

Article

# A Generic Software Development Process Refined from Best Practices for Cloud Computing

Soojin Park <sup>1</sup>, Mansoo Hwang <sup>2</sup>, Sangeun Lee <sup>3</sup> and Young B. Park <sup>4,\*</sup>

<sup>1</sup> Graduate School of Management of Technology, Sogang University, #1, Shinsu-dong, Mapo-gu, Seoul 121-742, Korea; E-Mail: psjdream@sogang.ac.kr

<sup>2</sup> School of IT Convergence Engineering, Shinhan University, #95 Hoam-ro, Uijeongbu, Gyeonggi 480-701, Korea; E-Mail: mshwang@shinhan.ac.kr

<sup>3</sup> Software Engineering Center, Nurikkumsquare R&D tower #13, 396 Worldcupbukno, Mapo-gu, Seoul 121-795, Korea; E-Mail: selee@nipa.kr

<sup>4</sup> Department of Computer Science and Engineering, Dankook University, San29, Anseo-dong Choan-si, Chungnam 330-714, Korea; E-Mail: ybpark@dankook.ac.kr

\* Author to whom correspondence should be addressed; E-Mail: ybpark@dankook.ac.kr; Tel.: +82-10-3438-8840; Fax: +82-41-559-7875.

Academic Editor: Young-Sik Jeong

Received: 15 January 2015 / Accepted: 29 April 2015 / Published: 30 April 2015

---

**Abstract:** Cloud computing has emerged as more than just a piece of technology, it is rather a new IT paradigm. The philosophy behind cloud computing shares its view with green computing where computing environments and resources are not as subjects to own but as subjects of sustained use. However, converting currently used IT services to Software as a Service (SaaS) cloud computing environments introduces several new risks. To mitigate such risks, existing software development processes must undergo significant remodeling. This study analyzes actual cases of SaaS cloud computing environment adoption as a way to derive four new best practices for software development and incorporates the identified best practices for currently-in-use processes. Furthermore, this study presents a design for generic software development processes that implement the proposed best practices. The design for the generic process has been applied to reinforce the weak points found in SaaS cloud service development practices used by eight enterprises currently developing or operating actual SaaS cloud computing services. Lastly, this study evaluates the applicability of the proposed *SaaS cloud oriented development process* through analyzing the feedback data collected from actual application to the development of a SaaS cloud service *Astation*.

**Keywords:** SaaS-cloud computing; software development process; process improvement

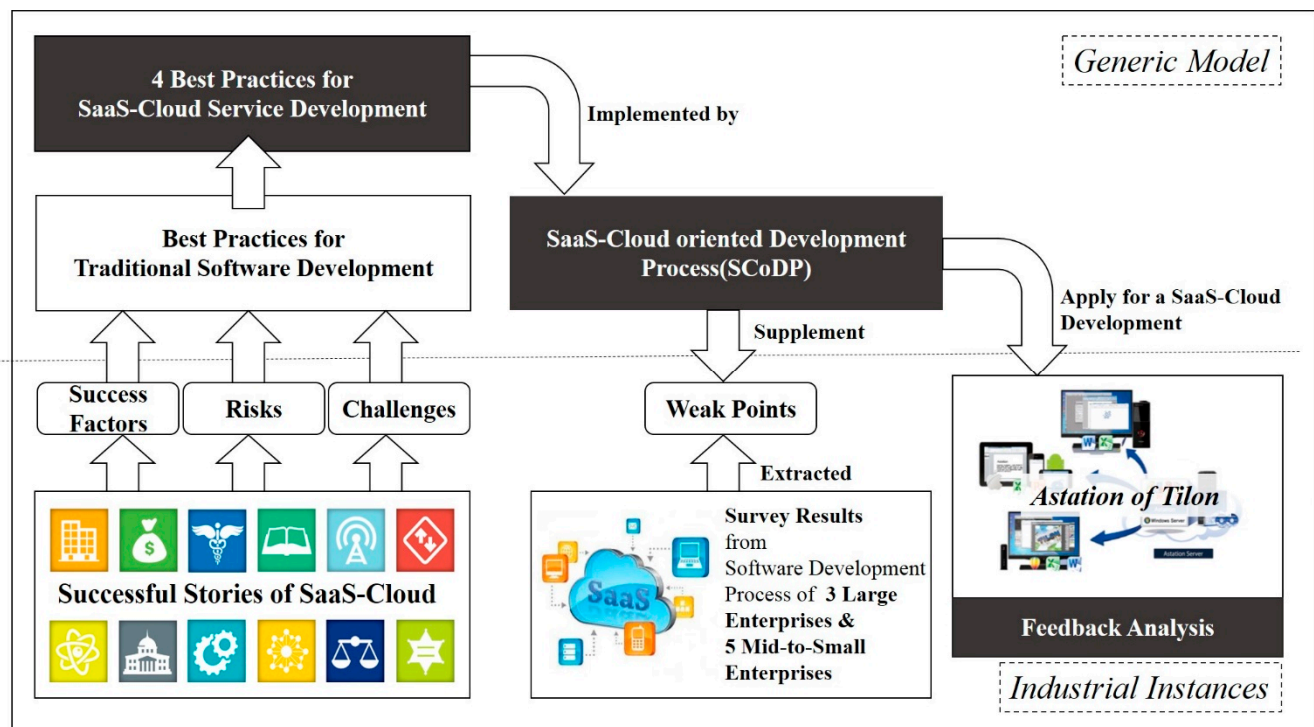
---

## 1. Introduction

One of the goals of cloud computing is to maximize the effectiveness of shared resources [1]. In other words, cloud computing implements green technology by significantly reducing repeated consumption of IT resources. According to [2], “to avoid repeats of Internet bubbles and to maintain business operations, achieving long-term sustainability is an important success factor for organizations”. In this background, three fundamental cloud service models, *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)*, and *Software as a Service (SaaS)*, have been introduced.

SaaS cloud computing in particular has enjoyed recognition as a recent keyword in the software industry with powerful ripple effects influencing how software is produced and consumed [3–8]. Due to the expansion of SaaS cloud computing, software is no longer regarded as a resource to be purchased; instead, it is rented. Much software for consumers is already serviced in various cloud computing environments [9,10]. Social network service (SNS) usage, for example, has become more and more widespread while the stratum of machine to machine (M2M) networks grows exponentially due to increased smartphone use. Such trends inflated the volume of data to be processed by enterprises, causing the data to take on the aspects of big data. According to Cisco [11], Cloud applications will account for 90% of total mobile data traffic by 2018, compared to 82% by the end of 2013. Mobile cloud traffic is expected to grow twelvefold from 2013 to 2018, attaining a compound annual growth rate of 64%.

In keeping with this trend, software development environments have changed since the 90s from client server environments to web-based environments. Now SaaS cloud computing is recognized as a representative paradigm of the next software development and sharing environments. Such a paradigm shift in the IT environment requires a change in software development methods as well. To contribute to this transition, we propose a new software development process which incorporates the best practices and provides guidelines for efficient software development in the new software consumption environment. The steps in the best practice extraction, software design, and evaluation are as shown in Figure 1. First, we have analyzed existing successful cases in making the transition to cloud computing environments in order to derive their success factors and to highlight risks and challenges they have met. Numerous challenge factors exist in diverse aspects of qualities provided by cloud computing services [12]. Based on the result of the analysis, we have defined four new best practices below that are proven to be useful in the scenes of software development conducted in cloud computing-oriented paradigms. To actualize the newly defined best practices, application guidelines were documented and formulated into a single process called *SaaS Cloud Oriented Development Process (SCoDP)*. Prior to designing SCoDP, we conducted a survey involving eight enterprises currently developing or operating SaaS cloud services to understand their current processes. The goal of the survey was to verify if the weak points in the surveyed organizations’ software development processes could be improved.



