RC24932 (C1001-004) January 12, 2010
Computer Science

IBM Research Report

Software Outsourcing Project Governance Platform

Ying Liu, Xin Zhou, Yabin Dang, Wei Zhao
IBM Research Division
China Research Laboratory
Building 19, Zhouguancun Software Park
8 Dongbeiwang West Road, Haidian District
Beijing, 100193
P.R.China

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Software Outsourcing Project Governance Platform

Ying Liu
IBM China Research Lab
ZGC Software Park, # 19 Dong Beiwang Road,
Shangdi
Beijing, 100094, P.R.C.
86-10-58748070
aliceliu@cn.ibm.com

Xin Zhou, Yabin Dang, Wei Zhao
IBM China Research Lab
ZGC Software Park, # 19 Dong Beiwang Road,
Shangdi
Beijing, 100094, P.R.C.
86-10-58748000
xinzhou@cn.ibm.com,dangyb@cn.ibm.com,
weizhao@cn.ibm.com

ABSTRACT

It is well known that software outsourcing projects are more difficult to be managed because of cross organizations participated, especially for those projects with multiple level outsourcing. Although some software vendors passed the certification of CMMI or ISO9000, it is still a big challenge to execute a project in a consistent governance mechanism when the activities happened in different organizations can not be controlled in a unified platform. In this paper, we introduce a web-delivered software engineering service platform to facilitate the governance to software outsourcing projects. Service providers can provide independent services to the platform which needs to be evaluated the compliance degree with the process areas of CMMI, therefore, the consumers can select the appropriate service set to construct the specific governance supporting environment which complies with CMMI in a quantitative degree. Then the software outsourcing projects can be executed under such unified governance supporting environment to guarantee the practical activities are controlled in a consistent governance model. We not only present the framework of governance platform, but also introduce one implementation, called Internet-based Software Outsourcing Platform.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management –*Software process models (e.g. CMM, ISO, PSP).*

General Terms

Management, Economics, Human Factors, Standardization.

Keywords

Software outsourcing, process governance, CMMI, software engineering as a service, project-oriented governance model.

1. INTRODUCTION

In the age of global economy, many of enterprises are outsourcing their software development to other regions no matter because of resource problems or cost saving. However, everyone is aware of the difficulties and risks of outsourcing projects brought by cross-culture challenges, multiple involved vendors, requirement ambiguity, resource instability, etc [1]. A lot of research work and commercial solutions are emerging to address the challenges faced by software outsourcing industry. For example, IBM integrated software delivery platform targets to solve the integration problem among different development phases, such as requirement capturing, coding, testing, bug fixing, etc. [2] The HP portfolio of outsourcing services spans a broad spectrum that includes infrastructure management, application outsourcing, end-user workplace solutions, business continuity services and business process outsourcing[3]. Most of existing solutions, services and products focus on traditional software development activities, including requirement management, development environment, change management, configuration management, project management, test management and so on. Besides these traditional challenges, outsourcing project is more difficult to be controlled because of additional four major reasons:

1. *Cross-organization*: An outsourcing project is often cross different organizations, then the unified governance model among multiple organizations are critical important.

2. *Project-oriented governance model*: For different projects, the required governance model may be different even the participants are the same. It means that the governance supporting system should be flexible enough to accommodate different usage scenarios.

3. *Practical process compliance*: In order to reduce the risk, most outsourcing projects require the vendors should pass CMMI, ISO or other certification. However, the certification to the organization doesn't mean that the project team can comply with the CMMI process or ISO standard in practice.

4. *Open 'black box'*: Software outsourcing project is often sub-contracted to multiple vendors, the activities of vendors are not visible to the project owner. This kind of 'black box' project management may bring high risk. It is necessary to enable all the project participants execute the project in a unified platform.

It is obvious that the specific challenges above mentioned can not be solved by existing software engineering or project management products. As we know, most of buyers only take the vendors who passed CMMI or ISO9000 as their candidates, which will exclude a large quantity of vendors out of their candidates list. However, the real situation is that the certified vendor does not mean that they can fully comply with CMMI processes while executing the outsourcing projects because they often sub-contract part of work to other small companies.

Fortunately, most of people have been customized to working in the internet environment, and the concept of web-delivered service has been adopted by more and more people. We propose to leverage the business model of Software as a Service (SaaS) to build a web-delivered governance service platform to serve software outsourcing projects. This governance platform should be flexible enough to provide governance services for different projects in different levels. For example, some projects are required to comply with CMMI level 5, and some projects are required to comply with CMMI level 2. The platform can compose the selected services as a unified supporting environment to satisfy the compliance requirement of the specific project. Because the governance policies are embedded to the governance platform, a project can strictly follow the governance rules embedded to the platform if all the activities of a project are controlled by the platform.

Similar to the name of SaaS, we called the platform which delivers software development services to the users through internet as Software Engineering as a Service (SEaaS) platform. The platform has two major types of users: one is software engineering service providers; the other is software engineering service consumers. The platform has a common governance model, for example, a platform may select CMMI as the governance model, or it takes ISO 9000 as the governance model. Any services contributed to the platform must follow the governance model. We provide quantitative assessment mechanism to evaluate the compliance degree with the related governance rule. A service consumer can raise their request of governance goal, for example, a project requires to following CMMI level 2. The platform will present the CMMI level 2 service set, so that the user can select the appropriate services to meet their governance requirement. The platform can integrate the selected services as a unified supporting environment to meet the governance requirement to the specific project. This general governance framework can have different implementations based on difference governance standards. In papers, we take CMMI as an example to present the governance platform and one of its implementations. For the implementation of the platform, it is critical important to guarantee the platform is flexible enough to enable service plugin and integration. While introducing the platform example, we use Jazz Foundation [4] as the fundamental platform to enable the scalability, extensibility, and composibility of the governance platform.

