cloud paradigm to meet with different challenges posed by Global Software Development (GSD). We are suggesting that this will result in GSD benefitting from the cloud's infrastructure, platform, and provision of software as a service features. Information and data on the cloud is transmitted and shared by means of web services which work on underlying Service Oriented Architecture (SOA) principle.

We argue that the cloud paradigm has the potential to turn over a few more unturned stones of GSD issues which are a significant hurdle for the development of successful projects in the GSD situation. But, we are planning to develop our ideas further with a view to filling the gap between technical proficiency and meeting the needs of developers. So, like SOA, we cannot expect the cloud paradigm to address some psychological and social issues like *trust* but we can reduce their negative impact through the use of this model.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube). It was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero – the Irish Software Engineering Research Centre (www.lero.ie), and a Higher Education Authority grant PRTLI 4 to the Lero Graduate School in Software Engineering.

REFERENCES

[1] J. D. Herbsleb, "Global software engineering: the future of socio-technical coordination," in Proceedings of the Future of Software Engineering (FOSE'07), 2007, pp. 188-198.

[2] R.E. Grinter, J.D. Herbsleb, and D.E. Perry, "The geography of coordination: dealing with distance in R&D work," in Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP '99), ACM Press, New York, 1999, pp. 306–315.

[3] J. A. O'Brien, "Management information systems: managing information technology in the business enterprise," Mc Graw Hill Irwin, 2002.

[4] S. S. Toaff, "Don't play with "Mouths of Fire," and other lessons of global software development." Cutter IT Journal 15(11), 2002, pp. 23-28.

[5] E. Carmel and P. Tjia, "Offshoring information technology: sourcing and outsourcing to a global workforce." Cambridge, UK, Cambridge University Press, 2005.

[6] A. J. Espinosa and E. Carmel, "The impact of time separation on coordination in global software teams: a conceptual foundation" Software Process Improvement and Practice, 8(4), 2003, pp. 249 –266.

[7] A. Sarma and V. D. Hoek, "Towards awareness in the large," in Proceedings of the International Conference on Global Software Engineering, 2006, pp. 127-31.

[8] E. Carmel, "global software teams: collaboration across borders and time zones". Saddle River, NJ, Prentice Hall, 1999.

[9] D. W. Karolak, "Global software development: managing virtual teams and environments." Los Alamitos, CA, USA, IEEE Computer Society Press, 1999.

[10] J. D. Herbsleb and D. Moitra, "Global software development." IEEE Software 18(2), 2001, pp. 16-20.

[11] V. Clerc, P. Lago, and H. V. Vliet, "The architect's mindset," 3rd International Conference on the Quality of Software Architectures, Volume 4880 of Lecture Notes in Computer Science, 2007, pp. 231-249, Springer Berlin / Heidelberg.

[12] E. Carmel, "Building your information systems from the other side of the world: how Infosys manages time differences," Management Information Systems Quarterly -MIS Quarterly Executive, vol. 5, no. 1, Mar 2006, pp. 43-53.

[13] V. Casey, S. Deshpande, and I. Richardson, "Outsourcing and offshoring software development: the remote developers' perspective," Global Sourcing Workshop, Val d'Isere, France March 2008.

[14] F. Lanubile, D. Damian, and H. L. Oppenheimer, "Global software development: technical, organizational, and social challenges," SIGSOFT Software Engineering Notes 28(6): 1 - 4.

[15] R. T. Watson, T. H. Ho, and K. S. Raman, "Culture: a fourth dimension of group support systems," Communications of the ACM, 37-10, 1994, pp. 44-55.

[16] C. M. Beise, "IT project management and virtual teams," in Proceedings of the 2004 SIGMIS Conference on Computer Personnel Research: Careers, Culture, and Ethics in A Networked Environment (Tucson, AZ, USA, April 22 - 24, 2004). SIGMIS CPR '04. ACM, New York, NY, pp. 129-133.

[17] J. Yang and Z. Chen, "Cloud computing research and security issues," International Conference on Computational Intelligence and Software Engineering *(*CiSE*)*, 2010, pp. 1-3.

[18] M. Turner, D. Budgen, and P. Brereton, "Turning software into a service," Computer, vol.36, no.10, Oct. 2003, pp. 38- 44.

[19] S.-o Setamanit, W. Wakeland, and D. Raffo, "Improving global software development project performance using simulation," Portland International Center for Management of Engineering and Technology, 5-9 August, 2007, pp. 2458-2466.