**Figure 1.** Framework for designing the SaaS cloud oriented development process (SCoDP).

Fortunately, the weak points were identified as remediable through realization of the proposed best practices. Based on survey results and analysis, we have designed a process which implements the best practices in the actual software development environment. Then we applied the resulting process to an actual case of SaaS cloud development project so as to verify the effectiveness of the proposed approach. The target project for the process application test was *Astation* [13], a cloud service which enables users to use Windows-based applications on various devices regardless of the native operating system. The results from a seven month-long application of SCoDP to the developers working on *Astation* have concluded with a very positive reception.

The remainder of this paper is as follows: Section 2 reviews the results from a survey of successful SaaS cloud computing adoption stories. Section 3 describes the challenges in software development to support SaaS cloud computing-based IT environments. Section 4 proposes new best practices for software development to support SaaS cloud computing environments. Section 5 discusses the SCoDP design process and its structure of implementing the proposed best practices. Section 6 shows the feedback results from a real-life application of SCoDP to a service development project. Finally, Section 7 summarizes this paper and shares future research directions.

## 2. Survey of Success Stories

The first step of our research was to collect as many noteworthy cases of SaaS cloud computing adoptions as possible. A total of nine cases were collected as a result, which we analyzed to derive the success factors effecting IT environments being converted to SaaS cloud computing environments. Table 1 summarizes the result of SaaS cloud computing adoption from the case studies. The case studies are divided into two categories; private enterprises and government-level system adoptions.

**Table 1.** The list of reviewed success cases.

<b>Industry</b>	
Salesforce.com	<ul style="list-style-type: none"> <li>▪ Provides web-based CRM (customer relationship management) solution to different enterprises</li> <li>▪ Sells various business applications through AppExchange</li> </ul>
Google Chrome OS	<ul style="list-style-type: none"> <li>▪ Provides web-based cloud OS</li> <li>▪ Every computer work is processed in the main server of Google through web browser</li> </ul>
Microsoft Office 365	<ul style="list-style-type: none"> <li>▪ Provides in-house communication system such as e-mail, integrated communication and collaboration package online</li> <li>▪ Provides Office Professional Plus, Lync Online, SharePoint Online, and Exchange Online</li> </ul>
IBM Lotus Live	<ul style="list-style-type: none"> <li>▪ Provides collaboration solution for companies such as web-mail, meeting, connection, and LotusLive Engage</li> <li>▪ Provides solution with IBM's technical know-hows in IT service industry</li> </ul>
Apple iCloud	<ul style="list-style-type: none"> <li>▪ Allocates 5GB of storage capacity , and provides music, application program, book, and picture streaming services</li> <li>▪ Sells different applications on Apple AppStore</li> </ul>
<b>Government</b>	
USA [14]	<ul style="list-style-type: none"> <li>▪ Applies cloud computing with focus on public services</li> <li>▪ Announces the report of federal government's security policy and cloud service usage action plan in December 2010</li> <li>▪ Adopted "Cloud First" policy which promotes cloud service importation as its the first priority</li> <li>▪ Emphasizes the effectiveness of public institutional assets, which is followed by improved quality of the service they provide</li> </ul>
China [15]	<ul style="list-style-type: none"> <li>▪ Built the Wuxi SW development complex to provide computing resources through applying Cloud computing technology in 2008</li> <li>▪ Operates IDC to benchmark foreign advanced technologies—eleven SW development complexes are to be built all over China in the future</li> <li>▪ Reduces development time and service operation cost for resident enterprises while improving their quality</li> </ul>
England [16]	<ul style="list-style-type: none"> <li>▪ Plans to construct cloud service on the pan-government level as part of its innovative strategy for public sector data centers</li> <li>▪ Expects to save time and cost in constructing and operating work services at public institutions</li> <li>▪ Plans to create various added value by making efficient use of data that used to be monopolized by the government</li> </ul>
Japan [17]	<ul style="list-style-type: none"> <li>▪ Kasumigaseki Cloud</li> <li>▪ Will converts government IT systems to a single cloud infra by 2015</li> <li>▪ Plans to integrate servers used by 13 central office groups and to build three data centers for importing local government clouds</li> <li>▪ Expects to maximize the efficiency of government administration systems by linking data through cloud and remodeling the process</li> </ul>

After collecting as much data as we could from the successful cases of SaaS cloud computing adaptation at enterprises and governmental public sites, including publicized reports, we analyzed what new services were newly introduced through adopting SaaS cloud computing and what factors

contributed to the successful introduction of the new paradigm. The following is a detailed analysis of the cases from Table 1, with a focus on enterprise cases.

- **Salesforce. Com: CRM (Customer Relationship Management) Cloudization**

Salesforce.com is the first enterprise to actively utilize a differentiated business model of providing on-demand software, even when the concept of SaaS cloud was still new in the market. Through their early adaptation, Salesforce.com was able to significantly increase their profit earned from a single software, and thereby built a sizable customer base composed of corporate and end-user clients. Today, Salesforce.com is a leader in the SaaS cloud sector, recently expanding its business to PaaS services by making available its entire CRM platform. Salesforce.com continues to diversify and differentiate its business model.

Salesforce.com ensures 99.9% availability, in conjunction with reliability and performance under 300 milliseconds processing and response time. Further setting itself apart from other SaaS cloud providers, Salesforce.com ensures equal performance, scalability, and stability for all clients, regardless of their business sizes; equal service whether it is for a single user or for 10,000 users. Lastly, Salesforce.com provides a customization feature for its clients, allowing them to construct or re-configure their solutions through configuring metadata.

- **Google Chrome Operating System (OS): OS as Cloud Service**

Google Chrome OS is a web-based OS provided by Google, capable of managing a computer's hardware resources and its peripherals while enabling the supported applications to run. All work executed on the Chrome-running computer is passed on through a web browser to Google's central server to be processed and have its results returned. As all executions are run through web technologies, the user only needs basic hardware with minimal functions and Chrome OS to be able to run basic office applications and Google-provided services.

Google Chrome OS maximizes the utilization efficiency of network, CPU, memory, and hard disk resources to near 100%, minimizing idle resources. In addition, the OS is capable of distributed data processing through numerous servers across 36 nations by leveraging technologies such as BigTable, MapReduce, Chubby, and Hadoop. This feature has enabled Google to compete with Microsoft entrenched in the dominant position in the PC sector, and also in the business to business (B2B) market. The weakness in security and privacy is being addressed through controlling access to the OS core code base through plug-in sandboxing.

- **Microsoft Office 365: Cloudization of business application**

Office 365 is an online solution service providing high-quality intra-corporate communication systems, including email, integrated communication, office portal, and collaboration packages. The service comprises "Office Professional Plus" which provides document processing and editing functionalities directly through a web browser, "Lync Online", providing instant messaging and online conferencing, "SharePoint Online", enabling corporate portal and internal collaboration tools, and "Exchange Online", providing email and schedule management features. Recent releases introduced customizations for specific client needs (Office 365 for governmental or educational organizations), and supports compatibility with custom security policies or custom-purpose devices. Office 365 provides

multiple service packages with varied features for various numbers of users or functional requirements at the customers' site, coupled with a flexible payment option program.

With significant market share held in the global PC OS and business software market, Microsoft had two success factors in making its transition towards a business model providing cloud services to enterprise customers. First, Microsoft had a large user base of preexisting customers. In 2011, when Microsoft first released Office 365, Microsoft Office had over 90% market share in the business software market (according to a Gartner research). This translates to a global 90% share of the potential customer pool. When an enterprise attempts to convert its internal office software to a cloud-based service, it has to face the issue of educating its users to familiarize them with the new system, and to convert existing documents and files to make them compatible with the new system. Enterprises that previously used Microsoft Office did not face these hurdles when they chose Office 365. In this regard, Microsoft held a significant advantage over other cloud service providers regarding user base expansion by converting its very large pool of existing users. Second, the quality of software provided by Microsoft was high. Microsoft Office had already been quality-tested by its numerous users in terms of functionality and stability, and Office 365 enjoyed a recognition of likely high-quality software from existing Microsoft users.