The rest of the paper is organized as follows. In Section 2, we cover the related works. Section 3 introduces CMMI, SEaaS and the compliance degree evaluation method. Section 4 describes the sample governance platform that takes CMMI as the standard governance framework, including its usage process and platform architecture. Section 5, present an instance of the governance platform introduced in section 4 that facilitate project-oriented governance to software outsourcing projects. Section 6 concludes and gives directions for further research.

2. RELATED WORKS

With the increasingly significant role software applications place in present businesses, the aspiration for close alignment of software applications and business goals becomes more and more urgent. On this background, Software Development Governance (SDG) is widely studied by academic and industries for a clear understanding of the value that software development projects provide, as well as the risks they carry. In [7] and [8], the static component of SDG is defined as establishing chains of responsibility, authority and communication to empower people within a software development organization. The dynamic component of SDG is defined as establishing measurement and control mechanisms to enable software developers, project managers and others within a software development organization to carry out their roles and responsibilities.

[3] discuss SDG's relationship to management and process. Starting with an understanding of the enterprise strategy and goals, SDG establishes chains of responsibility and authority and setup measurement and control mechanisms. This determines the management structure that makes decisions about directing resources and assigning decision rights to them. Management also deploys processes to implement governance policies and work toward the strategic goals of the enterprise. Practitioners then participate in executing the processes. Also, [9] presents important concerns that have been adapted from the existing IT governance literature.

Software outsourcing is increasingly adopted by modern enterprises to leverage global resources, technological advancements and save cost. However, software outsourcing governance encounters significant challenges as traditional project planning, project management, and software process seldom consider the "distributedness" characteristic of software outsourcing. According to [10], the governance of distributed software governance should at least : 1) develop reliable and empirically tested dispersion measures that can be the basis for other software engineering activities 2) extend current project planning models related to cost estimation and team organization to accommodate the dispersion measures developed in step 1. 3) extends current normative process frameworks to accommodate key process areas specific to distributed environments, and 4) bridge the gap between software engineering and organizational learning approaches to build organizational level practices suitable for distributed software development. However, there has been few work published in this area except [11] on key process areas and [12] on process selection. [11] presents 24 new key process areas essential for managing distributed software product development and for continuously improving product management capabilities. [12] builds an empirical process choice model by analyzing the process choices made by two firms for executing more than a hundred offshore software development projects. The process choice model uses six easily measurable inputs that can be obtained at the starting phase of a software project.

In comparison to above research, our paper focuses on evaluating the compliance of a software outsourcing process to standard processes instead of studying the standard process specific for software outsourcing. Besides, our paper presents a SEaaS based software outsourcing governance model and its implementation with one SEaaS instance – ISOP, which is not ever mentioned in related works.

3. CMMI and SOFTWARE ENGINEERING SERVICES

3.1 Capability Maturity Model Integration

Capability Maturity Model Integration (CMMI) [5] is based on the notion of process area (PA). A process area is a cluster of related practices in an area. For example, Project Planning (PP) is one typical process area, which is to establish and maintain plans that define project activities through implementing some specific goals, specific practices, related work products, generic goals, and generic practices. CMMI has 22 process areas.

CMMI has been adopted by more and more organizations to help handle the complexity of today's products development in an integrated view. CMMI can reduce the cost of process improvement across enterprises that depend on multiple functions or groups to produce products and services. Currently, most of outsourcers only consider those companies passed CMMI certification while calling for vendors.

CMMI offers two different representations for approaching process improvement: continuous and staged. The continuous representation offers more flexibility for process improvement. An organization can choose a focused process area, determine the dependent process areas, improve these at priority, and then concentrate on other process areas. The staged representation uses predefined sets of process areas to define an improvement path for an organization. This improvement path is characterized by maturity levels. Each maturity level provides a set of process areas that characterize different organizational behaviors.

In this paper we only consider the staged representation. It means that the outsourcing organizations are allowed to require a project to be executed by following the defined maturity level. But in our method, process area is the key factor to drive the governance process of outsourcing project. Since the notion of process area is independent from the representation, our results hold for the continuous representation as well.

3.2 Software Engineering as a Service

Most traditional software engineering products are adopted by licensing mechanism, i.e. a product is deployed to serve the individual enterprises. While a collaborative project, such as software outsourcing projects, includes members from different organizations, it is necessary to provide the unified platform for smooth collaboration among team. Furthermore, different projects may have different governance requirements which mean that it is not easy to have a product satisfying different governance models. However, to build integrated management platform for a specific project is a cost-intensive work which can not be afforded by most of organizations.

In the era of internet, web-delivered services or SaaS have been recognized by more and more people. The pay-as-you-go model of SaaS can be leveraged to deliver software engineering services and governance services. We called it as Software Engineering as a Service (SEaaS). From technical perspective, SEaaS platform can leverage a lot of SaaS technologies, such as multi-tenancy, scalability and composition supporting [6]. For software engineering services, the common usage scenario is to integrate multiple services to support software development process. Therefore, it is critical important to introduce service evaluation mechanism to the platform. It means that any SEaaS platform should have independent assessment standard, so that any service to be contributed to the platform must be evaluated in a quantitative or qualitative way. For example, CMMI is an international framework and has been accepted by a lot of organizations. In this paper, we will take CMMI as the sample to introduce how to evaluate the compliance degree of a service. Meanwhile, the service implementation should be compatible with the platform and be composible with other services.

