Platform-as-a-Service as an Enabler for Global Software Development and Delivery

Ivo Stankov, Rastsislau Datsenka

Lehrstuhl für Wirtschaftsinformatik,
Europa-Universität Viadrina Frankfurt (Oder)
August-Bebel-Str. 12, 15234 Frankfurt (Oder)
{stankov|datsenka}@euv-frankfurt-o.de

1 Introduction

The convergence of several technological, organizational and political trends such as outsourcing, distributed software work, the emergence of internet-based collaborative environments, and the expansion of the global delivery of IT services, made a new offshoring strategy possible. This is described in this paper as Global Software Development and Delivery (GSDD).

At the same time, the emergence of the Platform-as-a-Service (PaaS) concept in the last couple of years has opened new horizons for software outsourcing and offshoring in general and for GSDD in particular. This is a result of the establishment and proliferation of the Software-as-a-Service delivery paradigm and the expansion of web-based development and collaboration.

This paper focuses on the Platform-as-a-Service concept and aims to explore its current and potential technological capabilities as an enabler for innovative service-based global software development and delivery strategies.

The paper begins with an introduction to the Global Software Development and Delivery concept with reference to the IT Infrastructure Library (ITIL) and discusses the origins of PaaS. Next, using deductive argumentation, recommendations for an on-demand platform are made based on the technical and organizational considerations about GSDD. In the next section the current PaaS market is briefly examined with the aim to estimate its suitability to support GSDD activities at the present level. Finally, concluding remarks are made about the findings.

2 Global Software Development and Delivery

The term Global Software Development and Delivery describes the workflow and governance of distributed IT offshoring projects. GSDD exploits time, geographical location (onsite, near- and offshore), and collaborative techniques with the aim to reduce costs and to optimize the quality of software products (Kurbel and Datsenka 2009).

The three major elements of GSDD as described by IBM (Parvathanathan et al. 2007, p. 5) are geographically shared software development, global delivery, and seamless collaboration between production teams. This paper focuses mainly on its delivery component and references the ITIL V2 as a sample model of investigation. The choice in favor of the older ITIL V2 is motivated by the fact that it specifically identifies and describes service delivery as a distinctive entity within the software lifecycle, whereas ITIL V3 introduces a different framework which does not explicitly refer to the delivery component (Bon 2008, p. 4).

In terms of ITIL, development is an integral part of application management and includes requirements analysis, design, implementation, operation, and optimization, whereas delivery describes capacity, availability, continuity, and financial management (Cusick and Prasad 2006, p. 33). Additional to development and delivery, a collaborative environment is an extra component of GSDD that is also a part of the IT Infrastructure Library and works as a bridge that binds globally distributed teams. Cusick and Prasad (2006, p. 97) describe such an environment as follows: "A collection of hardware, software, network communications and procedures that work together to provide a discrete type of computer service. There may be one or more environments on a physical platform, e.g. test, production. An environment has unique features and characteristics that dictate how they are administered in similar, yet diverse manners".

Figure 1 illustrates a sample methodology of global software development and delivery from the ITIL perspective adjusted to the needs of globally distributed software project.

The emergence and expansion of GSDD can be attributed to the rapid growth of offshoring activities by the top Indian software providers. At the turn of the century, Indian companies such as Wipro, Infosys and TCS had brought to the market a number of so-called Global Delivery Models (OECD 2006, p. 128).

The Global Delivery Model (GDM) is an implementation of GSDD which brings software delivery to both onsite and globally distributed offshore locations (Buxmann et al. 2008, p. 172). The rapid emergence of this new generation of offshoring models was highlighted by Kurbel (2008, p. 224) and can be explained by the general move of the IT industry from classical offshoring to agile and distributed techniques of software development and by the advances in networking technology. The popularity of the model is stressed by the fact that already in 2006 51% of all offshoring contracts involved global software delivery (TPI 2006, p. 20).