- IBM Lotus Live: A Cloudization of Various Business Solutions

LotusLive is an online service providing an array of collaboration solutions for enterprises, including a webmail service ("iNotes"), online meeting and connection service, and LotusLive Engage. In particular, LotusLive is a collaboration software with powerful security features that help employees within an enterprise connect and collaborate to achieve better business results. Supported by IBM-operated cloud environments, customers can reach and collaborate with meaningful clients and partners without concern for their firewall management, thereby creating a powerful collaborative network. LotusLive is the foundation for IBM's web-based collaboration service solution, and takes into account factors beyond the software product—such as the context of the client's organizational situation—to provide an integrated solution. IBM also supports private cloud computing to enable client enterprises to construct cloud computing environments tailored to their individual needs and provides IBM softwares as public cloud services.

A highly successful technology service and consulting company, IBM has accumulated a strong understanding of its client needs in the IT service domain and maintains professional services catering to those needs. IBM's solutions and software provided through its cloud environments are based on its understanding and insights, resulting in a reliable and high quality private cloud service differentiated from other public services in the B2B market.

- Apple iCloud: Personal Cloud

iCloud is a cloud computing service provided by Apple, composed of music, photo, application, document, book, and contact services in addition to Apple's email and calendar services. Each iCloud account is allocated to a certain cloud storage space, and downloaded contents such as music, app, book, or photo stream service are not counted against the allocated storage. All purchased applications are automatically accessible from the purchasing user's other registered devices; when a user registers a new device, previously purchased contents can also be downloaded to the new device. Apple not only provides music, movie, or applications but also manufactures and sells hardware to use such contents.

Apple has a diversified business model incorporating various revenue streams, such as iPhone or iPod sales, storage usage fees from iCloud users, and sales from its App Store.

In functionality, iCloud does not differ from existing services provided by other competitors. iCloud's calendar, email, photo, or online storage are largely similar to services from Google, Dropbox, or Skydrive. However, Apple's iCloud is significant in that it led the transition from the enterprise-centric clouds by Microsoft, Amazon, Google, or IBM to personal clouds. While numerous competitors contemplated enterprise users, enterprise environments, and the customers of those enterprises, Apple shifted its focus to Apple customers and created a service for its customers only. This strategy resulted in powerful, positive user reception regardless of the lack of innovative functional improvements. Another success factor in Apple's iCloud was in innovative hardware for cloud services. Apple's clouds are closed services constrained to its own hardware, and service expansion mandates hardware sales. Apple was successful in increasing sales of its iPhone and iPod hardware through providing differentiated user interfaces. Furthermore, Apple leveraged its cloud services to continuously lock-in users to its hardware products, acquiring new users while retaining existing users to expand its powerful user base.

Through analyzing above success cases, the following list of success factors (SF) in the successful stabilization of SaaS cloud computing environment adoption was derived:

### **SF1. Providing differentiated business model and professional service**

Salesforce.com built a powerful base of enterprise and commercial users by becoming the first provider of a differentiated business model that provides software as an on-demand service, establishing itself as a leader in the SaaS cloud sector. IBM's LotusLive was based on a firm understanding of various user needs in the IT service industry, followed by professional services catering to those needs by providing reliable, high quality private cloud service that ultimately differentiated the company from other public cloud service providers in the enterprise B2B market. In this light, differentiated business model design combined with professional service development are preconditions for success in the competition with other service providers.

### **SF2. Ensuring high-reliability and high-quality of service**

As shown in the Salesforce.com's CRM case, ensuring equal performance, scalability, and stability regardless of user base size can lead to acquisition of enterprise and commercial users. In the case of Google Apps, its 99.9% service availability insurance led to positive user recognition about cloud service reliability. In this regard, high service reliability and performance are not nice-to-have add-ons but must-have requirements.

### **SF3. Outstanding price policy**

As in the case of Google Chrome OS, a software cost reduction from conversion to SaaS environment becomes a strong competency. Where enterprises previously spent thousands of dollars annually to purchase and maintain software assets, new cloud service environments greatly reduced that cost.

Therefore, it can be assumed that, to successfully compete with competitors in the previous market, a pricing policy enabling cost lower than purchasing and maintaining software solutions must be considered.

**SF4. Establishing a broad strategy for insurance of potential customers**

As seen in the case of Microsoft's Office 365, acquiring potential customers from the previous market is very important. Microsoft Office's over 90% market share in the business software market implied that 90% of the market are potential customers; this is because an enterprise must consider compatibility with its previous systems and its users educational overhead. In this regard, converting a service from an existing market to a cloud environment necessitates an in-depth analysis of the previous market and establishing of user acquisition strategy for potential users.

**SF5. Prompt transition to the personal cloud environment**

Apple's iCloud case shows that the personal cloud must be considered as important as an enterprise-centric cloud. Where many competitors focused more on enterprise users, environments and customers, Apple preoccupied the personal cloud market by adding differentiation through personalization even though utilizing similar technologies and environments as its competitors. Apple's success indicates that the previously untapped personal user market is becoming increasingly larger, and therefore a faster transition into personal cloud environments through easy accessibility and low pricing policy is required.

**SF6. Providing innovative hardware using the service**

Apple's iCloud case presents another success factor, which is the service-optimized hardware. Apple has shown success through building a closed cloud service environment composed of its cloud services and hardware dedicated to its cloud services. The hardware sales were successful through providing differentiated user interface which facilitated users to use Apple's services in more optimized forms, and, in turn, led users more easily to try other Apple services. This case shows that maximizing customer satisfaction by providing an optimized experience is crucial to a cloud service's success. Improved customer satisfaction will result in higher retention and less user churn, leading to a persistent and loyal user base in the turbulent cloud environment.

**SF7. Building organic cooperative relationships between related institutions and data sources**

As seen through the cases of cloudization of governmental public services in the US, UK, Japan, and China, the impact of data sharing and interconnected analysis across multiple organizations may deliver significantly increased results compared to an individual system utilization. For instance, oil price forecasting is influenced by several factors such as the relationships between nations, growth rate of emerging economies, or financial crises in other regions. Having access to relevant information enables scientific analysis of the factors and improves accuracy of oil price forecasting. Such government-led cloudization success cases indicate that sharing and incorporating as much correlated data as possible into data analysis is important in improving analysis accuracy. To realize this need, a strategy for creating and maintaining collaborative relationships with relevant organizations is necessary.

**SF8. Constantly expanding constructed bulk information and ensuring accumulation management technology**

Other major factors for success in government-led cloudization of public services include the amount of accessible data and the procurement of technology to manage and analyze the resulting data.



Collaborative data analysis must go beyond storing the bulk information resulting from large-scale system integrations between organizations. It should also introduce effective management of the large data and secure mechanisms to analyze valuable meanings of the accumulated data. Successful collaborative data analysis is likely to lead to further applications of the data, resulting in even more services.