Figure 1 shows the conceptual diagram of SEaaS. While a consumer require a software governance supporting environment, the platform can offer an integrated platform with satisfying the required governance level since the individual services have compliance evaluation results. For software outsourcing projects, the service consumers can be any users involved in outsourcing projects, including outsourcer and outsource. So to provide unified governance platform to the ross-geography team will be much easier.

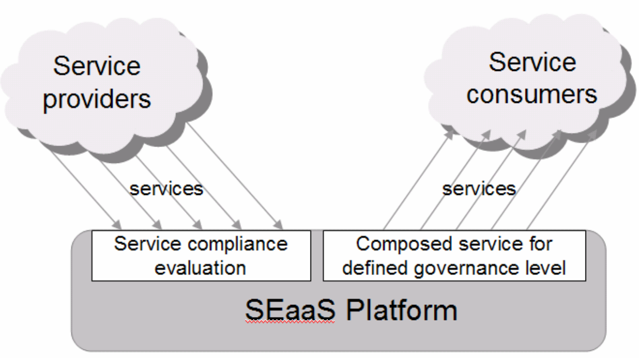

Service providers
services
Service consumers
services
Service compliance evaluation
Composed service for defined governance level
SEaaS Platform

Figure 1: **CONCEPTUAL DIAGRAM OF SEAAS**

Software engineering services provided by SEaaS platform can be categorized as four types: project management, engineering, process, and support. This category refers to the process category of CMMI.

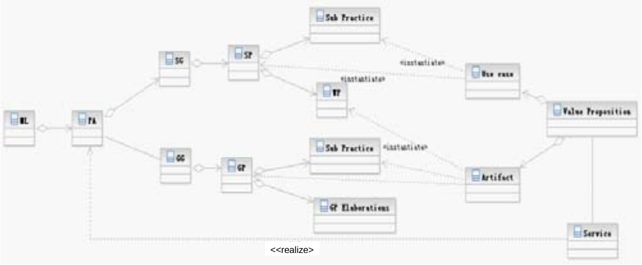
3.3 Service Compliance Evaluation

In CMMI, the major elements of the process area are sub-practices and work products of specific practice. Although general practices are also important, we ignore them in our compliance model because almost all the process area follows the same general practices. We focus on specific practice and work products in this paper.

The function of a service is specified with the use cases and related artifacts. Use case of service matches with the sub-practice of process area, and artifact of service matches with work product of process area. But in reality, the artifacts are often closed bounded with related uses. So the work products of sub-practices are directly matched with use case but not independent artifacts. The compliance model is showed in Figure 2. Three levels (high, middle, low) are defined to evaluate the matching degree between use cases and sub-practice or between artifacts and work products. We have a complex quantitative evaluation model for the

Conference'04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

compliance of service with standards. For the simplification, we simplify it in this paper but only presenting the qualitative compliance evaluation approach.



ML: Maturity Level; PA: Process Area; SG: Specific Goal; GG: General Goal; SP: Specific Practice; GP: General Practice; WP: Work Product

Figure 2: **SERVICE AND CMMI COMPLIANCE MODEL**

For example, a service called ‘optimized tasks partition and allocation (OTPA)’. This service provides the capability to automatically generate development tasks based on software design model. This service can bridge project management people and technical people while defining WBS (Work Breakdown Structure). From functional level, it dedicates to comply with Project Planning (PP).

The compliance evaluation is given in Figure 3. The up box describes the sub practices of PP. The left bottom box list some use cases of OTPA. And the right bottom box describes the compliance degree between use cases and sub-practices or work products. For the simplicity, we only list some samples in Figure 2. Overall, PP has 39 sub-practices and 46 related work products. We can notice that only partial sub-practices and work products are complied by the service. At the same time, some use cases of this service don’t have obvious mapping relationships with the sub-practices of PP. The compliance degree is marked with High (H), Middle (M), and Low (L) according to their compliance degree. This marking work is done by the expert of OTPA and CMMI.

If several use cases complying with one sub-practice in high level, it means that this sub-practice is strongly complied. On the contrary, if a sub-practice complies with a use case in low level, it means that this compliance is very weak. For the sample showed in Figure 3, there are 11 sub-practices and 14 work products are complied, which is account for 30% of the overall sub-practices and work products. In addition, the high compliance percentage is about 60%, the middle level compliance is about 30%, and the low compliance is about 10%. We can roughly calculate the overall compliance degree is about 0.12 (Supporting the weight of High, Middle and Low separately are W_{k1}=0.5, W_{k2}=0.3, W_{k3}=0.2, ComplianceDegree=30%*(W_{k1}*60%+W_{k2}*30%+W_{k3}*10%=0.12). These three indicators (sub-practice cover percentage, work products cover percentage and compliance degree) are presented to the users, so that they can select appropriate services.