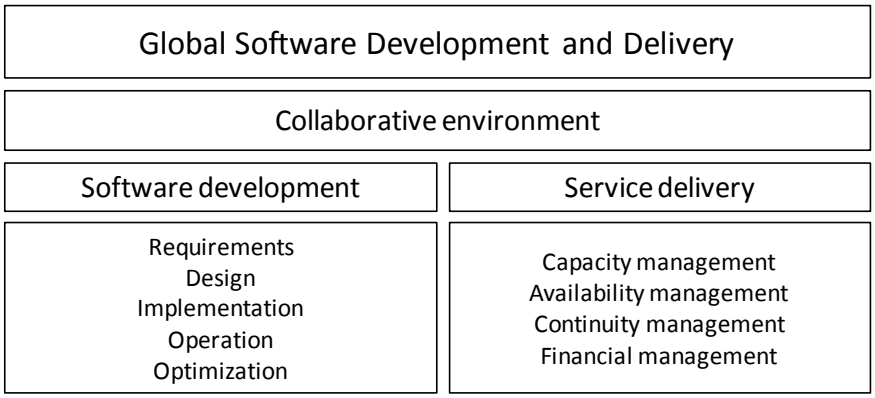

Figure 1: GSDD methodology based on IT infrastructure library

Major benefits of GSDD include reduction of cost (development in low-cost locations in India, China or Eastern Europe), diminishing risks of project failures (distributed development), agile approach to customization, and access to the global pool of talented developers.

Despite these benefits, GSDD is exposed to a number of risks including, among others, complexity of project management in the geographically distributed teams, communication issues, technical challenges, and capacity problems (Hussey and Hall 2007, pp. 79-80). In order to reduce these risks major offshoring providers employ a number of standardized process optimization techniques and reference models including RUP, CMMI, ISO 9001.

This paper proposes and examines a different approach that can support further improvement of GSDD. This new approach is based on web-based development and PaaS technologies which, when mature, may change the offshoring landscape significantly.

3 Platform-as-a-Service within the cloud computing paradigm

3.1 Cloud Computing

In their effort to summarize a first comprehensive definition of clouds in the sense of cloud computing, Vaquero et al. (2009, p. 50) point out the following main characteristics of the new phenomenon: scalability, pay-per-use utility model and virtualization.

In this sense, according to Armbrust et al. (2009, p. 1), from an infrastructure perspective, the cloud computing paradigm provides three distinctive new aspects that set it apart from previous developments in the client/server sphere: (1) the "illusion" of a seemingly indefinitely big resource pool which eliminates the need to plan ahead for possible increases in demand. (2) The lack of need for an up-front commitment to a particular level of service, allowing companies to pay as much for resources as they actually use, which leads to the third point - (3) the granulation of the services - the user pays for the actual amount of resources used.

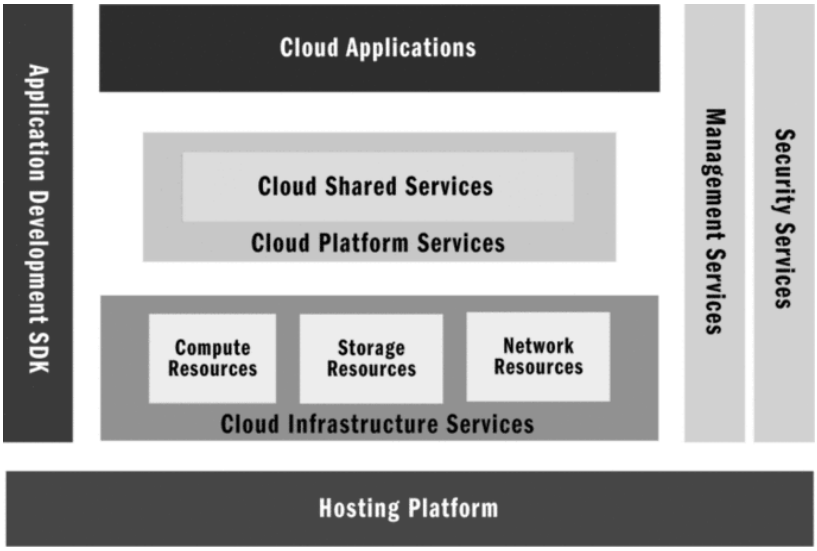

Figure 2: Layers of a typical cloud computing architecture (Joseph 2009)

The term cloud computing is used to describe not only a type of virtual, elastic infrastructure, but also a range of services that are based on an Internet on-demand, pay-as-you-go paradigm. Figure 2 depicts a general view of a typical cloud computing architecture. This paper focuses mainly on the middle layer between the infrastructure (IaaS) and the applications – the platform.