For practicality in analyzing the success cases, ROI (return on investment) was also listed. ROI can be a factor of determining when an organization considers the transition to the SaaS cloud computing environment. The following are expected benefits and risks of introducing SaaS cloud computing:

Expected **benefits** of the transition to SaaS cloud computing environment:

- Increased service speed due to increased agility, adaptability, and flexibility
- Reduced service cost
- Retainable bulk storage without additional facility investment

**Risks** of the transition to SaaS cloud computing environment:

- Vulnerable security and privacy
- Enterprise support and service maturity
- Return on investment concerns
- Possibility of performance decline

### 3. Challenges of the Software Engineering Process

As mentioned above, the transition of the IT environment to SaaS cloud computing is viewed as a paradigm shift rather than a minor technology change. Accordingly, the transition requires changes in many aspects of software development and operation. In this section, we analyze changes due to introducing SaaS cloud computing. The following are descriptions of each element of change:

- **Challenges in service management:** In order to provide resilient and flexible service, a representative benefit of cloud-based systems, new services must be created through compounding applications in the cloud. To institutionalize new service creation through compounding applications, a catalog must be made and controlled to constantly manage the list of services in the cloud environment in the current organization. In addition, the service catalog must use business terms, not IT terms. This is because the need for new cloud services is created by the users engaged in general business, not the IT industry.
- **Challenges in service level management:** Cloud computing-based services must provide different services depending on the price their users pay. In a cloud environment, a solution must be defined, differentiated by service levels, and be applicable to all processes such as strategy generation, supplier management, service portfolio management, and service catalog management.
- **Challenges in incident and problem management:** An advanced form of virtualization significantly affects the impact analysis of incidents or errors. Virtualization results in changes in traditional 1st-2nd-3rd support organization structure. The biggest change due to the adoption of SaaS cloud computing is the change of organization and management structures.
- **Challenges in change management & release/deployment management:** Due to the dynamics of cloud computing, risk and impact analysis becomes more complicated. Therefore it requires constructing mature change management and automatized release & deployment process.

- **Challenges in capacity management:** The dynamics of cloud computing may complicate the process of predicting the total service capacity and provisioning it. The prediction model for capacity calculation needs to be changed to reflect cloud computing properties.
- **Challenges in IT financial management:** Due to the flexible pricing policy of cloud computing, more strategic decision makings are necessary when pricing the organization's IT services. In cloud computing, flexibility itself becomes a price. The greater the flexibility, the higher the price.
- **Challenges in supplier management:** Whether the IT organization is combined with IT service inside and out or not, it is highly necessary to integrate different IT services according to defined SLA (Service Level Agreement). Therefore, the policy to manage many suppliers for a single service includes insuring potential vendors and meeting the challenge of maintaining the balance between them. In addition, the standard with which to select a supplier becomes much more important.
- **Challenges in supporting common operations:** As provisioning needs to be performed much quicker than before, the software service development process must be clearly defined and every process must be approved in advance. Tools for development must be able to support high levels of automatized environment.
- **Challenges in service portfolio management:** Service portfolio management process is a new mechanism to demonstrate new business opportunities proposed by cloud computing. The presence of service portfolio is relevant with every risk. It means the service system must be able to react to every possible situation.
- **Challenges in service validation and testing:** A service validation and testing system that can support dynamics of cloud computing becomes a requirement. However, prompt provisioning and flexible scale-up have made things easier. For a certain time, a testing environment must be constructed to prepare when services with and without Cloud computing are provided simultaneously.

#### 4. Best Practices for SaaS Cloud Environment

In this section, we present the best practices for software development in the SaaS-Cloud environment extracted from the survey and from the analysis of risks and challenges. The best practices are presented in Table 2. Before the explanation of each best practice, the relevant risks, success factors, and challenges introduced in the previous section are denoted as rationales for its inclusion of the best practice. A set of prerequisite solutions to implement each best practice in development of SaaS cloud services is also defined.

**Table 2.** Summary of best practices for Software as a Service (SaaS) cloud environment.

<b><i>Best Practice 1: Incrementally transform the application to the cloud computing environment</i></b>	
Risks mitigated	-Enterprise support and service maturity -Return on investment concerns
Success factors	-Building up an organic cooperative relationship between related institutions and data sources -Constantly expanding constructed bulk information and ensuring accumulation management technology
Challenges solved	-Challenges in incident and problem management -Challenges in capacity management -Challenges in change management & release/deployment management

Table 2. Cont.

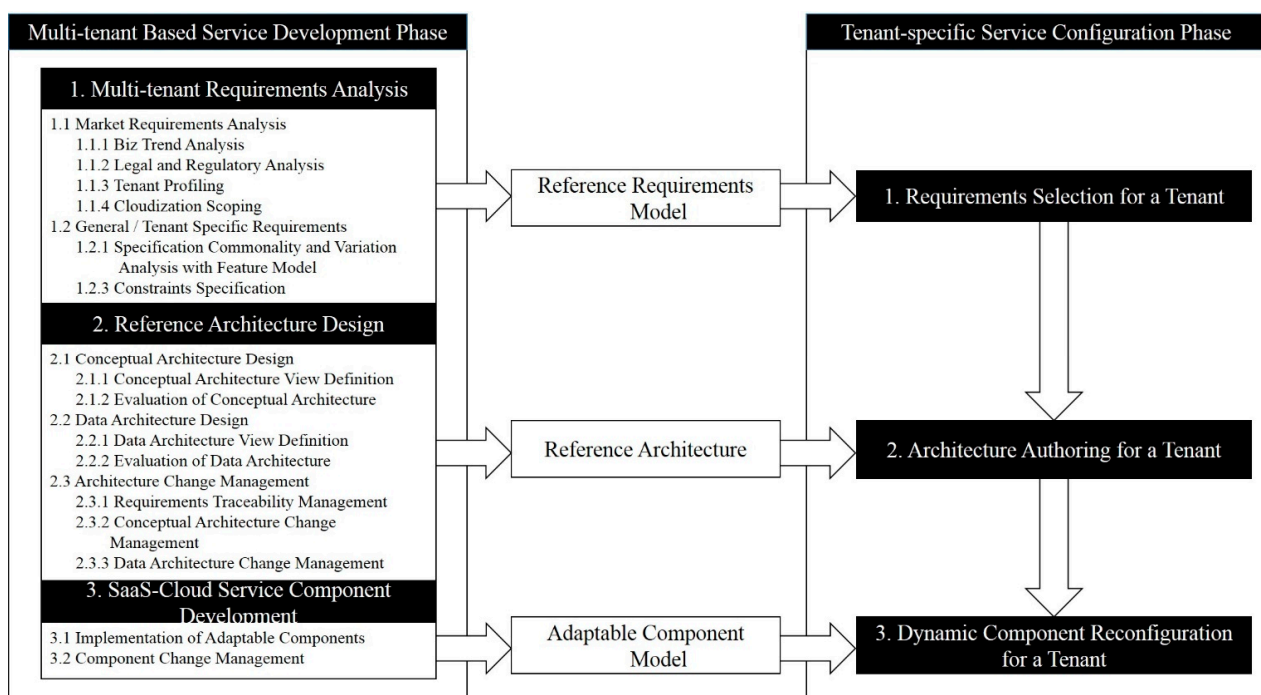
<b><i>Best Practice 1: Incrementally transform the application to the cloud computing environment</i></b>	
Explanation	
<p>Transiting existing applications to cloud environments obviously requires high initial costs. Therefore, one must consider the gain by investment before transit, while also contemplating the maturity of the service to be converted and level of the enterprise. Even in the case where the decision for cloudization has already been made, a plan must be established through analysis for gradual cloudization to minimize the risk and verify potential incidents and issue from multiple viewpoints.</p> <p>Through repeated small-sized cloudization, an organization can gradually secure necessary capacity. Such gradual cloudization is not limited to applications operated by a single organization. The same strategy is applicable to cloudization of systems in organic cooperative relationship between big data sources.</p>	
Prerequisite solutions	<ul style="list-style-type: none"> <li>-Re-engineering methods for legacy applications</li> <li>-Cloudization method for application</li> <li>-Supporting tools, and cloudization planning technique.</li> </ul>
<b><i>Best Practice 2: Respond resiliently to extended changes</i></b>	
Risks mitigated	<ul style="list-style-type: none"> <li>-Enterprise support and service maturity</li> <li>-Return on investment concerns</li> </ul>
Success factors	-Prompt transition toward the personal Cloud environment and innovative hardware using the service
Challenges solved	<ul style="list-style-type: none"> <li>-Challenges in change management &amp; release/deployment management</li> <li>-Challenges in service portfolio management</li> </ul>
Explanation	
<p>Cloud service is related to quite extensive stakeholders and concerned systems more closely than single application-based service. There are frequent requests for software changes from stakeholders, many physical systems, and changes in virtual devices. Such issues are some of the quality factors required of cloud service to prove how fast cloud service can accept those external change elements. Detailed planning needs to be done from the stage of designing the service to flexibly react to multilateral external change elements that are difficult to predict. Based on the predictive value calculated in regards to the system capacity, data needs to be distributed properly, and a guideline must be given to indicate which process must be reflected in the case of software change requests. In cloud service environments, the main element leading change management becomes the service itself. Apart from the change management, strategies and automatized support are necessary to release and distribute changed software in the right place, at the right moment.</p> <p>In addition, cloud addition and deletion requests may occur for certain services in the business dimension. Such a new request becomes the element to require top-down portfolio management for the IT service. While IT engineers used to design IT service in technology dimensions, IT service provided in the cloud environment must estimate exact cost and predictive value to decide on cloudization based on the precondition that cloud service is a business model.</p>	
Prerequisite solutions	<ul style="list-style-type: none"> <li>-Domain engineering technology</li> <li>-Service-oriented software requirements traceability management</li> <li>-Service portfolio management process which integrates legacy application and cloud life cycle</li> </ul>
<b><i>Best Practice 3: Consider multi-tenants for an application</i></b>	
Risks mitigated	<ul style="list-style-type: none"> <li>-Vulnerable security and privacy</li> <li>-Enterprise support and service maturity</li> <li>-Possibility of performance decline</li> </ul>
Success factors	<ul style="list-style-type: none"> <li>-Providing differentiated business model and professional service</li> <li>-Outstanding price policy</li> <li>-Ensuring high-reliability and high-quality of service</li> <li>-Establishing a broad strategy for insurance of potential customer</li> </ul>