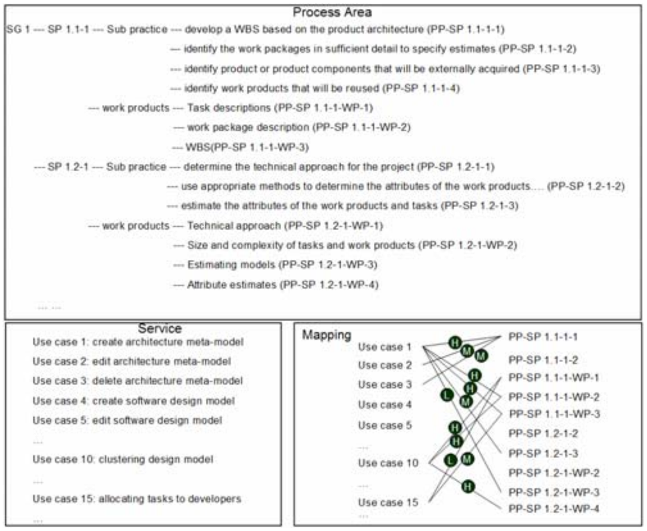

Process Area
SG 1 --- SP 1.1-1 --- Sub practice --- develop a WBS based on the product architecture (PP-SP 1.1-1-1)
--- identify the work packages in sufficient detail to specify estimates (PP-SP 1.1-1-2)
--- identify product or product components that will be externally acquired (PP-SP 1.1-1-3)
--- identify work products that will be reused (PP-SP 1.1-1-4)
--- work products --- Task descriptions (PP-SP 1.1-1-WP-1)
--- work package description (PP-SP 1.1-1-WP-2)
--- WBS(PP-SP 1.1-1-WP-3)
--- SP 1.2-1 --- Sub practice --- determine the technical approach for the project (PP-SP 1.2-1-1)
--- use appropriate methods to determine the attributes of the work products.... (PP-SP 1.2-1-2)
--- estimate the attributes of the work products and tasks (PP-SP 1.2-1-3)
--- work products --- Technical approach (PP-SP 1.2-1-WP-1)
--- Size and complexity of tasks and work products (PP-SP 1.2-1-WP-2)
--- Estimating models (PP-SP 1.2-1-WP-3)
--- Attribute estimates (PP-SP 1.2-1-WP-4)
... ...
Service
Use case 1: create architecture meta-model
Use case 2: edit architecture meta-model
Use case 3: delete architecture meta-model
Use case 4: create software design model
Use case 5: edit software design model
...
Use case 10: clustering design model
...
Use case 15: allocating tasks to developers
...
Mapping
Use case 1
Use case 2
Use case 3
Use case 4
Use case 5
...
Use case 10
...
Use case 15
...
H
M
M
H
H
L
M
H
H
L
M
H
PP-SP 1.1-1-1
PP-SP 1.1-1-2
PP-SP 1.1-1-WP-1
PP-SP 1.1-1-WP-2
PP-SP 1.1-1-WP-3
PP-SP 1.2-1-2
PP-SP 1.2-1-3
PP-SP 1.2-1-WP-2
PP-SP 1.2-1-WP-3
PP-SP 1.2-1-WP-4

Figure 3: **COMPLIANCE EVALUATION SAMPLE**

For a specific project, the project team can select one stage to be complied, such as CMMI level 3 for project x. Once a stage is selected, the group of process areas is clear to the platform. The platform can present the corresponding services complying with the required stage to the users. The user can select the appropriate services according to the compliance degrees. This is an intuitive approach, but it is easy to use. The details are introduced in section 4. We have a complex quantitative compliance evaluation system which can provide precise evaluation, but the service providers and consumers need to input more information.

4. PROJECT-ORIENTED GOVERNANCE PLATFORM

4.1 Project-oriented Governance Model

As we know, CMMI is often used to assess and improve the product and service process of enterprises. Different from the traditional usage model, this paper proposes to enable the project-oriented governance model which is described in Figure 4. As the first step, the project team can raised the governance requirement to the specific project. Having this request, the platform returns a suggested governance model to the user in the second step that includes the CMMI process areas. Since each process area may have multiple services complied, the users need to select the appropriate service set in the third step. The selected services are notified to the platform, and the identified services are composed as an integrated environment for supporting the processes in the final step.

If a project team requires the project to comply with CMMI level 2, it means that six process areas (Requirement Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance, and Configuration Management) need to be complied by the corresponding services. This is the raised request in step 1.

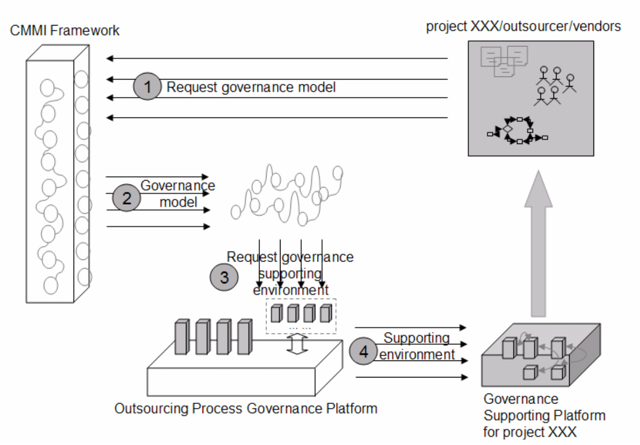

CMMI Framework
project XXX/outsourcer/vendors
1 Request governance model
2 Governance model
3 Request governance supporting environment
4 Supporting environment
Outsourcing Process Governance Platform
Governance Supporting Platform for project XXX

Figure 4: **Project-Oriented Governance Model**

In step 2, the submitted requested will be automatically handled and returned a service list. For each process area, it is possible that multiple services complied with it with a compliance degree indictor. The user can select the appropriate services according to the select criteria. The selection criteria is a complex multiple objects optimization problem, but it will not be covered in this paper for the simplicity. The user is presented with an intuitive view that shows the compliance degree with related process areas.

Once the service list is selected, it is returned to the platform for the integrated supporting environment generation as described in step 3. Then the platform will provide the integrated environment as the return in step 4. We'd like to address that the integrated environment doesn't mean that all the services are composed well for communication since it is difficult to guarantee any service is composible with others. But the user will be provided with an integrated portal to access the selected services.

