e-Government Standard Framework with Open Innovation Strategy

Eun-Ju Kim[1]
1 National Information Society Agency

Abstract. Improving the quality of e-Government Application and the productivity, SW development framework is needed. eGovFrame is an open source project to provide a standardized set of software tools for developing and running e-Government applications in Korea. It consists of four software environments: runtime environment for applications, development environment for application developers, management environment for framework managers, and operations environment for application operators. The open source initiative has led to continuous enhancements of the eGovFrame through an open community, with quarterly meetings between experts and a collaboration forum of public-private partnerships. The project's portal encourages voluntarily participation through open source software-from developers, suppliers and government officials into the implementation process.

1 Introduction

Korea has actively pursued e-Government as a crucial means to make its government more competitive, by leveraging the world's best information and communications technology (ICT) including broadband internet. After laying the groundwork for e-Government, the Korean government made the implementation of e-Government a major national agenda for the 2000s. As a result, e-Government has become firmly established in all areas of the Korean government and produced visible results. Accordingly, the effectiveness of the e-Government of Korea is widely acknowledged by the international community. The Korean e-Government is evaluated as one of the world's best by international organizations including the United Nations, and various e-Government systems are being exported to foreign countries [1].

To achieve these outcomes, the Korean government has performed many e-Government projects and made a lot of e-Government applications. For those projects, software frameworks were applied to a considerable part of them. From the report which is about the status of frameworks applied in e-Government projects from 2004 to 2007 [2], it is recognized that software frameworks were applied to seventy-one percent of e-Government projects in budget expenditure. Software framework is a useful tool to increase the productivity and quality for application development, and nowadays it become a popular tool for developing e-Government applications. On the other hand, there are some drawbacks originated from software frameworks.

By applying software frameworks, e-Government projects become highly dependent on IT companies' framework. Consequently, it is difficult to maintain an application without technical support from a framework provider who implemented the original application. In case of continuous projects, the framework applied in the previous project works as a technical barrier to new competitor, which is a vicious cycle of unfairness in software market. The dependence of IT companies' framework causes a number of problems. Firstly, Business logic of an application is also dependent on a certain Framework. Secondly, due to the certain framework being a black box type, only the framework provider can maintain the application, bringing lock-in for the framework provider. Third, multiple frameworks cause redundant activities in application process setup, recruit, education and maintenance.

In order to overcome these problems, the Korean government standardized a software framework, eGovframe (e-Government Standard Framework). eGovframe is a standardized set of software tools for developing and running e-Government applications in order to improve the efficiency of ICT investment and the quality of e-Government services. It focuses on improving the reusability and interoperability of e-Government applications by setting a standard framework for developing software of e-Government, ensuring the independency from IT companies by adopting open and neutral software tools and enhancing the competitiveness of IT SMEs by sharing the tools openly through the various channels.

To standardize the software framework, eGovframe, there were a lot of opinions and issues from many stakeholders. Large companies feared on dominated market collapse, public organizations concerned on whether they can get the stable technical support, developers rejected the newly developed tools, government concerned on business effectiveness, and SMEs concerned on project promotion centric only by large companies. So that, many stakeholders had to agree on standardization of the software framework. To realize the standardization of the software framework by overcoming these issues, we implemented the open innovation strategy with four phases: (1) Open sourcing, (2) Open process, (3) Open output and (4) Open ecosystem. We will discuss about the eGovframe and the open innovation strategy in detail.

2 Open Innovation Strategy

Open Innovation is a term promoted by Henry Chesbrough, a professor and executive director at the Center for Open Innovation at the University of California, Berkeley [3][4]. Open innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology. The boundaries between a firm and its environment have become more permeable; innovations can easily transfer inward and outward. The central idea behind open innovation is that in a world of widely distributed knowledge, companies cannot afford to rely entirely on their own research, but should instead buy or license processes or inventions (i.e. patents) from other companies. In addition, internal inventions not being used in a firm's business should be taken outside the company [3][4][5].

In order to solve the problems derived from the standardization of a software framework for e-Government, we implemented a strategy based on open innovation paradigm which is named by Open Innovation Strategy. The standardization of e-Government framework could not be realized solely by the Korean government’s effort. It requires not only the government’s effort and promotion, but also many stakeholders’ knowledge, participation, corporation and feedback. To lead these requirements for realizing the standardization and application of e-Government framework, our strategy is composed of 4 phases, open sourcing, open processes, open outputs and open ecosystem. Figure 1 shows the overall structure of Open Innovation Strategy.