Table 2. Cont.

<b>Best Practice 3: Consider multi-tenants for an application</b>	
Challenges solved	<ul style="list-style-type: none"> <li>-Challenges in service management</li> <li>-Challenges in service level management</li> <li>-Challenges in service portfolio management</li> <li>-Challenges in service validation and testing</li> <li>-Challenges in supporting common operations</li> </ul>
Explanation	
<p>One of the major changes in cloud environment is that many devices using virtualization of original physical infrastructure technologies (VM, vDisk, vSwitch, vDesktop, etc.) compose another infra environment. Due to such challenges, composed services must correspond with various tenants, according to the contract terms even when they provide the same application functions. Though such requirements about multi-tenants enforce guaranteed independence levels, and performance is possible if the cloud service provider was subordinated to only a single customer, such assumptions are not practical. The following are the best subordinate practices which precede the best practices:</p> <ul style="list-style-type: none"> <li>- Based on the consistent policy of the organization, cloud resources shall be managed dynamically.</li> <li>- Based on the automatized support, time taken to provisioning shall be reduced (from months to minutes, in a radical case).</li> <li>- End-to-end monitoring shall be supported in the business process of cloud service from the network and system management to application performance management and single view offers for cloud service.</li> </ul> <p>Through the co-relative analysis between business service performance and infra service performance, general cloud performance plan shall be established.</p>	
Prerequisite solutions	<ul style="list-style-type: none"> <li>-SLA management per business service tenant</li> <li>-Sharing service standardization method for operational cost saving</li> </ul>
<b>Best Practice 4: Manage vulnerability of software</b>	
Risks mitigated	<ul style="list-style-type: none"> <li>-Vulnerable security and privacy</li> <li>-Possibility of performance decline</li> </ul>
Success factors	<ul style="list-style-type: none"> <li>-Ensuring high-reliability and high-quality of service</li> <li>-Establishing a broad strategy for insurance of potential customer</li> </ul>
Challenges solved	<ul style="list-style-type: none"> <li>-Challenges in incident and problem management</li> <li>-Challenges in change management &amp; release/deployment management</li> <li>-Challenges in service validation and testing</li> </ul>
Explanation	
<p>The wider the range of cloud service, the more likely that related software becomes vulnerable. Apart from security issues such as securing countless pieces of personal information scattered in the cloud or possible privacy violations, countermeasure solutions must be prepared to control software vulnerability. An issue created by one tenant can escalate to every tenant if they share the same service. Overcoming risks related to vulnerabilities and providing more reliable, higher quality service than the competitors can be the biggest success factor. However, software vulnerability has not been significantly handled in best practices for traditional software development yet, so the technology is not mature enough to deal with the weaknesses in cloud service. Vulnerability control of cloud service requires corresponding countermeasures for the predicted vulnerability issues when the range of service users is not limited. However, a strategy must be also prepared to minimize performance degradation due to unpredictable vulnerabilities.</p>	
Prerequisite solutions	<ul style="list-style-type: none"> <li>-Software vulnerability analysis methods</li> <li>-Software safety guarantee guideline per levels of the software lifecycle</li> <li>-Verification technology for software architecture in various operational environments</li> </ul>

## 5. A Generic Process for Implementing Best Practices : SaaS Cloud Oriented Development Process (SCoDP)

Applying the best practices proposed in Section 4 to actual development organizations requires a well-designed process. Thus, we proceed to design a process framework which could be utilized as a new service development guideline in SaaS cloud environments. We refer to it as *SaaS Cloud Oriented Development Process (SCoDP)*. The composition of SCoDP is described in Figure 2. SCoDP involves two main strategies. The first strategy is to develop a reference architecture that can be commonly referred to regarding the same domain including multi-tenant requirements. This is to eliminate unnecessary development overlap through re-utilization when developing a service in the same domain. It shares the same cause of cost reduction which is the original objective of cloud computing, avoiding unnecessary development overlap by reusing overlapping resources. As reference architecture consists of components developed based on precise market analysis and tenant-specific variation analysis, it becomes the foundation which facilitates real-time provisioning for certain tenants when sufficient automation tools are supplied.



**Figure 2.** SaaS cloud oriented development process (SCoDP) framework.

The second strategy is to design SCoDP where two main development cycles coexist: a multi-tenant-based service development phase which develops a reference architecture and a tenant-specific service configuration phase which composes the service for certain tenant. Among the reference requirements model, which is the output of multi-tenant based service development phase, a requirement set for a certain tenant can be selected. When a tenant-specific requirement set is selected, software artifacts related to the selected requirements can be designated correctly through the traceability link of defined requirements trace model. Executable components of the selected artifacts can be configured as a cloud service in real-time by automated tools. Likewise, an immediate response to selecting and changing requirements per tenant can improve the general sustainability of the system.

Before identifying the activities to be included in SCoDP, we have surveyed the weak points in currently used processes for SaaS cloud services development in real industrial projects. Then, we identified current weak points of SaaS cloud service companies in terms of process maturity. The activities composing SCoDP are selected to make up for the weaknesses and to realize the proposed best practices in the actual field. The following explains how we have designed SCoDP from the survey results.

### 5.1. Industrial Survey for Identification of Weak Points for SaaS-Cloud Service Development

To ensure a new software development process soft-land on a new organization, the target organization's characteristics and maturity must be measured in advance. In accordance with this need, a survey on the organizational maturity of software development capability has been conducted on eight enterprises currently developing and operating SaaS cloud services, in preparation for designing a generic process for efficient SaaS cloud service development. The eight target enterprises have been classified as large (A, B, and C) or mid-to-small (D, E, F, G, and H). Their respective SaaS cloud development profiles are shown in Table 3. Regardless of the company size, organizations actually involved in SaaS cloud service development ranged from under ten members to under a hundred members in size and were primarily dedicated to SaaS operation platforms or cloud-based web services.

We defined the SaaS cloud service development process as three phases: service requirements phase, service construction phase, and service operation and management phase. Survey question sets aimed at checking an organization's development process maturity have been prepared and have been incorporated into the organization in Table 3. Responses were defined as follows: if the activity in question is in practice, then the answer is 1, otherwise the answer is 0. The shared items in Table 4–6 indicate activities that more than half of the questioned enterprises have responded to as currently not engaged in such activity. Such activities have been identified as the weak points of SaaS cloud service development processes, as they represent elements of process not properly carried out.