4.2 Architecture of Governance Platform

For the above introduced project-oriented governance model, the implementation can follow the architecture in Figure 5. One critical feature of this platform is to guarantee the platform is easy to plug software engineering services. Jazz Foundation is recommended to be the fundamental platform, which is a scalable, extensible, and open source team-collaboration platform that integrates tasks across the software lifecycle. The platform also provides useful building blocks and frameworks (e.g. team repository, OSGi bundle, RESTful services, etc.) that facilitate the development of new products and tools. On top of it, a thin layer called project information and activity container, which manages the basic information and fundamental activities of software projects, including project creation, project work products submission, project closing, etc.

The platform supports two ways of service contribution mechanisms: one way is to embed services into the container; the other is to connect the service through adapters. From implementation perspective, these two mechanisms are not new but intuitive. The key is how to effectively manage these services, including register, selection, integration, assessment, evolving, and so on. In addition, how to manage the project-specific governance models and its supporting environment are very critical. Therefore, four components (service management, service compliance evaluation, project governance model management, project governance practical assessment) on top of project management foundation are must components to enable end-to-end project-oriented governance. Besides them, the platform has two optional components: Project Governance System Implementation and Governance System Improvement. If the selected services can be easily integrated, it will be easer to use. On the other hand, it the users can collect some accumulated governance execution information, the future governance supporting can become better and better which is called Governance System Improvement. For example, if some services are not used well in the past project execution, they will be ranked as low score. Otherwise, some services will be ranked high because of good feedback. This mechanism can help a healthy governance ecosystem.

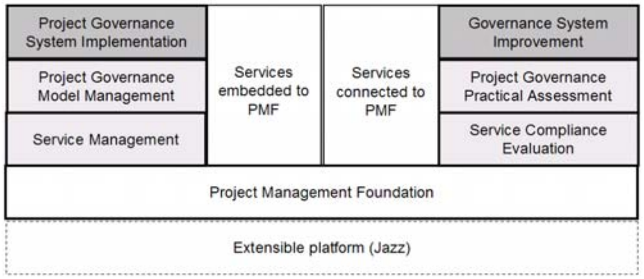

Project Governance System Implementation
Project Governance Model Management
Service Management
Services embedded to PMF
Services connected to PMF
Governance System Improvement
Project Governance Practical Assessment
Service Compliance Evaluation
Project Management Foundation
Extensible platform (Jazz)

Figure 5: **Governance Platform Architecture**

To implement this architecture on top of Jazz, all the components (including the container, the embedded services, and the connectors) follow the Jazz Integration Architecture (JIA), which is to enable diverse tools to be used together providing an integrated experience to their users. A typical component, providing a capability that helps with some facet of the software development lifecycle, is composed of a set of plug-ins, some of which are installed on the server and some on the client. The plug-ins use extension points to register with Jazz. In order to be more loose coupled inter-components, the governance platform component should expose its data and services through a REST API. These resource representations and web service should be OSLC(Open Services for Lifecycle Collaboration)-compliant for the governance domain, so that other tools can begin linking to these data resources via their URLs. New presentations can be created, relying only on details divulged in the REST API. And others can begin writing add-on tools that analyze the data resources, again, without needing to know more than what is revealed through the REST API.

4.3 A Governance Platform Sample

By following the architecture introduced in section 4.2, we implement a platform specifically for software outsourcing projects, called Internet-based Software Outsourcing Platform (ISOP). The platform architecture is given in Figure 6.

In the PMF layer, the components are designed specifically for software outsourcing projects. For example, it is a common situation that a project is outsourced to more than one vendor in the same level or multiple levels, therefore, the Hierarchical Project Information Management is provided for managing collaboration among different outsourcers and outsources. This function can help solve the big challenge about how to open the black box of managing outsourcing software projects. For software outsourcing projects, how to guarantee the project is executed or delivered by following contract or Specification of Work (SOW) are critical important from project management perspective. An independent component, called SOW/Contract Management is one of key components in PMF layer. To avoid missing contract items while executing the outsourcing project or over committing to the outsourcer, SOW is regarded as the first class citizen in the ISOP. SOW/Contract Management component provides the function to in-process track and check the status of delivery plan defined in the SOW or contract. The third component is Tenants Management that enables the isolation of the data and governance model for different companies, including service provides and service comsumers. It can fully leverage existing SaaS platform technologies since this is one common feature of SaaS platform.

The service management layer includes four major components. Service register serves service providers to submit services to the platform. Service selection, service composition and service enablement serve service consumers, including both outsourcers and vendors.

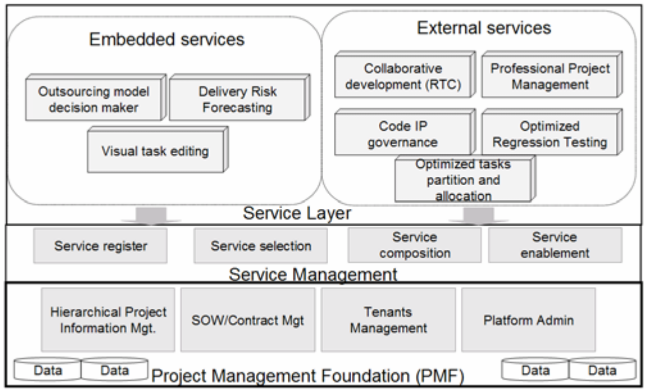

Embedded services
Outsourcing model decision maker
Delivery Risk Forecasting
Visual task editing
External services
Collaborative development (RTC)
Professional Project Management
Code IP governance
Optimized Regression Testing
Optimized tasks partition and allocation
Service Layer
Service register
Service selection
Service composition
Service enablement
Service Management
Hierarchical Project Information Mgt.
SOW/Contract Mgt
Tenants Management
Platform Admin
Data
Data
Project Management Foundation (PMF)
Data
Data