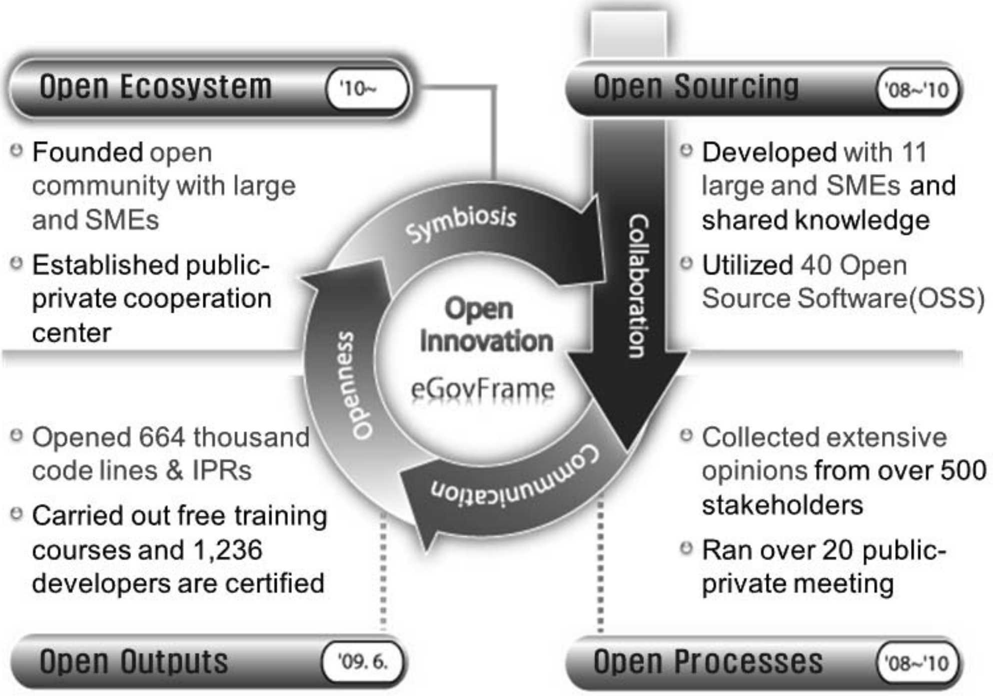

Open Ecosystem '10~
Founded open community with large and SMEs
Established public-private cooperation center
Open Sourcing '08~'10
Developed with 11 large and SMEs and shared knowledge
Utilized 40 Open Source Software(OSS)
Symbiosis
Collaboration
Openness
Communication
Open Innovation
eGovFrame
Opened 664 thousand code lines & IPRs
Carried out free training courses and 1,236 developers are certified
Collected extensive opinions from over 500 stakeholders
Ran over 20 public-private meeting
Open Outputs '09. 6.
Open Processes '08~'10

Fig. 1. Open Innovation Strategy

● **Open sourcing**

In order to standardize eGovframe, environment and functionality analysis on five major IT companies’ frameworks as well as a survey and intensive interview with every stakeholder were carried out. As a result, four environments that consist of thirteen service groups and fifty-four service functionalities were derived.

To prevent repeated development of same functions among government systems, an analysis on sixty-seven e-Government projects from year 2004 to 2007, more specifically, 31,114 functionalities were reviewed. Criteria that extract common functionalities for component were high probability on repeated development, reusability among government system and standard adoptability. After five refining processes passed, 219 common components were defined.

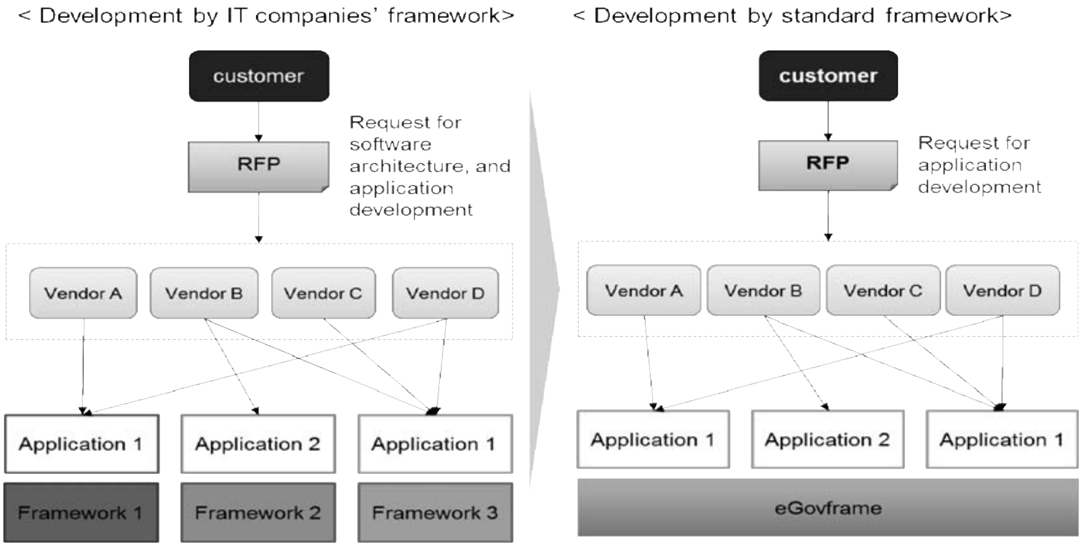


Fig. 2. To-be image of eGovframe

To reduce the dependency on major IT companies, well-known and proven open sources were selected. Using international software evaluation process model (ISO 14598) and practical software evaluation process (SEI PECA), open source software evaluation process for eGovframe was defined. In the first logical test, one hundred and seventy-five open source software were evaluated against requirements that mainly focused on the constraints for integration and interfaces of eGovframe. In the second physical test, eighty-five open source software, which derived from the first logical test, were evaluated for the basic functions and non-functional requirements. As a result, forty open source software were selected for composing eGovframe. Open source based eGovframe have several benefits. It can easily adopt fast-changing technologies and be utilized in the overseas e-Government applications.