3.2 Platform-as-a-Service

Platform-as-a-Service is positioned between SaaS and IaaS and generally refers to internet-based software delivery platforms on and for which third-party independent software vendors or custom application developers can create multi-tenant, web-based applications that are hosted on the provider's infrastructure and are offered as a service to customers.

The main premise of PaaS is providing software developers and vendors with an integrated environment for development, hosting, delivery, collaboration, and support for their on-demand software applications. Like other software platforms

(Gawer and Cusumano 2008, p. 28), PaaS aims to be a foundation for a broad, interdependent ecosystem of users and businesses. It can support tasks from code editing to deployment, runtime, and management (Lawton 2008, p. 13). The current PaaS ecosystem shows a wide range of different levels of service and is described briefly in section 5. Some platforms offer little more than a set of APIs on top of an elastic infrastructure, while others offer fully functional web-based IDEs and/or 4th generation programming language environments allowing an easy creation of metadata-level (Coffee 2009, p. 26) mash-ups. Additionally, a PaaS could support built-in back-end functionalities of applications like billing, metering, advertising, etc. (Dubey and Wagle 2007, p. 10), which are generally inherent in the "as-a-service" model. Revenue is generated, similar to other software platforms (Evans et al. 2008, p. 3), mainly from the end users. To summarize, a web-based service platform offers a wider range of functions and services than the standard stationary platforms.

Considering this, it can be argued that the PaaS model might be disruptive to the business and strategic practices of ISVs and developers that decide to embrace it and develop their applications for it. The following section discusses the potential implications of PaaS on the evolution of software development and delivery, based on relevant components of the ITIL framework. Considering the global nature of the network-based PaaS and SaaS paradigms, this paper specifically looks into the concepts of distributed work and collaboration within a global software development and delivery framework.

4 PaaS as an enabler of GSDD

4.1 Methodology

This section outlines the compatibility between GSDD and PaaS based on the ITIL V2 framework and materials gathered from the investigation of several case studies describing organization of workflow in a typical distributed offshoring project.

For the ITIL perspective we mainly refer to the ITIL's Service Delivery Book (OGC 2004) and to the ITIL® V2 Glossary v01 (OGC 2006). As a reference for GSDD concept we denote a number of practical examples from Parvathanathan et al. (2007), Cusick and Prasad (2006), and Lawton (2008).

The aim is to integrate GSDD with the PaaS technology described in the previous section in order to outline an on-demand platform that may suit the needs of offshoring development and delivery.

4.2 Software development

According to ITIL (OGC 2004, p. 128), software development involves designing, developing, testing, supporting, and implementing applications. In an offshoring project these functions are normally spread between globally distributed teams and are executed simultaneously from multiple locations. As Cusick and Prasad (2006, p. 27) mention in their GSDD case study, in order to provide a sustained quality of final code, vendors have to heavily rely on the industry specific standards: "Projects that have only partial environments replicated offshore have had significant integration problems once the code was brought back onsite. Teams must evaluate their environments for portability and might need to execute tactical projects to convert code bases before offshoring".

This passage highlights that a PaaS suitable for an offshoring project should possibly rely on standard development tools and languages in order to reduce training time and to provide integration of development environments between distributed teams, thus reducing costs of development (Lawton 2008, p. 15). PaaS also makes possible the integration of development tools in form of IDE-as-a-Service (Gotel et al. 2009, p. 89).

4.3 Service delivery

According to the ITIL glossary (OGC 2006, p. 33), service delivery involves the core IT service management processes that have tactical or strategic focus, namely, service level management, capacity management, IT service continuity management, availability management, and financial management for IT services. Service delivery is also used to mean the delivery of IT services to customers.