Figure 6: **Internet-based Software Outsourcing Platform**

The service layer includes eight implemented services in the current ISOP version which can be extended in future. Three of them (Outsourcing model decision maker, Delivery risk forecasting and Visual task editing) are embedded services including. Five of them (Collaborative development platform, Professional project management, Code IP governance, Optimized regression testing and Optimized task partition and allocation) are connected to the platform. In order to give a clear overview understanding about the services of ISOP, we give the brief introduction of these services as well as their complied process areas and compliance degree.

Outsourcing model decision maker (OMDM) [13]: For the outsourcing software project, this service acquires some project and organization basic information, including the project's characteristic, the client's and vendor's skill and expertise on this project, the client's expectation on the lifecycle of the project and the lifecycle of the project's final deliverables. Based on the analysis to the basic information, this service can recommend a dedicated process about how to execute the project by the outsourcer and vendors. It complies with two process areas, 'Decision analysis and Resolution' and 'supplier agreement management',

Delivery risk forecasting (DRF): Based on two key information, the collected real execution status information and delivery plan defined at the beginning stage, this service can help forecast the probabilities of the delivery risk in a continuous way. It complies with three process areas, 'risk management', 'measurement and analysis' and 'project monitoring and control'

Visual task editing (VTE): Different from traditional table based or form based task editing tools, this service provides a tree view to enable visual editing function for work breakdown structure (WBS) [14][15]. It complies with 'project planning'.

Collaborative development platform (RTC) [16]: It is an IBM Rational product to facilitate the collaboration among project team, such as developer and tester. It complies with 'Integrated project management +IPPD'.

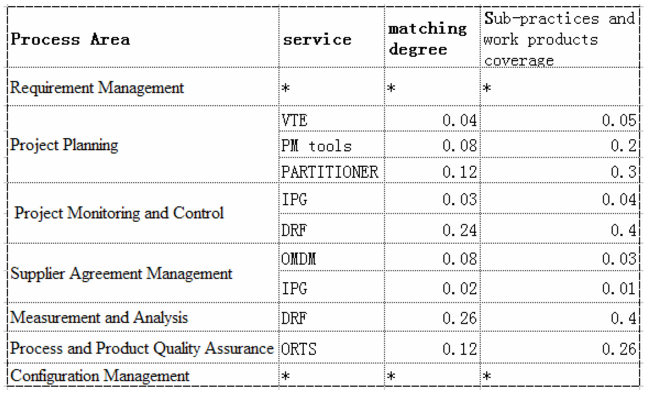
Professional project management (PM tools) [17]: It is traditional project management tool, such as Microsoft project 2003 and Gantt project. It complies with 'project planning'.

Code IP governance (IPG) [18]: This service provides the function to monitor developers' coding behaviors and generate warning reports if some predefined packages are found to be included in the software code. Besides that, it can also avoid illegal copy and past code actions. It complies with three process areas, 'supplier agreement management', 'risk management' and 'project monitoring and control'.

Optimized regression testing (ORTS) [19]: While the version is changed, this tool can help select the impacted test cases because of code change. It can avoid the labour cost and risk of missing test cases because of human selection. It complies with three process areas, 'Process and product quality assurance', and 'decision analysis and resolution'

Optimized tasks partition and allocation (PARTITIONER) [20]: Having the software design, this service can help partition the design as some tasks according to the principles of minimal correlation cost. And then the tasks can be allocated to the right person by following best matching principles. It complies with 'project planning'.

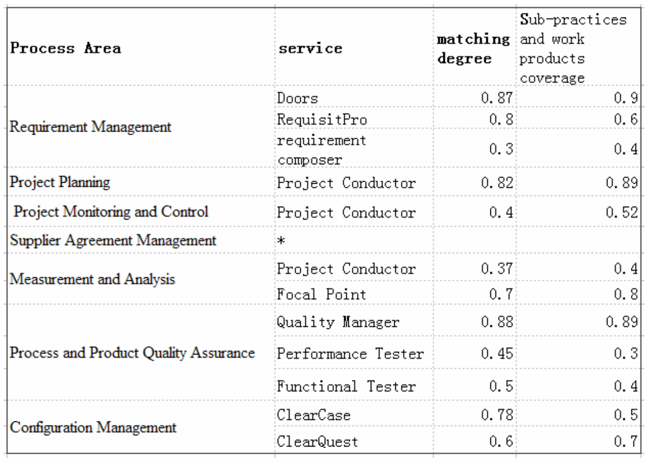
According to the compliance degree evaluation method introduced in Section 3.3, the compliance degree with CMMI process area of above eight services are calculated as showed in Table 1. If the value of a cell is star (*), it means that there is not committed service to comply with this process area. We must indicate that this evaluation is in coarse level since it doesn't consider the relationships among different services. In fact, if two services have Middle level compliance, it is possible that the combination of these two services can achieve High level compliance. Another situation is that the High level compliance may mean nothing to the users because one critical important sub-practice is not complied. These complex scenarios are not supported by the current evaluation method.