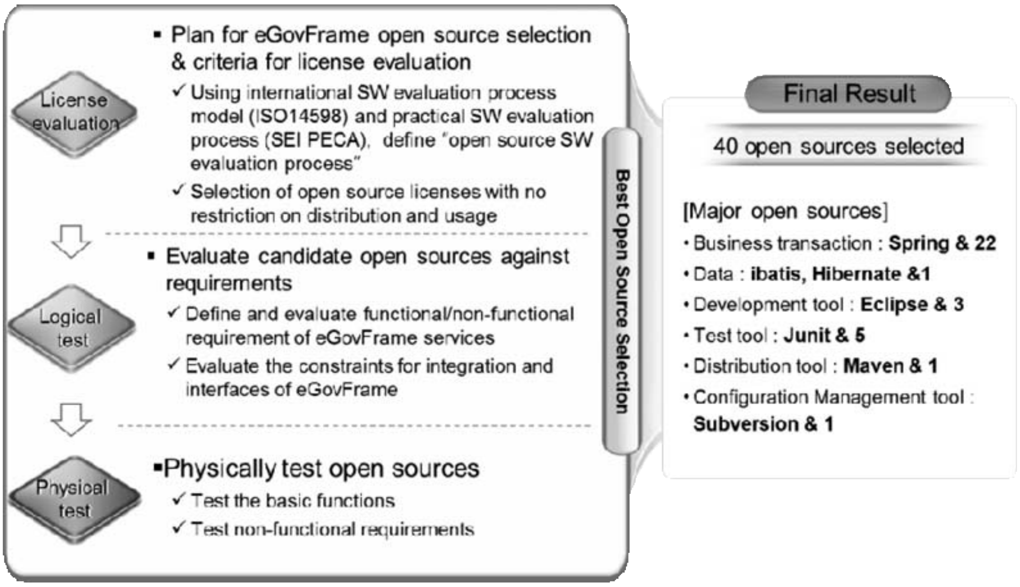

License evaluation
▪ Plan for eGovFrame open source selection & criteria for license evaluation
✓ Using international SW evaluation process model (ISO14598) and practical SW evaluation process (SEI PECA), define "open source SW evaluation process"
✓ Selection of open source licenses with no restriction on distribution and usage
Logical test
▪ Evaluate candidate open sources against requirements
✓ Define and evaluate functional/non-functional requirement of eGovFrame services
✓ Evaluate the constraints for integration and interfaces of eGovFrame
Physical test
▪Physically test open sources
✓ Test the basic functions
✓ Test non-functional requirements
Best Open Source Selection
Final Result
40 open sources selected
[Major open sources]
• Business transaction : Spring & 22
• Data : ibatis, Hibernate &1
• Development tool : Eclipse & 3
• Test tool : Junit & 5
• Distribution tool : Maven & 1
• Configuration Management tool : Subversion & 1

Fig. 3. Evaluation and final selection of open source

● **Open processes**

The development processes are open to public that creates the environment to collect extensive opinions from over 500 stakeholders. Moreover, we took over 20 public-private meetings that lead the understanding and consensus of many stakeholders.

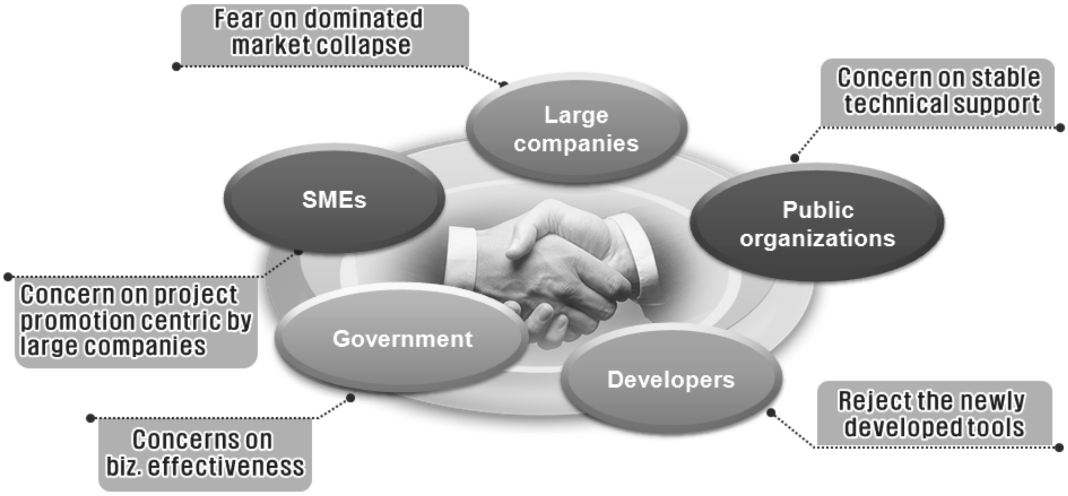

Fear on dominated market collapse
Concern on stable technical support
Large companies
SMEs
Public organizations
Concern on project promotion centric by large companies
Government
Developers
Reject the newly developed tools
Concerns on biz. effectiveness

Fig. 4. Many stakeholders of eGovfram

● **Open outputs**

All of the outputs are open to public such as source code and ER diagram, available on the eGovframe website (www.egovframe.go.kr) that creates the environment to encourage voluntarily participation from developers, providers as well as government officers in the implementation process. Also, we carried out free training courses and 1,236 developers are certified.

- **Open ecosystem**

We founded an open community with large and SMEs and established a public-private cooperation center. They are the central point that promotes eGovframe to global, provides a solid technical support and carries out a continuous improvement. A continuous enhancement of eGovframe is to be carried out by the open community, quarterly experts meeting and open forum of public-private partners. As a result, we establish the open ecosystem for the eGovframe.

Fig. 5. Open forum of public-private partners

3 Overview of eGovflame

This project aims to provide a standardized set of software tools named eGovframe for developing and running e-Government applications in order to improve the efficiency of ICT investment and the quality of e-Government services. It focuses on improving the reusability and interoperability of e-Government applications by setting a standard framework for developing software of e-Government, ensuring the independency from IT companies by adopting open and neutral software tools and enhancing the competitiveness of IT SMEs by sharing the tools openly through the various channels.