Not all of these processes can be transferred to the cloud as some of them (e.g. Service Level Management) refer to the business, not to the technical aspects and thus can hardly be translated to a platform solution. However, certain functions are good candidates to be placed on the PaaS.

Capacity management (CM) is responsible for ensuring that adequate capacity is available at all times to meet the requirements of the business (OGC 2004, p. 11). Lawton (2008, p. 14) notes in his article that PaaS is suitable for distributed development through: "Programmers' ability to use a shared, high-capacity platform that is easy to provision to additional developers to code and test software and also enables the easy expansion of work groups when necessary".

IT service continuity management (ITCM) refers to the process that ensures that required IT technical facilities (including computer systems, networks, applications, and telecommunications) can be recovered within required and agreed business timelines (OGC 2004, p. 163). PaaS architecture should theoretically provide a high level of continuity, especially if PaaS is deployed by a large, trustworthy provider.

In ITIL, *availability management* (AM) is a term that represents the same aspect of a problem as continuity management did, but only from the client's side (OGC

2004, p. 211). An offshoring provider, especially if its main capacities are located far away from the customer, should guarantee the availability of its products. If a vendor chooses a PaaS as a delivery method then this availability is automatically provided by a platform. This can be viewed as one of the major benefits of PaaS architecture for GSDD.

Financial management (FM) refers to the typical business functions and involves IT accounting, charging, and budgeting. PaaS should include some basic support of this delivery discipline through the billing mechanisms based on the actual use of the underlying service. It has also to offer easy management of expenses with the help of web-based administrative boards and basic automating of billing processes.

From this short investigation it is evident that most of the delivery processes can be transferred to a PaaS and successfully deployed in the cloud. This finding confirms that our initial focus on the service delivery component of PaaS was right.

4.4 Collaboration

The role of collaboration is vividly described by the following passage from the IBM Redbook (Parvathanathan et al. 2007, p. 14): "Inadequate collaboration can pose serious challenges to a distributed project, in terms of unexpected rework, mismatched processes, and poor project synchronization and team dynamics."

An adequate PaaS that meets the need of a GSDD project should provide plenty of tools that make seamless real-time interaction between teams possible. These tools may include shared source code development and IDE integration, similar to those used in the IBM's Jazz project (Bartelt et al. 2009), web-based dashboards, project management tools, discussion threads, and automatic tracking systems and may even be integrated with popular social networks.

4.5 Summary

It is evident that each of the three major components of GSDD can be successfully transferred to the cloud-based platform. Table 1 summarizes this section and provides an outlook of a PaaS suitable for an offshoring project that makes use of GSDD.

It also needs to be mentioned that despite its bright perspectives PaaS is exposed to critique ranging from possible PaaS system unavailability and vendor's lock-in within a particular platform to clients' reluctance to keep private data in a cloud. However, these risks can be diminished as the PaaS technology matures.

Table 1: Major characteristics of PaaS-based GSDD

GSDD element	PaaS characteristics
Development	Web-based IDE, common programming languages, industry standards, code repositories, easy/automatic updates
Delivery	
CM	Scalable hosting, easy expansion of teams and computing power
ITCM	Hosting by large trustworthy providers, computing power available on-demand
AM	24/7 delivery of product to the customer, diminishing risks of unavailability due to limited infrastructure on vendor's side
FM	On-demand scalable billing systems, automatic charging, and transparent financial management accessible per web-interface
Collaborative environment	Social networking, shared code development, project management, tracking systems, web dashboards.

Additionally, the introduction of PaaS in the offshoring business may lead to further reduction of costs (through flexible pay-as-you-go billing), deeper integration of teams, new collaborative mechanisms, increased product availability and improved management of the GSDD projects. The next section, however, explores the current state of PaaS solutions and their potential as GSDD enablers.

5 Current PaaS offerings

While SaaS has gained some market traction in real business scenarios, mainly in the area of customer relationship management (CRM), it is still at the early adopters' stage of development. PaaS, being a newer development (Falkner and Weisbecker 2009, p. 148), is even at a more experimental level.