Process Area	service	matching degree	Sub-practices and work products coverage
Requirement Management	*	*	*
Project Planning	VTE	0.04	0.05
	PM tools	0.08	0.2
	PARTITIONER	0.12	0.3
Project Monitoring and Control	IPG	0.03	0.04
	DRF	0.24	0.4
Supplier Agreement Management	OMDM	0.08	0.03
	IPG	0.02	0.01
Measurement and Analysis	DRF	0.26	0.4
Process and Product Quality Assurance	ORTS	0.12	0.26
Configuration Management	*	*	*

Table 1: **ISOP Services Compliance Degree Results**

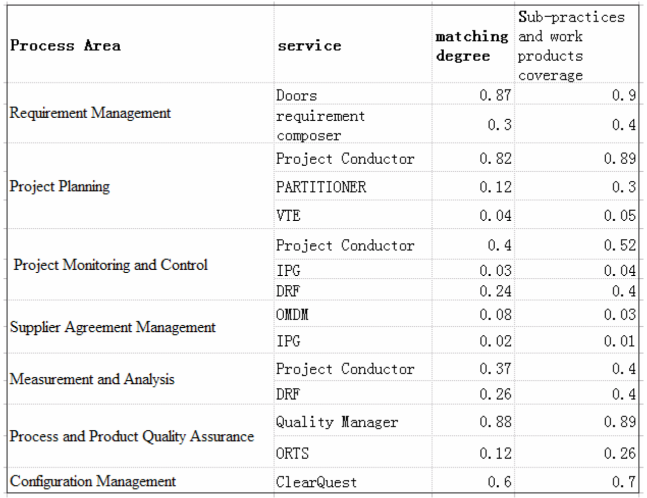
While the project team raised the request of the project governance level with CMMI 2, the platform will return the compliance degree data to the user as showed in Table 1. Because the current services don't comply with two important process areas, Requirement Management and Configuration Management, the user can consider products involved since the platform is able to connect other services. So, the user decides to select some products to enhance the compliance degree by referring to the compliance degree result of Rational Products showed in Table 2. The data included in Table 2 is only our personal evaluation results, but not from experts. In fact, we should invite the experts for each products to give the evaluation results in future.



Process Area	service	matching degree	Sub-practices and work products coverage
Requirement Management	Doors	0.87	0.9
	RequisitPro	0.8	0.6
	requirement composer	0.3	0.4
Project Planning	Project Conductor	0.82	0.89
Project Monitoring and Control	Project Conductor	0.4	0.52
Supplier Agreement Management	*		
Measurement and Analysis	Project Conductor	0.37	0.4
	Focal Point	0.7	0.8
Process and Product Quality Assurance	Quality Manager	0.88	0.89
	Performance Tester	0.45	0.3
	Functional Tester	0.5	0.4
Configuration Management	ClearCase	0.78	0.5
	ClearQuest	0.6	0.7

Table 2: **Products Compliance Data**

Based on the compliance data of services and products in ISOP given in Table 1 and Table 2, the project team can select the services to their specific project. Of course, the selection is not very simple if multiple constraints need to be considered, such as price, easy-to-use, and composability. It is a complex multiple objectives optimization problem which is not covered in this paper. Table 3 presents an example that a project team selects a service list for their specific project to comply with CMMI level 2. Although the number in the table is a little abstract, it is obvious that the bigger number means that the compliance degree is higher. The user can dive into the use case level to learn the functions of the services, furthermore, the compliance mapping between use cases and practices can be showed to the user.



Process Area	service	matching degree	Sub-practices and work products coverage
Requirement Management	Doors	0.87	0.9
	requirement composer	0.3	0.4
Project Planning	Project Conductor	0.82	0.89
	PARTITIONER	0.12	0.3
	VTE	0.04	0.05
Project Monitoring and Control	Project Conductor	0.4	0.52
	IPG	0.03	0.04
	DRF	0.24	0.4
Supplier Agreement Management	OMDM	0.08	0.03
	IPG	0.02	0.01
Measurement and Analysis	Project Conductor	0.37	0.4
	DRF	0.26	0.4
Process and Product Quality Assurance	Quality Manager	0.88	0.89
	ORTS	0.12	0.26
Configuration Management	ClearQuest	0.6	0.7

Table 3: **Selected Service List Example**

Since the project team selects the service list in Table 3, the governance platform to their specific project can be regarded as a governance platform customized from ISOP. The customized governance platform is showed in Figure 7. Compared to ISOP, the difference is only in the service layer, including the selected services and connected products.

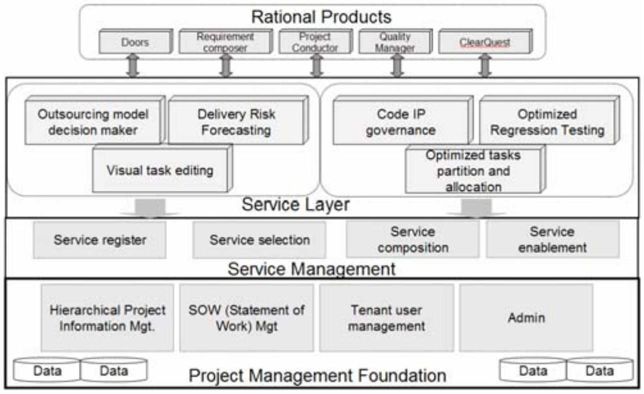


Figure 7: **Customized ISOP**

No matter for the services or products, it can be regarded as a set to comply with the process areas of CMMI. However, it is not equal to the integration of the services since the flexible integration or composition to support any projects is a hard work. In our complex service evaluation method, integration is one of evaluation indicator.