**Table 3.** SaaS cloud development organization profiles.

Enterprise	Total Staff	SaaS Organization Size	Development Profile
A	3000	6	Open source software-based next generation cloud computing technology, cloud DaaS/storage software technology
B	500	20	SaaS operation platform development in partnership with Microsoft (ERP, CRM, security, M2M, <i>etc.</i> )
C	10,000	30	Consulting and solution implementation for CRM sector in partnership with Salesforce.com
D	15	7	Implementation of cloud storage service providing various client tools and web services
E	10	5	Cloud media platform and cloud-streamed broadcasts for cloud vision and multimedia display equipments
F	16	2	Email link and mass messaging (email, text, <i>etc.</i> ) service
G	190	10	An integrated app store providing content through multi POCs (PC, mobile, TV, <i>etc.</i> )
H	47	23	Cloud application

The weak points identified through analysis of survey results of SaaS cloud service development processes are as follows:

- SaaS-Cloud service requirements phase

Among SaaS cloud service requirement elicitation activities, general software development activities such as business/functional/interface requirements extraction were being performed to a satisfactory level. However, analysis on the common and variable service elements to be provided to a variety of users was found to be less sufficient. In particular, mid-to-small enterprises appear to have more difficulty in identifying common user requirements due to unsatisfactory levels of stakeholder identification and management. While redundancy or error monitoring activities for extracted requirements were sufficiently maintained in all organizations, actions to handle identified inconsistencies or to analyze risks were insufficient. Further, mid-to-small enterprises showed need for improvement in utilizing requirements-based test cases, and creation and management of requirements traceability tables.

Regardless of size, nearly all organizations were unsatisfactory in requirements for management activities, such as change impact analysis or requirements for change identification and control. Mid-to-small enterprises were weak in managing the traceability between requirements and deliverables from each development phase, or in checking for consistency between requirements and deliverables. Evaluation of service costs per requirements was not executed in all organizations.

**Table 4.** Survey results from process maturity measurement at SaaS cloud requirements phase.

List of activities to elicit client requirements	Large Enterprises	Mid-to-Small Enterprises	Total
Define business requirements	3	4	7
Identify and manage stakeholders	3	3	6
Elicit user requirements from stakeholders	2	1	3
Identify common and individual user requirements	2	4	6
Elicit functional requirements	3	5	8
Elicit quality requirements	2	4	6
Elicit data requirements	1	4	5
Elicit interface requirements	3	5	8
Elicit operational requirements	3	3	6
Elicit constraints	1	3	4
List of activities to analyze client requirements	Large Enterprises	Mid-to-Small Enterprises	Total
Identify redundancies and errors in the elicited requirements	3	4	7
Assign priority to the requirements	2	5	7
Identify and resolve inconsistencies in the requirements	1	1	2
Confirm and refine common/individual requirements	1	2	3
Analyze risk of the identified requirements causing schedule delay or cost overrun	1	2	3

Table 4. Cont.

<b>List of activities conducted to specify client requirements</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Document requirements specification	3	5	8
Build requirements traceability table	2	1	3
Create test cases based on requirements	3	1	4
<b>List of activities conducted to verify client requirements</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Reach agreement on individually specified requirements with clients (checking sufficiency and consistency of the derived requirements)	3	5	8
Check implementation results based on requirements	3	5	8
<b>List of activities to manage client requirements</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Service cost calculation per requirement	1	0	1
Requirements change management	2	4	6
Requirements change history & content management	3	3	6
Requirements change impact analysis	1	4	5
Identify and control requirements change (Change Control Board)	2	0	2
Establish mutual traceability between requirements and deliverables in phases (design documents, source code, etc.)	3	1	4
Review consistency between requirements and project deliverables affected by requirements change	2	0	2

▪ SaaS-Cloud service construction phase

Enterprises participating in the survey generally had no difficulties in SaaS cloud service implementation, but were relatively weak in designing and constructing architecture. In particular, they displayed unsatisfactory levels of initial architecture definition, variability modeling, reference architecture design and utilization. Further, early stage identification of opportunities for development, procurement, or reuse as a function of platform status, along with in-depth design, data design and documentation activities were generally insufficient. A difference based on organization size was observed: large enterprises designed conceptual and functional architecture but fell behind in reference architecture design and utilization, where mid-to-small enterprises performed internal and external interface design relatively well while falling behind in overall architecture design activities.

Software reusability, one of the core SaaS cloud service development activities, was generally insufficient. Both large enterprises and mid-to-small enterprises exercised efforts to reuse modules, components and source codes, but severely lacked architecture reuse. Some enterprises did not reuse their architecture at all.

Regarding development support activities, mid-to-small enterprises showed low levels in the application of standard processes. Large enterprises did have defined organization specific processes and methodologies but were limited in their utility due to ambiguous guidelines or standards



**Table 5.** Survey results from process maturity measurement at SaaS cloud construction phase.

List of design activities	Large Enterprises	Mid-to-Small Enterprises	Total
Identify what to develop in-house, procure external solution for, and to reuse legacy system	2	3	5
Design conceptual or functional architecture	3	1	4
Design and utilize reference architecture	0	2	2
Define common architecture and model architectural variability	1	4	5
Design and document interface between internal and external systems	2	5	7
Design and document detailed processes and data	2	3	5
List of implementation activities	Large Enterprises	Mid-to-Small Enterprises	Total
Preemptively identify what to develop in-house, procure external solution for, and to reuse legacy system	2	3	5
Implement based on completed designs	2	4	6
Create user manuals, operator manuals, and online help for finalized product	2	4	6
List of activities to facilitate software reuse	Large Enterprises	Mid-to-Small Enterprises	Total
Reuse legacy source code	2	4	6
Reuse legacy execution modules and components	1	2	3
Reuse legacy architecture	1	1	2
List of development support activities	Large Enterprises	Mid-to-Small Enterprises	Total
Establish standard development methodology	2	1	3
Establish standard development process	3	1	4
Utilize development support tools	2	2	4

▪ SaaS cloud service operation and management phase

In this phase, data collection and management procedures to evaluate service operation quality were not quantitatively defined, and further insufficiencies were found in data verification and data analysis result management. In large enterprises, the metrics for service capacity measurement were defined, but the process for metric sampling and management were undefined. In mid-to-small enterprises, overall service capacity measurement was unsatisfactory in terms of process definition and operation.

SaaS service modification and distribution activities were insufficient in performance due to poor management of data. The majority of organizations did not maintain guidelines for service modification and did not establish an effective approval process. In some organizations, configuration management environment for maintaining modification history were not in place. Automatic service distribution was not supported in almost all organizations.

**Table 6.** Survey results from process maturity measurement at SaaS cloud service operation and management phase.

<b>List of preparatory activities preceding service monitoring</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Establish if quantifiable target information exists, then define the goal of measurement	3	4	7
Define measurement targets necessary to acquire the quantitative information as described in measurement goals	3	4	7
Define data collection procedures (collection frequency, method, role, <i>etc.</i> ) and storage procedures	2	3	5
<b>List of activities to monitor service</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Collect and verify data defined as measurement targets (to detect data error, tampering, or omission)	2	2	4
Analyze collected data, review analysis results, and conduct corrective actions if necessary	1	1	2
Store collected data and analysis results (with control over repository access privileges for data modification and update)	1	0	1
<b>List of activities to modify/distribute service</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Maintain configuration management tools and supporting systems to manage service modification history	2	4	6
Provide a guideline for modifying services	2	1	3
Provide an approval process for modifying services	2	1	3
Provide an automated process for modifying and distributing services	1	0	1
Define formal process for modifying and distributing services	2	0	2
<b>List of activities to support service while under contract with a client</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
Service is made immediately available after all process steps are automatically performed	1	1	2
Service is made available after it has been installed under predefined process	1	3	4
Service is made available after development work has been completed to satisfy additional requirements	2	3	5
<b>List of activities to manage capacity</b>	<b>Large Enterprises</b>	<b>Mid-to-Small Enterprises</b>	<b>Total</b>
A system for anticipating service capacity is established	1	4	5
Quantified metric for evaluating service capacity is established	0	3	3
A standard for increasing/decreasing service capacity exists	1	4	5
Service capacity increase/decrease is automated	1	0	1
Service capacity management process is repetitively applicable	1	0	1

## 5.2. Identification of Required Activities from Industrial Surveys

The survey results showed the weak points which the SaaS cloud service companies currently have in terms of process maturity. In order to find out if the identified weak points could be remedied by applying the best practices extracted from success stories of leading organization, we mapped the unsatisfactory activities with corresponding best practices that could solve the problem.