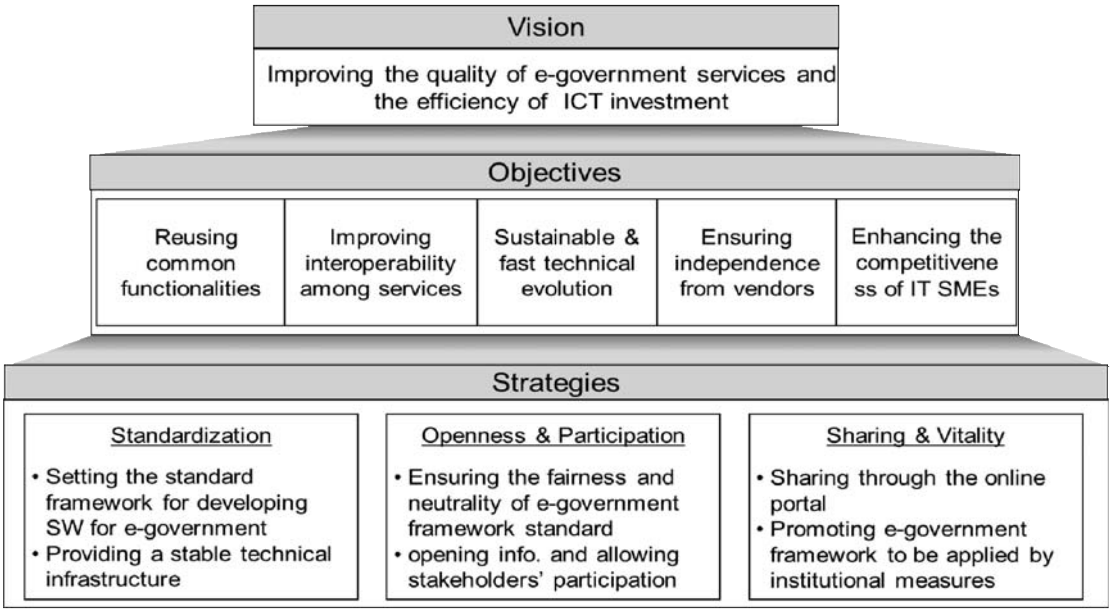

Vision
Improving the quality of e-government services and the efficiency of ICT investment
Objectives
Reusing common functionalities
Improving interoperability among services
Sustainable & fast technical evolution
Ensuring independence from vendors
Enhancing the competitivene ss of IT SMEs
Strategies
Standardization
• Setting the standard framework for developing SW for e-government
• Providing a stable technical infrastructure
Openness & Participation
• Ensuring the fairness and neutrality of e-government framework standard
• opening info. and allowing stakeholders' participation
Sharing & Vitality
• Sharing through the online portal
• Promoting e-government framework to be applied by institutional measures

Fig. 6. Vision and strategies of eGovframe

eGovframe is derived from well-known and proven open source software and whole of the source codes are open through the online portal to every stakeholder. It consists of four software environments: runtime environment for applications, development environment for application developers, management environment for framework managers, and operations environment for application operators.

In the application development stage, about thirty per cent of development costs and efforts can be saved by applying eGovframe. This means eGovframe works as a buffer for adopting various applications to a specific type of infrastructure. It also serves as a common ground for developing common functions.

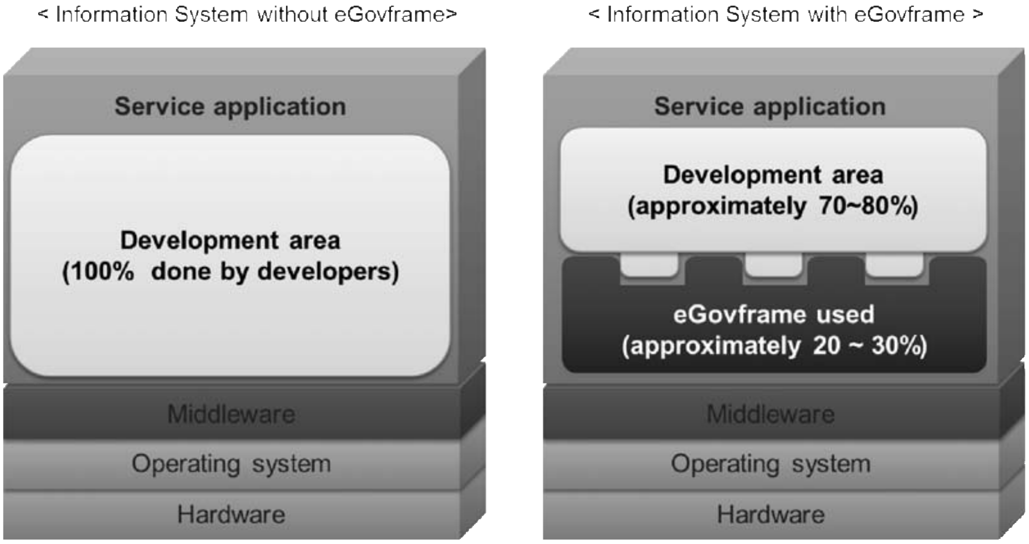
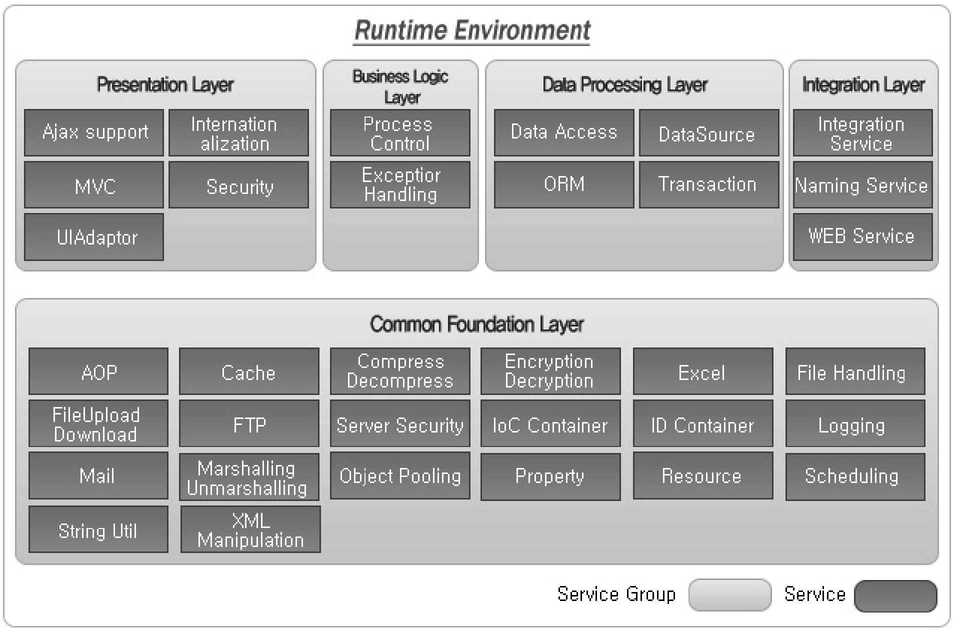