A number of startup companies[1] have sprung up in the past 2 to 3 years that have specialized in offering different platforms as a service. Some of these companies have been short-lived (Coghead has ceased operation and the technology was bought by SAP (Hoover 2009)), while others, like Bungee Labs and Etelos, seem to have also become dormant since the beginning of 2009.

Simultaneously, big industry players like Microsoft, Google, and Salesforce.com, have developed their own cloud-based platform offerings, each with a slightly different approach.

This section of the paper aims at providing a concise overview of the current market of PaaS offerings with reference to the GSDD requirements laid out in the previous section. Table 2 summarizes the findings from a qualitative comparison between five PaaS offerings, which have been selected on the basis of their market impact and specific features.

[1] Coghead, Bungee Labs, Etelos, LongJump, Rollbase, Zoho etc.

Table 2: Support of major GSDD elements by current PaaS solutions

GSDD element	Google AppEngine	Microsoft Azure Platfom	Force.com	Heroku	Bungee Connect
Development	SDK download + local run-time environment	Visual Studio integration, Eclipse SDK	Web-based and SDK download	SDK download	SDK and Web-based IDE
Delivery					
CM	Supported	Supported	Supported	Supported	Supported
ITCM	Supported	Supported	Supported	Supported	Supported
AM	Supported	Supported	Supported	Supported	Supported
FM	Supported	Supported	Supported	Not supported	Billing supported
Collaborative environment	Not a core component of the App Engine	Not a core component of Azure platform	Collaboration tools not focused on development	Yes, but not as part of the platform	Supported

5.1 Development

There are a number of different approaches towards software development for the platform. Some providers (Salesforce, Bungee Labs) have introduced their own version of a fourth-generation language at the metadata level that would simplify the creation of new applications even by inexperienced programmers with limited programming skills. The same providers also support, with different levels of complexity, web-based IDEs for these languages.

Alternatively, the platforms of Google and Microsoft as well as Heroku rely on standard programming languages: Java and Python for the App Engine; .NET, PHP, Python and Java for Azure, and Ruby for Heroku. The development is not web-based but mostly done with the help of downloadable SDKs for standard development platforms like Eclipse or Visual Studio. Normally, designers and testers are then able to run the applications on a custom runtime environment on a local host that simulates the platform.

To summarize, currently there is a lack of viable solutions that support high-end, standard language, web-based development.

5.2 Delivery

Being mainly a delivery mechanism, most of ITIL's service delivery management processes are covered by a typical PaaS, but there are still differences in the approaches of the different providers.

Capacity management, IT service continuity management, as well as the client-facing availability management are supported by all examined PaaS solutions as these aspects of delivery, most prominently capacity management, stand actually at the core of the cloud computing idea. It can still be argued, that small and not well established PaaS providers pose a risk to ITCM and AM, considering the possibility of the providers going out of business and the vendor lock-in.

Concerning financial management, most platforms inherently support these functions as they themselves are based on pay-per-use billing and not on up-front licensing fees.

5.3 Collaboration

Collaboration over the internet has been a driving force of distributed software development, on one hand, and one of the main selling points of end user cloud-based applications such as web-based word editors and other office applications, on the other. Interestingly, not all PaaS solutions offer collaboration tools as a core component of their architecture.

Bungee Connect is rather an exception with its solid support for collaborative development based on its web-based IDE. Heroku supports collaboration by acting as a live repository for development projects which could be universally accessible by authorized development team members. In the case of Microsoft Azure and the Google App Engine, collaboration is not a functionality of the platform itself but is offered through auxiliary tools by the providers.

6 Conclusion

Offshoring and globally distributed work, made possible by the convergence of several technological, organizational, and political factors, have been ubiquitous in recent years. Some of these factors, like high level of connectivity, increased processing power, improved internet technologies, have also enabled the current cloud computing trend. The paper looked into the theoretical and practical suitability of the currently emerging Platforms-as-a-Service model to support and facilitate global software development and delivery practices and standards for internet-based Software-as-a-Service solutions. The examination of particular benefits and risks of such implementations, as well as surveying their reception by users and vendors is not included in the scope of this text and can be pursued in future research efforts.