Table 7 shows the results. Each weak point was confirmed to be positive in its applicability to be supplemented by the four best practices proposed in Section 4. To define “what” each best practice must do for efficient SaaS cloud service development, practical activities must be defined in detail, instructing “how” to do it. We identified necessary activities from the survey data and then mapped them with related individual best practices as listed in Table 7. For example, “*deriving user requirement per interested party*” and “*defining and refining common/individual requirement*” can be supplemented with the strategy that gradually extending the scope of cloudization, that is, the firstly identified best practice called “*Transform application to the cloud computing environment incrementally*”. Each item in the weak point column is labelled with (R), (C), and (M), which means activities belonging to the Requirements phase, Construction phase, and Operation and Management phase, respectively. To realize the best practices, as listed in the right-most column of Table 7, activities such as “*Market Requirements Analysis*”, “*Business Trend Analysis*”, “*Legal and Regulatory Analysis*”, “*Tenant Analysis*”, and “*Cloudization Scoping*” must be performed.

Apart from the weak points that can be mapped to each best practice, some weak points can be resolved with defining general process only. “*Repeatedly apply service capacity management process*,” “*Define formal service change and distribution process*,” “*Provide standard development methodology*,” and “*Provide standard process*”—listed on the last row of Table 7—are the insufficient activities due to the absence of documented guidelines. They can be resolved by creating a well-defined process guideline because it is hard to tell that they are related to a certain best practice.

**Table 7.** Relationship between diagnosed weak points and required activities via four best practices for the SaaS cloud environment.

Weak Points	Best Practices	Required Activity
<ul style="list-style-type: none"> <li>Identify and manage stakeholders(R)</li> <li>Identify common and individual user requirements (R)</li> </ul>	<ul style="list-style-type: none"> <li>Transform application to the cloud computing environment incrementally</li> </ul>	<ul style="list-style-type: none"> <li>Market Requirements Analysis</li> <li>Biz. Trend Analysis</li> <li>Legal and Regulatory Analysis</li> <li>Tenant Profiling</li> <li>Cloudization Scoping</li> </ul>
<ul style="list-style-type: none"> <li>Design conceptual or functional architecture (R)</li> <li>Design and utilize reference architecture (R)</li> <li>Reuse legacy execution modules and components (R)</li> <li>Reuse legacy architecture (R)</li> <li>Utilize development support tools (C)</li> <li>Provide an automated process for modifying and distributing services (M)</li> <li>Service is made immediately available after all process steps are automatically performed(M)</li> <li>Service is made available after it has been installed under predefined process (M)</li> <li>Quantified metric for evaluating service capacity is established (M)</li> <li>Service capacity increase/decrease is automated (M)</li> </ul>	<ul style="list-style-type: none"> <li>Respond resiliently to extended changes</li> </ul>	<ul style="list-style-type: none"> <li>Implementation of Adaptable Components</li> <li>Conceptual Architecture View Definition</li> <li>Data Architecture View Definition</li> <li>Specification Commonality and Variation Analysis with Feature Model</li> </ul>

Table 7. Cont.

Weak Points	Best Practices	Required Activity
<ul style="list-style-type: none"> <li>Service cost calculation per requirement (R)</li> <li>Identify and resolve inconsistencies in requirements (R)</li> </ul>	<ul style="list-style-type: none"> <li>Consider multi-tenants for an application</li> </ul>	<ul style="list-style-type: none"> <li>Specification Commonality and Variation Analysis with Feature Model</li> <li>Tenant-specific Service Configuration Phase</li> </ul>
<ul style="list-style-type: none"> <li>Elicit constraints (R)</li> <li>Analyze risk of the identified requirements causing schedule delay or cost overrun (R)</li> <li>Build requirements traceability table (R)</li> <li>Create test cases based on requirements (R)</li> <li>Identify and control requirements change (Change Control Board)(R)</li> <li>Establish mutual traceability between requirements and deliverables in phases (design documents, source code, etc.)(R)</li> <li>Review consistency between requirements and project deliverables affected by requirements change (R)</li> <li>Provide a guideline for modifying services (M)</li> <li>Provide an approval process for modifying services (M)</li> <li>Collect and verify data defined as measurement targets (to detect data error, tampering, or omission)</li> <li>Analyze collected data, review analysis results, and conduct corrective actions if necessary(M)</li> <li>Store collected data and analysis results (with control over repository access privileges for data modification and update)(M)</li> </ul>	<ul style="list-style-type: none"> <li>Manage vulnerability of software</li> </ul>	<ul style="list-style-type: none"> <li>Conceptual Architecture Change Management</li> <li>Data Architecture Change Management</li> <li>Component Change Management</li> <li>Requirements Traceability Management</li> <li>Evaluation of Conceptual Architecture</li> <li>Evaluation of Data Architecture</li> </ul>
<ul style="list-style-type: none"> <li>The service capacity management process is repetitively applicable (M)</li> <li>Define formal process for modifying and distributing services (M)</li> <li>Establish standard development methodology (C)</li> <li>Establish standard development methodology (C)</li> </ul>	<p>Develop a Process Framework</p>	

Using the list of required activities summarized at the rightmost column of Table 7, we proceed with designing SCoDP as previously depicted in Figure 2.

## 6. Feedback on SCoDP from a Process Deployment Case

In order to understand the utility of designed SCoDP, we applied it to a SaaS cloud service development project at a mid-sized company and collected feedback from the developers on each proposed activity. The following are the brief profiles of the company:

- Company Name: Tilon ([www.tilon.co.kr](http://www.tilon.co.kr))

- Number of Employees: 33 developers (22 SaaS cloud service developers)
- Main Business:
  - SaaS cloud service development
  - Presentation client desktop virtualization
  - Smart office products
  - Thin client, zero client
- Average career length of SaaS cloud service development team: 12 years

Like many mid-sized companies, *Tilon* did not have clearly defined development methodology and process. Though there wasn't a standardized process routinely called and used, *Tilon* developed SaaS cloud services through a process defined by a tacit understanding based on years of business experience.

The SaaS cloud service developed through SCoDP was a cloud computing solution called *Astation*. *Astation* provides cloud based services where Windows-based applications installed in the server can be used on various terminal devices regardless of native operation system (Windows, iOS, Android) through use of presentation virtualization technology. This solution installs applications and data in servers, which are used to be installed and managed on clients by each user. Then, the client supports various technologies required for running high-definition video/Flash, IE (Internet Explorer), and virtual IP (Internet Protocol), apart from the basic technology of virtualization, which displays the executed results of the servers only. The user is able to apply his or her work environment whenever, wherever, and whatever the terminal is. Two challenging factors were identified to be considered when developing *Astation*.

- Obscurity and diversity of customer range: *Astation* is not limited to a certain domain but used in many domains (manufacturing, finance, public service, education, hospital, etc.) that it needs to be analyzed in the aspect of marketing for requirements from the general.
- Changeability per client/tenant: *Astation* service is operated for each client, and the group of certain users in the client company is defined as a tenant. Therefore, feature sets—varying with client—is decided at the moment of contract. The users in the same client company can be defined as a number of user groups depending on the authority for using applications or security access. A tenant is decided for each user group, and applicable service set is different for each tenant.