Fig. 7. Vision and strategies of eGovframe

4 Architecture of eGovframe

4.1 Runtime Environment

The Runtime Environment of the eGovernment Standard Framework is composed of 5 service layers and provides 34 services. Refer to the following figure for the details.



4.1.1 Runtime Environment Service Group

4.1.1.1 Presentation Layer

The Presentation layer, as an interface between the business logic layer and users, provides various functions such as user screen construction, user input verification.

- Ajax Support: Ajax is a technique which uses the combination of technologies such as HTML, CSS, DOM, Java script, XML and XSLT to develop interactive web applications. The presentation layer provides Custom Tag Library for Ajax development.
- Internationalization: Internationalization as a service to support various regions and languages, supports automated multilingual function according to the server configuration and the user browsing environment.
- MVC: The presentation layer provides a MVC-based infrastructure for developing user screens applying MVC design patterns.
- Security: This service allows developers to address various web security vulnerabilities(XSS, SQL Injection, etc.) at the development time.

- UI Adaptor: This module provides data type conversion service between the presentation layer and the business logic layer to decouple these two layers.

4.1.1.2 Business Logic Layer

The business logic layer provides various functions such as business logic flow control, transaction management, error processing for application programs.

- Process Control: This service supports the separation between business logic and business logic flow and executes predefined business logic processes. The business logic flow can be defined by external XML files.
- Exception Handling: This service provides standard procedures for runtime exception handling.

4.1.1.3 Data Processing Layer

The Data Processing Layer provides data CRDU functions for the business logic application programs.

- Data Access : Provides functions for excluding dependencies between business logic processing and database solutions or access technologies by providing abstracted access methods for various database solutions and database access technologies.
- Data Source : Provides functions for excluding dependencies between business logic processing and database connection methods by providing various database connection methods as well as an abstract layer about these methods.
- ORM : Supports the development of business logic code not by SQL but by Object by providing the ORM(Object-Relational Mapping) function which is the mapping function between the object model and the database model.
- Transaction : As a service for processing the database transaction, provides a consistent programming model by providing abstracted methods for the transaction processing.

4.1.1.4 Integration Layer

Integration layer supports communication and integration with other systems using standard interfaces.

- Naming Service: This service allows application programs to find remote modules or remote resources.
- Web Service: Using this service, application programs or systems can export their business logic service to other applications or systems through Web Service standard interface.

- Integration Service: This service, providing common interfaces with eGovernment system, provides standardized processing methods to cope with various integration methods of different solutions.

4.1.1.5 Common Foundation Layer

This layer provides commonly used functions for runtime environment services.

- AOP: This service implements and supports ASP(Aspect Oriented Programming).
- Cache: This service provides caching functions for instant access to frequently used contents to minimize access overhead or time.
- Compress/Decompress: This service provides data compression and decompression functions for storing or transmitting data efficiently.
- Encryption/Decryption: This service provides data encryption and decryption functions for secure data transmission.
- Excel: This service provides a Java library for handling Microsoft Excel files.
- File Handling: This service provides file handling functions such as file generation, file access, file update, etc.
- File Upload/Download: This service, mainly used in the presentation layer, provides file upload and download functions.
- FTP: This service provides FTP(File Transfer Protocol) client functions for transmitting and receiving data via FTP protocol.
- ID Generation: This service provides system ID(Identifier) generation function conformant to the UUID(Universal Unique Identifier) standard.
- IoC Container: This service provides the IoC(Inversion of Control) function, one of the basic functions of the framework.
- Logging: This service provides log recording and controlling functions via a simple configuration to minimize the overhead due to casually used System.out.println statements.
- Mail: This service provides SMTP(Simple Mail Transfer Protocol) client functions for transmitting and receiving emails.
- Marshalling/Unmarshalling: This service provides marshalling and unmarshalling functions which convert specific data types into objects or vice versa.
- Object Pooling: This service provides available object(s) to application programs when there is(are) available object(s). If no available object(s), this service generate and allocate the object(s) according to the predefined pool size.
- Property: This service keeps key-value tuples of external files or environment information and provides the access to these tuples for application programs when needed.

- Resource: This service, for Internalization and Localization, provides functions for retrieving appropriate messages in the desired locale using key-value pairs.
- Scheduling: This service supports periodically or repeatedly executed jobs in the application server, similar to the Cron command in UNIX.
- Server Security: This service provides user authentication and rights management functions for secure server function call or data access.
- String Util: This service provides various functions for handling string data.
- XML Manipulation: This service provides various functions for generating, parsing and writing XML data.

4.1.2 Open Source Usage Status

Many services provided by the runtime environment of the framework has been implemented based on open source SWs by reusing or expanding them. Some services, for which no open source SWs satisfied the selection criterion and none was chosen, has been implemented from the ground by the project team.