[20] K. Zhang, X. Zhang, W. Sun, H. Liang, Y. Huang, L. Zeng, and X. Liu, "A policy-driven approach for software-as-services customization," in Proceedings of the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, CEC/EEE 2007, pp.123-130.

[21] A. Bertolino and A. Polini, "SOA test governance: enabling service integration across organization and technology borders", IEEE International Conference on Software Testing Verification and Validation Workshops.

[22] L. Osterweil, "Software processes are software too," in Proceedings of the 9th International Conference on Software Engineering, IEEE Computer Society Press Los Alamtos, CA, USA.

[23] S. Dustdar and W. Schreiner, "A survey on web services composition", International Journal of Web and Grid Services, vol. 1, no. 1, 2005, pp. 1-30.

[24] L. Cheng, C. DeSouza, S. Hupfer, J. Patterson, and S. Ross, "Building cllaboration into IDEs," ACM Queue, vol.1, no.9, 2004, pp. 40-50.

[25] A. Sarma, Z. Noroozi, and A. V. D. Hoek, "Palantír: raising awareness among configuration management workspaces," in Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 444-454.

[26] S. Zhang; S. Zhang, X. Chen, and X. Huo, "Cloud computing research and development trend," in Proceedings of the 2nd International Conference on Future Networks, 2010, pp. 93-97.

[27] J. Yang and Z. Chen, "Cloud computing research and security issues," in Proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE), 2010, pp. 1-3.

[28] T.J. Allen, "Managing the flow of technology," Cambridge, MA: MIT Press, 1977.

[29] R. Martignoni, "Global sourcing of software development – a review of tools and services", in Proceedings of the 4th IEEE International Conference on Global Software Engineering, 2009, pp. 303-308.

[30] M. P. Papazoglou, "Web Services and Business Transactions", World Wide Web, vol. 6, no. 1, March 2003, pp. 49-91.

[31] F. Cabrera, G. Copeland, B. Cox, T. Freund, J. Klein, T. Storey, and S. Thatte, "Web services transaction specifications," available online, last accessed 16 August, 2005: http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/

[32] F. Cabrera, et al., "Web services coordination (WS-coordination)," Version 1.0, available online, last accessed August 2005: ftp://www6.software.ibm.com/software/developer/library/ws-coordination.pdf

[33] D. Bunting, M.C.O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, "Web services transaction management" (WS-TXM), Ver1.0. http://developers.sun.com/techtopics/webservices/wscaf/wstxm.pdf

[34] D. Cubranic and G. Murphy, "Hipikat: recommending pertinent software development artifacts," in Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 408-418.

[35] M. Jensen, S. Menon, L.E. Mangset, and V. Dalberg, "Managing offshore outsourcing of knowledge intensive projects – A people centric approach," in Proceedings of the 2nd International Conference on Global Software Engineering, 2007, pp. 186-196.

[36] C. R. B. D. Souza, D. Redmiles, L. T. Cheng, D. Millen, and J. Patterson, "Sometimes you need to see through walls: a field study of application programming interfaces," in Proceedings of the 2004 ACM conference on Computer supported cooperative work, pp. 63-71.

[37] J. D. Herbsleb and R. E. Grinter, "Splitting the organization and integrating the code: Conway's law revisted," in Proceedings of the 21st International Conference on Software Engineering, 1999, pp. 85-95.

[38] L. W. Pires, L. F. Wombacher, A. V. Sinderen, and M. J. Chihung Chi, "Stakeholder interactions to support service creation in cloud computing," in Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010, pp. 173-176.

[39] Armbrust et. al., "Above the clouds: a Berkley view of cloud computing," EECS Department, University of California, Berkeley Technical Report No. UCB/EECS-2009-28 February 10, 2009.

[40] E. Kit, "Software testing in the real world: improving the process", Addison-Wesley, Reading, MA, USA, 1995.

[41] H. Do and G. Rothermel, "An empirical study of regression testing techniques incorporating context and lifetime factors and improved cost-benefit models," in Proceedings of 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2006, pp. 141-151.

[42] J. Kasurinen, "Elaborating software test processes and strategies," in Proceedings of the 3rd International Conference on Software Testing, Verification, and Validation, 2010, pp. 355-358.

[43] L. J. Zhang, "EIC editorial: introduction to the body of knowledge areas of services computing," IEEE Transactions on Service Computing, vol. 1, no. 2, April-June, 2008, pp. 62-74.

[44] L. J. Zhang, J. Zhang, and H. Cai, "Services computing, core enabling technology of the modern services industry," published by Springer and Tsinghua University Press, 2007.

[45] L. J. Zhang and Q. Zhou, "CCOA: cloud computing open architecture," in Proceedings of the 7th IEEE International Conference on Web Services, 2009, pp. 607-616.