The two factors above are defined as risks that can be resolved when SCoDP is applied, keeping in line with the category of challenging factors of SaaS cloud service development that were identified in Section 3.

*Astation* was developed by nine developers for about seven months, according to the guidelines of SCoDP. It currently provides service. We posed the nine developers questionnaires asking how the guideline about each activity of SCoDP is helpful for developing *Astation*. We asked them to score the most leaf node activity by selecting a nominal value from 1 to 4 (1: Very Unhelpful, 2: Unhelpful, 3: Helpful, 4: Very Helpful). The right column of Table 8 shows the average of the nine developers' answers that had been rounded off to two decimal places. The higher activities having sub-activity have the average value of the sub-activity score indicated.

Feedback results in Table 8 indicate that the average score of the whole activities of SCoDP is around 3.45. It reveals that *Astation* developers evaluate SCoDP as helpful for the actual development process.

Activities in Table 8 that are in the shaded rows are the ones that received an average score below 3. We asked the developers to explain why, if any, negative opinions (scoring 1 or 2). The following are the answers. Firstly, in the case of *Legal and Regulatory Analysis* activity, *Astation* was relatively less restricted by certain regulations or rules. It seems, as it does not have many opportunities to follow provided guidelines, it was given a low score for the question about its utility. In case of *Cloudization Scoping* activity, *Astation* is not the project converting original legacy service to cloud environment, but it is planned to be an SaaS cloud service from the planning stage. As it does not require special scoping, developers do not seem to feel the need for it. On the other hand, in cases of *Evaluation of Conceptual Architecture* activities and *Implementation of Adaptable Components* activities, they appear to have difficulties in adapting to new techniques and automation tools. It is necessary to make the components able to self-adapt in order to realize real-time provisioning. However, it is a technique that demands quite a difficult skill set. It seems a gradual adaptation is necessary.

**Table 8.** Feedback from *Astation* developers.

SCoDP Activity	Avg. Score
Multi-tenant Based Service Development Phase	
1. Multi-tenant Requirements Analysis	3.5
1.1 Market Requirements Analysis	3.2
1.1.1 Biz Trend Analysis	3.6
1.1.2 Legal and Regulatory Analysis	2.9
1.1.3 Tenant Profiling	3.8
1.1.4 Cloudization Scoping	2.3
1.2 General/Tenant Specific Requirements	3.7
1.2.1 Specification Commonality and Variation Analysis with Feature Model	3.7
1.2.2 Constraints Specification	3.6
2. Reference Architecture Design	3.3
2.1 Reference Architecture Design	3.0
2.1.1 Conceptual Architecture View Definition	3.1
2.1.2 Evaluation of Conceptual Architecture	2.9
2.2 Data Architecture Design	3.4
2.2.1 Data Architecture View Definition	3.5
2.2.2 Evaluation of Data Architecture	3.3
2.3 Architecture Change Management	3.6
2.3.1 Requirements Traceability Management	3.8
2.3.2 Conceptual Architecture Change Management	3.5
2.3.3 Data Architecture Change Management	3.6
3. SaaS-Cloud Service Component Development	3.0
3.1 Implementation of Adaptable Components	2.8
3.2 Component Change Management	3.1
Tenant-Specific Service Configuration Phase	
1. Requirements Selection for a Tenant	3.9
2. Architecture Authoring for a Tenant	3.7
3. Dynamic Component Reconfiguration for a Tenant	3.3

## 7. Conclusions and Future Work

We have proposed a set of best practices to satisfy the requirements emerging from adopting new software development processes in the new IT environment paradigm centered on SaaS cloud computing. In addition, we have presented four new best practices to help realize the proposed development process in the field of actual SaaS cloud service development. The value of the proposed process as a guideline towards cloud service adoption has been illustrated through its application to a SaaS cloud service development case, *Astation*. The resulting feedback from developers who applied the proposed process to their work has been very positive. The core concept of the proposed process identifies itself with the goal of cloud computing, which is to prevent repetitive use of resource as well as to prevent redundant development, and focus on maximizing process efficiency through utilizing reusable, sharable assets as much as possible. In that sense, the proposed software development approach of this paper is aligned with the efforts to find ways to provide the most flexible yet sustainable service to all users who come with various requirements born out of the era of cloud computing. The proposed process will continue to be refined along with ongoing attempts to apply the process to multiple upcoming SaaS cloud service development projects, resulting in feedback loops that strengthen our efforts.

## Acknowledgments

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2012M3C4A7033348).

## Author Contributions

The research and paper were designed and written by Soojin Park. Mansoo Hwang and Sangeun Lee were responsible for surveying the industrial data. The collected data is analyzed by Young B. Park.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Cloud Computing. Available online: [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing) (accessed on 15 January 2015).
2. Chang, V.; Wills, G.; de Roure, D. A review of cloud business models and sustainability 2010. In Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, 5–10 July 2010.
3. Song, Y.; Pang, Y. Leveraged BMIS Model for Cloud Risk Control. *J. Inf. Process. Syst.* **2014**, *10*, 240–255.
4. Song, Y.; Pang, Y. How to Manage Cloud Risks Based on the BMIS Model. *J. Inf. Process. Syst.* **2014**, *10*, 132–144.

5. Hussain, R.; Oh, H. Cooperation-Aware VANET Clouds: Providing Secure Cloud Services to Vehicular Ad Hoc Networks. *J. Inf. Process. Syst.* **2014**, *10*, 103–118
6. Tsai, J.C.; Yen, N.Y. Cloud-Empowered Multimedia Service: An Automatic Video Storytelling Tool. *J. Conver.* **2013**, *4*, 13–19.
7. Kim, J.S.; Byun J.; Jeong, H. Cloud AEHS: Advanced learning system using user preferences. *J. Conver.* **2013**, *4*, 31–36.
8. Weng, M.M.; Shih, T.K.; Hung, J.C. A Personal Tutoring Mechanism based on Cloud Environment. *J. Conver.* **2013**, *4*, 37–44.
9. Berena, A.J.; Chunwijitra, S.; Okada, H.; Ueno, H. Shared virtual presentation board for e-Meeting in higher education on the WebELS platform. *Human-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-6.
10. Augusto, J.C.; Callaghan, V.; Cook, D.; Kameas, A.; Satoh, I. Intelligent Environments: A manifesto. *Human-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-12.
11. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019 White Paper. Available online: [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white\\_paper\\_c11-520862.html](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html) (accessed on 15 January 2015).
12. Malkawi, M.I. The art of software systems development: Reliability, Availability, Maintainability, Performance (RAMP). *Human-Centric Comput. Inf. Sci.* **2013**, doi:10.1186/2192-1962-3-22.
13. Astation. Available online: [http://www.tilon.co.kr/solution/accord\\_intro1.asp](http://www.tilon.co.kr/solution/accord_intro1.asp) (accessed on 15 January 2015).
14. Overview and Issues for Implementation of the Federal Cloud Computing Initiative: Implications for Federal Information Technology Reform Management. Available at <http://fas.org/sgp/crs/misc/R42887.pdf> (accessed on 15 January 2015).
15. Wuxi Smart City Development Design. Available online: [https://www.academia.edu/4403337/Wuxi\\_Smart\\_City\\_Development\\_Design](https://www.academia.edu/4403337/Wuxi_Smart_City_Development_Design) (accessed on 15 January 2015).
16. An Analysis of G-Cloud Sales. Available online: <http://www.eduserv.org.uk/blog/2013/04/29/an-analysis-of-g-cloud-sales/> (accessed on 15 January 2015).
17. Japan to Tap Government Potential through the Kasumigaseki Cloud. Available online: <http://www.sucreproject.eu/content/japan-tap-government-potential-through-kasumigaseki-cloud> (accessed on 15 January 2015).