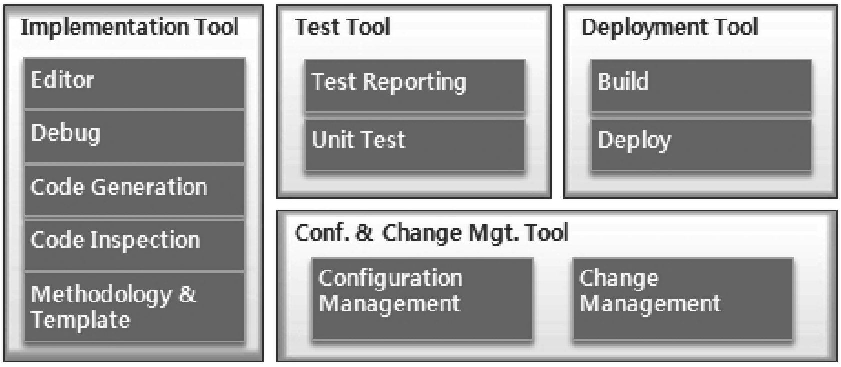
Service Layer	Service	Open Source SW	Expansion or Independent Development	Remarks
Presentation Layer	Ajax Support	Ajax Tags		
	Internationalization	Apache Commons i18n		
	MVC	Spring MVC	Functions including Custom Tag expanded	
	Security	Apache Commons Validator		
	UI Adaptor	None selected		UI Adaptor Interface Manual available
Business Logic Layer	Process Control	Spring Web Flow		
	Exception Handling	Spring	Exception Handling Function expanded	
Data Processing Layer	Data Access	iBatis SQL Maps	Spring-iBatis Function expanded	
	Data Source	Spring		

Service Layer	Service	Open Source SW	Expansion or Independent Development	Remarks
	ORM	Hibernate		
	Transaction	Spring		
Integration Layer	Naming Service Support	Spring		
	Integration Service	None selected	Standard Interface Processing Function developed	
	Web Service Interface	CXF	Web Service Interface	
Common Foundation Layer	AOP	Spring		
	Cache	EHCache		
	Compress/Decompress	Apache Commons Compress		
	Encryption/Decryption	java simplified encryption (jasypt)	Encryption Function expanded	
	Excel	Apache POI	Excel Handling Function expanded	
	File Handling	Jakarta Commons VFS	File Access Function expanded	
	File Upload/Download	Apache Commons FileUpload		
	FTP	Apache Commons Net		
	ID Generation	None selected	Unique System ID Generation Function developed	
	IoC Container	Spring		
	Logging	Log4j		
	Mail	Java Mail		
	Marshalling/Unmarshalling	Castor		
	Object Pooling	Apache		

Service Layer	Service	Open Source SW	Expansion or Independent Development	Remarks
		Commons Pool		
	Property	Spring	Property Function expanded	
	Resource	Spring		
	Scheduling	Quartz		
	Server Security	Spring Security	Authentication and Rights Management Function expanded	
	String Util	Jakarta ORO	String Handling Function expanded	
	XML Manipulation	Apache Xerces 2, JDOM	XML Manipulation Function expanded	

4.2 Development Environment

The Development Environment of the eGovernment Standard Framework is a collection of development tools for making effective use of various functions provided by the Runtime Environment in order to develop more precise and efficient applications. The environment is composed of 4 service toolsets. Refer to the following figure for the details.



4.2.1 Development Environment Toolset

4.2.1.1 Implementation Toolset

This toolset, as a collection of tools for supporting the coding process, is composed of several tools such as Editor, Debug and Methodology & Template.

- Java Editor: This Editor, developed based on the Eclipse, provides its own additional functions for the eGovernment Framework such as Perspective, Menu and New Project Wizard, in addition to the Eclipse's basic functions such as Code Assist, Quick Fix, Loading Code Style, Code Templates, String Search, Quick Type Hierarchy, Quick Outline, Source Code Navigation, Mark Occurrences and Local History.
- DBIO Editor : This Editor, developed appropriate for the projects based on iBatis, provides a GUI for generating SqlMapConfig files and SqlMap files and several convenient functions for testing, checking Query ID duplication and browsing Query ID for minimizing possible errors in query sentence writing.
- UML Editor : This Editor, as an object modeling tool, provides functions for editing the Class Diagram, the Use Case Diagram and the Sequence Diagram. In addition, developers can generate Java Source from the Class Diagram using this editor.
- ERD Editor : This Editor is a data modeling tool for developing the logical data model and the physical data model, and developers can generate schema generation scripts conforming to popular DBMS solutions using this tool.

4.2.1.2 Test Toolset

This toolset, as a collection of tools for testing the written code, is composed of several tools such as Unit Test, Test Automation, Test Coverage and Test Reporting.

- Unit Test : This tool supports test case design based on the Junit. Developers can design test cases using TestCase Generator, a GUI based tool, and also DB unit test cases. This tool supports virtual Mock object too.
- Test Automation : This tool supports automatic execution of the designed test cases. Developers can group test cases into a test suite or can create a test case class composed of test cases, and make them executed automatically through the Build Tool(Maven).
- Test Coverage : Using the open source EMMA, this tool performs an analysis of Test Coverage that the test cases which the developer designed have for the corresponding source code.
- Test Reporting : This tool generates reports on the test result of the test cases designed with JUnit in XML, HTML or Excel format through the

Build Tool(Maven) as well as the test coverage analysis result. Using the XML report file, the CI(Continuous Integration) server of the Deployment Tool displays the test result and the test coverage analysis result on the Dashboard.

4.2.1.3 Build Toolset

This toolset, as a collection of tools for compiling, building and deploying the written code, is composed of Build and Deploy.