[46] R. Guha, and D. Al-Dabass, "Impact of Web 2.0 and cloud computing platform on Software Engineering," in Proceedings of the International Symposium on Electronic System Design (ISED), 2010, pp.213-218.

[47] J. Schaper, "Cloud Services, " in Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies, 2010, pp. 91.

[48] A. Elfatatry and P. Layzell, "Software as a service: a negotiation perspective," in proceedings of the 26th International Conference on Computer Software and Applications Conference, COMPSAC 2002. pp. 501-506.

[49] A. Avritzer, D. Paulish, Y. Cai, and K. Sethi, "Coordination implications of software architecture in a global software development project," Journal of Systems and Software 83(10), pp. 1881-1895.

[50] P. Hartman, "ESB enablement of an international corporate acquisition, an experience report," in Proceedings of the 3rd IEEE International Conference on Global Software Engineering, 2008, pp. 200-204.

[51] M. D. Dikaiakos, G. Pallis, D. Katsaros, P. Mehra, and A. Vakali, "Cloud computing: distributed internet computing for IT and scientific research," Internet Computing, vol. 13, no. 5, pp. 10-13.

[52] E. E. Jennings, "Routes to the executive sites," New York: Mcgraw-Hill, 1971.

[53] S. L. Jarvenpaa and D. E. Leidner, "Communication and trust in global teams," Journal of Computer Mediated Communication, vol. 10, no. 6, 1999, pp. 791-815.

[54] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: a survey," in Proceedings of the 6th International Conference on Semantics, Knowledge, and Grids, 2010, pp. 105-112.

[55] L. Badger, T. Grance, R. P.-Comer, J. Voas, "Draft cloud computing synopsis and recommendations," National Institute of Standards and Technology, Special Publication 800-146, May 2011.

[56] http://itsm.certification.info/ownership.html

[57] A. Begel, N. Nagappan, C. Poile, L. and Layman, "Coordination in large-scale software teams," ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009, CHASE '09, pp. 1-7.

[58] A. Begel, N. Nagappan, "Global software development: who does it?," in Proceedings of the 3rd IEEE International Conference on Global Software Engineering, 2008. pp.195-199.

[59] M. Behrendt, et al., "Introduction and architecture overview: IBM cloud computing reference architecture 2.0," Draft Version V1.0, 2011.

[60] T. Nguyen, T. Wolf, and D. Damian, "Global software development and delay: does distance still matter?," in Proceedings of the 3rd IEEE International Conference on Global Software Engineering, 2008, pp. 45-54.

[61] P. Mohapatra, P. Björndal, and K. Smiley, "Causal analysis of factors governing collaboration in global software development teams," in Proceedings of the 5th IEEE International Conference on Global Software Engineering, 2010, pp. 128 – 132.

[62] T. Niinimaki, A. Piri, and C. Lassenius, "Factors affecting audio and text-based communication media choice in global software development projects", in Proceedings of the 4th IEEE International Conference on Global Software Engineering, 2009, pp. 153-162.

[63] M. Cataldo, M. Bass, J. D. Herbsleb, and L. Bass, "Factors affecting audio and text-based communication media choice in global software development projects", in Proceedings of the 4th IEEE International Conference on Global Software Engineering, 2009, pp. 153-162.

[64] V. Casey, "Leveraging or exploiting cultural difference?", in Proceedings of the 4th IEEE International Conference on Global Software Engineering, 2009, pp. 8 - 17 .

[65] V. Casey and I. Richardson, "Project management within virtual software teams", in Proceedings of the International Conference on Global Software Engineering, 2006, pp. 33-42.

[66] D. Damian, F. Lanubile, and H. L. Oppenheimer, "Addressing the challenges of software industry globalization- the workshop on global software development," in Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 793-794.

[67] R. Heeks, S. Krishna, B. Nicholsen, and S. Sahay," Synching or sinking-global software outsourcing relationships," IEEE Software, vol. 18, Issue. 2 , 2001, pp. 54 - 60.

[68] H. Klein, A. Rausch, and E. Fischer, "Process-based collaboration in global software engineering," in Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009, pp. 263 - 266.

[69] H. K. Edwards and V. Sridhar, "Analysis of the efectiveness of global virtual teams in software engineering projects," in Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.

[70] J. Portillo-Rodriguez, A. Vizcaino, C. Ebert, and M. Piattini "Tools to support global software development processes: a survey", in Proceedings of the 5th IEEE International Conference on Global Software Engineering, 2010, pp. 13 - 22.