5. CONCLUSTION AND FUTURE RESEARCH

Thi paper mainly introduces a project-oriented governance model. It includes four major steps: the first step is to raise the governance request for the specific project; the second step is to present the services list complied with the raised governance model; thirdly, thes selected service list is returned to the platform for the integrated supporting environment generation; finally, the platform provides the integrated governance supporting environment to the project. Corresponding to the project-oriented governance model, we introduce a governance platform which can accommodate kinds of services to comply with the required governance model. This platform leverages the business model of SaaS to construct Software Engineering as a Service. Similar to SaaS platform, the governance platform has two types of users: one is service providers; and the other is service consumers. Different from common SaaS platform, the governance platform introduced here is required to have the corresponding governance model, for example, CMMI or ISO 9000 are governance model samples. How to evaluate the compliance degree of services with the compliance model is one of key aspects of this platform. We take CMMI as the governance model sample to introduce how to apply the simple qulititative evaluation method to evaluate the compliance degrees of services with the related process areas.

We present one implementation of the project-oriented governance platform for supporting software outsourcing projects, called Internet-based Software Outsourcing Platform. This platform targets to support outsourcing software project process complying with CMMI process areas. This platform is built on top of the extensible Jazz Foundation which enables two ways of service contribution: one is to plug services to the platform as the embedded services; and the other is to connect services with the platform through adapters. We give the overview of the basic function and compliance degree of the services including in the current version of ISOP. A sample is introduced to show how to enable a project-specific governance model.

The big challenges of the project-oriented governance platform have two aspects: one is the service integration mechanism which should be easier and smoothly connected with each other; the second is the service compliance quantitative evaluation method. For the former challenge, we propose to leverage the capabilities of Jazz Foundation. However, Jazz Foundation only provides the implementation level extensibility and scalability but not in business level integration. We are going to have deep study how to provide additional features on top of Jazz Foundation to supporting governance model-based service composition. For the latter challenge we introduced a simple coverage-based evaluation method in this paper. This method heavily depends on the knowledge of service experts. We already have a quantitative evaluation method for service compliance degree calculation. Unfortunatly, that method needs a lot of input, such as benchmark and service model. It is not easy to use. We are studying how to using Bayesian theory or Monte Carlo model to simulate the service compliance degree.

6. REFERENCES

[1] Natovich, Joseph. Vendor Related Risks in IT Development: A Chronology of an Outsourced Project Failure. Technology Analysis & Strategic Management, Vol. 13, No. 4, December 2003.

[2] Global Development and Delivery in Practice: Experiences of the IBM Rational India Lab. http://www.redbooks.ibm.com/abstracts/SG247424.html?Open

[3] Outsourcing Solutions for BFSI, http://dqindia.ciol.com/itcasebook/casebook-HP.pdf

[4] Jazz Integration Architecture Overview. https://jazz.net/projects/DevelopmentItem.jsp?href=content/project/plans/jia-overview/index.html

[5] CMMI for Development (CMMI-DEV). http://www.sei.cmu.edu/cmmi/tools/dev/index.cfm

[6] Mark Turner, David Budgen, Pearl Brereton. Turning Software into a Service. Computer, vol. 36, no. 10, pp. 38-44, Oct. 2003, doi:10.1109/MC.2003.1236470

[7] Murray Cantor, Sunita Chulani, Clay Williams, Avi Yaeli, Personal communication, 2006-2008

[8] Jensen, M. C. Cambridge. A Theory of the Firm: Governance, Residual Claims, and Organizational Forms. MA: Harvard University Press, 2001.

[9] Sunita Chulani, Clay Williams, Avi Yaeli. Software Development Governance and Its Concerns. 1st International Workshop on Software Development Governance, Leipzig, 2008.

[10] Narayan Ramasubbu, Rajesh Krishna Balan. Towards Governance Schemes for Distributed Software Development Projects. 1st International Workshop on Software Development Governance, Leipzig, 2008.

[11] N. Ramasubbu, M. S. Krishnan, and P. Kompalli. Leveraging global resources: A process maturity framework for managing distributed development. IEEE Software, vol. 22, pp. 80-86, 2005.

[12] N. Ramasubbu, R. Balan, and G. K. Tayi. An Empirical Model of Software Development Process Choice. International Symposium of Information Systems Indian School of Business, Hyderabad, 2007.

[13] Yanbing Jiang, Xin Zhou, Ying Liu, Hu Jingkun. Study on Neural Network-Based Software Outsourcing Process Pattern Decision. COMPUTER SYSTEMS & APPLICATIONS. 2009 18(6).

[14] Work Breakdown Structure. http://en.wikipedia.org/wiki/Work_breakdown_structure

[15] Carl L. Pritchard. Nuts and Bolts Series 1: How to Build a Work Breakdown Structure. ISBN 1-890367-12-5

[16] Rational Team Concert. http://www.ibm.com/developerworks/rational/products/rtc/

[17] Harold Kerzner. Project management: a systems approach to planning, scheduling, and controlling. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN-13978-0-471-74187-9.

[18] Ya Bin Dang, Ping Cheng, Lin Luo, Adrian Cho. A code provenance management tool for ip-aware software development. International Conference on Software Engineering (ICSE 2008), Leipzig, Germany, May 10-18, 2008. Informal demonstration research.

[19] Sheng Huang, Jun Zhu, Yuan Ni. ORTS: a tool for optimized regression testing selection. Proceeding of the 24th ACM SIGPLAN conference companion on Object Oriented programming systems languages and applications. Demonstration paper, Pages 803-804.

[20] Ying, Liu. PARTITIONER: Software Development Task Partitioning and Allocation Tool. Submitted to ICSE 2010 as demonstration research paper.