- Build : This tool executes building process utilizing Maven as a standard project build management tool. Thanks to the dependency definition and the library management functions which are strong features of the Maven, developers does not need to manage libraries directly but can manage them through the library repository system. Also, the building process can be executed from the Eclipse equipped with the Maven Plugin.
- CI(Continuous Integration) : Utilizing the open source Hudson for the continuous integration and build, the CI server, integrated with the source repository, applies the uploaded source code into the repository and executes the building process. If building fails, this server notifies the related developers of the causes of the failure and allows them to analyze the causes and resolve the problems. This server also executes automatic building process according to the defined period.

4.2.1.4 Configuration Management Toolset

This toolset is composed of Configuration Management for identifying attributes of a configuration item, registering them and managing the changes of them and Change Management for registering and searching configuration issues.

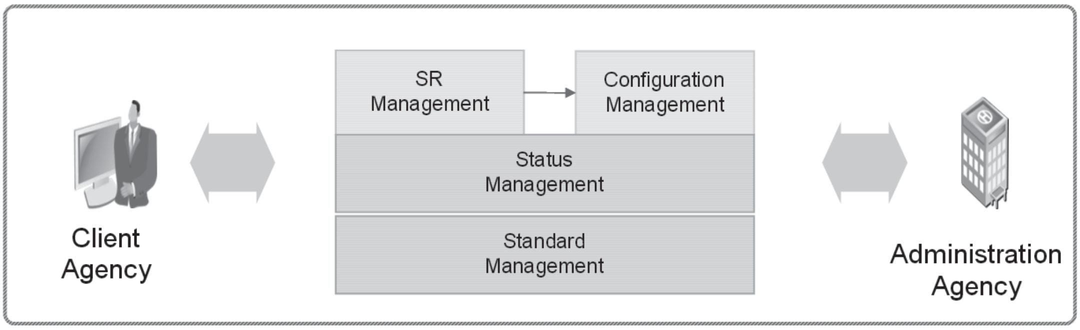
- Configuration Management : On the basis of Subversion, developers can browse the Check-in and Check-out history of the source code and documentations.
- Change Management : The developers can register, resolve and search various configuration issues that happen in the development time. The issues and their processing status can be notified to the related person by the notification function.

4.3 Open Source Usage Status

Service Group	Service		Open Source SW	Remark
Implemtation Tool	Editor	Source Editor	Eclipse	
		UML Editor	AmaterasUML	
		ERD Editor	AmaterasERD	
		DBIO Editor	N/A	developed
	Debug	Local Debug	Eclipse	
	Debug	Remote Debug	Eclipse	
TestTool	Unit Test	Test Case	JUnit	
		Mock Support	EasyMock	
		DB Support	DbUnit	
		Test Coverage	EMMA	
		Test Automation	Ant, Maven	
	Test Reporting		Ant, Maven	expanded
Deployment Tool	Build	Build Tool	Maven	
		CI Server	Hudson	
	Deploy		N/A	developed
Conf.&Change Mgt.Tool	Configuration Management		Subversion	
	Change Management		jTrac	developed

4.4 Management environment

Management environment is composed of Development Management Tool and Operations Management Tool.



4.4.1 Development Management Tool

- Configuration Management : Change and Configuration Management of the Developed Source Code, the Deployment File and the Documentation

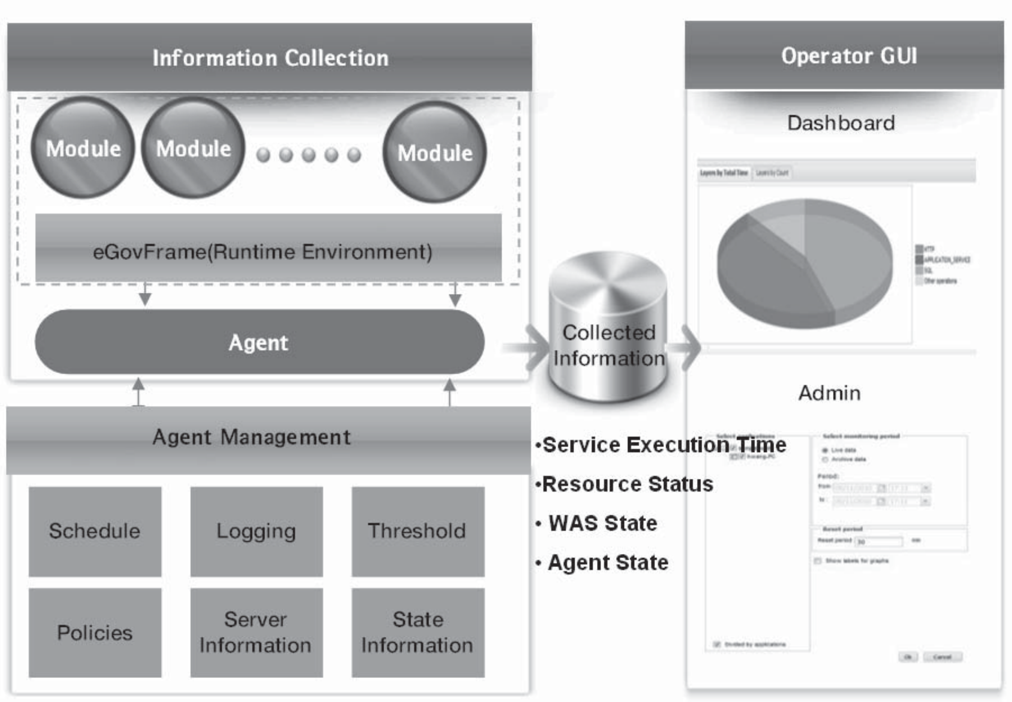
- SR(Service Request) Management : Service Request and Result Feedback

4.4.2 Operations Management Tool

- Status Management : Management of the Framework Deployment and Provided Supports
- Standard Management : Consideration of Version Up or New Functions

4.5 Operations environment

The administrator monitors the system state based on the information, collected and logged by agent programs, on the activities of users and the execution of application program on the Framework.