It can be concluded that the two concepts are definitely compatible, and, a suitably designed platform can theoretically support all required service delivery management processes even at the current technological level. It became evident that service platforms are mainly a software delivery mechanism, which can allow

software developers and vendors to concentrate on the quality of their products by removing the need to deal with problems and uncertainties of the delivery options.

The current PaaS market does not necessarily support the needed functionalities and workflows that would make it viable for a broad-scale global software development and delivery adoption. It should be noted that this area of computing is, while very new, also very dynamic and open for innovation, changes and research.

References

Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A (2009) Above the clouds: a Berkeley view of cloud computing EECS Department. University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28.

Bartelt C, Molter G, Schumann T (2009) A model repository for collaborative modeling with the jazz development platform. In: Hawaii International Conference on System Sciences, Los Alamitos, IEEE Computer Society.

Bon JV (2008) Foundations of IT service management based on ITIL. Van Haren Publishing, Zaltbommel.

Buxmann P, Diefenbach H, Hess T (2008) Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven. Springer, Heidelberg.

Coffee P (2009) Emergence of the platform as a service for enterprise technology. Alignment, Information Management and Consulting, 24(2).

Cusick J, Prasad A (2006) A practical management and engineering approach to offshore collaboration software. IEEE, 23(5): 20-29.

Dubey A, Wagle D (2007) Delivering software as a service. McKinsey Quarterly, 1-12.

Evans DS, Hagiu A, Schmalensee R (2006) Invisible Engines: How Software Platforms Drive Innovation and Transform Industries, MIT Press, Cambridge, MA.

Falkner J, Weisbecker A (2009) Engineering of services and business models for grid applications. In: Grid Economics and Business Models: 6th International Workshop, GECON 2009, Delft. Springer, Heidelberg.

Gawer A, Cusunamo MA (2008) How Companies Become Platform Leaders, MIT Sloan Management Review 49(2): 28-35.

Gotel O, Joseph M, Meyer B (2009) Software engineering approaches for offshore and outsourced development. Springer, Heidelberg.

Hoover JN (2009) SAP buys cloud computing startup Coghead. InformationWeek. http://www.informationweek.com/news/services/saas/showArticlej.html?articleID=214502010. Retrieved September 11, 2009.

Hussey JM, Hall SE (2007) Managing Global Development Risk. CRC Press, Boca Raton.

Joseph J (2009) Patterns for high availability, scalability, and computing power with Windows Azure. MSDN Magazine (May 2009). http://msdn.microsoft.com/en-us/magazine/dd727504aspx#id0430026. Retrieved September 16, 2009.

Kurbel K, Datsenka R (2009) Global Delivery Model. Enzyklopaedie der Wirtschaftsinformatik. http://www.oldenbourg.de:8080/wi-enzyklopaedie/lexikon/uebergreifendes/Globalisierung/Offshoring/Global-Delivery-Model. Retrieved October 2, 2009.

Kurbel KE (2008) The making of information systems. Springer, Heidelberg.

Lawton G (2008) Developing software online with platform-as-a-service technology. Computer 41(6): 13-15.

OECD (2006) OECD information technology outlook 2006. OECD Publishing, Paris.

OGC (2004) Service delivery. The Stationery Office, London.

OGC (2006) ITIL V2 glossary of terms, definitions and acronyms v01. http://www.best-management-practice.com/Glossaries-and-Acronyms. Retrieved September 23, 2009.

Parvathanathan K, Chakrabarti A, Patil PP, Sen S, Sharma N, Johng Y (2007) Global development and delivery in practice: experiences of the IBM Rational India lab. http://www.redbooks.ibm.com/abstracts/sg247424.html. Retrieved September 23, 2009.

TPI (2006) The TPI index: an informed view of the state of the global commercial outsourcing market. http://www.tpi.net/knowledgecenter/tpiindex. Retrieved September 23.

Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2009) A break in the clouds: towards a cloud definition. SIGCOMM Computer Communication Review 39(1): 50-55. doi: 101145/14960911496100.