- Agent management : The agent program runs on the monitored system based on the schedule and logging configuration
- Monitoring data collection : The agent program collects and logs the information on the system and the application program execution
- Operator GUI : Collected information is provided to the operator in various format using graphs or charts

4.6 Common Components

Common Components is a collection of reusable common modules developed to be commonly used in eGovernment Projects.

Download and utilize Common Components from eGovFrame Portal (http://www.egovframe.go.kr).

Component Type		Component
Common Technological Service (129)	Security (8)	services including Authentication, permission administration, encryption/decryption
	User directory/authentication (3)	General Login, login with authentication token, login policy
	User support (56)	services including user administration, inquiry administration, questionnaire administration, FAQ, Q&A
	Collaboration (28)	services including Bulletin board, community, directory
	System administration (25)	services including common code, menu/log administration, institution code
	System/service interface (4)	Services including Institution/Interface administration
	Statistics/reporting (5)	services including Statistics on posting, access, report
Elementary Technological Service (90)		services including calendar, format/calculation/conversion, validity check for format/calculation/conversion

5 Functionality

5.1 Overall benefit:

The main benefits of eGovframe are improving the quality of e-Government services and the efficiency of ICT investment through enhancement of development productivity, component reusability among government systems, interoperability by standardized interfaces, and application software standardization.

- Development productivity: Developer can focus on the business logic since eGovframe provides common modules and standard templates which are typically required to implement a system so that consequently duplicated work can be minimized.

- Component reusability: Increase reusability of the components of the framework by usages with the other projects.
- Interoperability: Using the standard interfaces, interoperability can be increased.
- Application software standardization: By providing a standardized development foundation on presentation, business and persistence layers, standardization of development source code can be achieved.

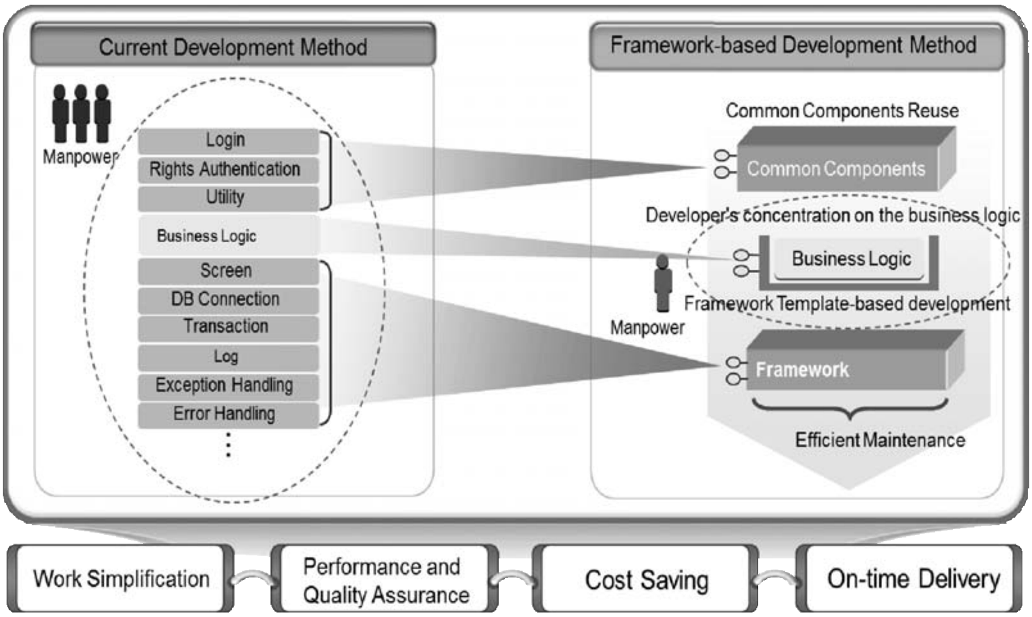


Fig. 8. Utilization of eGovframe

5.2 G2G benefit:

The project ensures interoperability which is necessary for information sharing and inter-linkage among agencies by constructing the application based on eGovframe. This improves interagency collaboration.

5.3 G2C benefit:

The project provides standardized screen and interface for various e-Government services. Easy implementation of inter-linkage among e-Government systems increases levels of public satisfaction by offering one-stop service that was unavailable in the previous system because respective government agencies offered different services.

5.4 G2B benefit:

e-Government implementation providers are now able to improve their core competencies, technological capabilities, and productivity by utilizing eGovframe. The project also promotes strengthening SMEs competitiveness by offering an equal chance of bidding for e-Government project to various companies such as SMEs, solution providers, and venture companies.

6 Productivity

The project performance is measured by the Balance Score Card (BSC) Model that identifies Key Performance Indicators (KPI) reflecting Critical Success Factors (CSF). KPIs have three categories of user, activity, and outcome and for each category KPI is measured through proper survey and number summation methods based on the period of six months or one year depending on its characteristics.

Table 1. BSC of the Project

Component	CSF	KPI
User (acquirer and developer)	✓ User satisfaction level	✓ Perform over thirty education and training sessions ✓ Over seventy per cent of intension to apply eGovframe in public and private sector
Activity	✓ Application of eGovframe ✓ Development of common components	✓ The number of eGovframe used ✓ The number of developed common components
Outcome	✓ Reducing IT budget and cost ✓ Productivity improvement	✓ Reducing effect on IT budget ✓ Improvement rate of development productivity

The achievements of the project are explained by KPIs as follows.

6.1 Performing education and training program for developers:

We have been running eGovframe education and training program for developers since June 2009. As a result, the education and training programme was carried out 41 times and 1,057 developers have been certified so far.

6.2 Survey on intension to apply eGovframe in public and private sector:

According to a survey, over eighty-two per cent of firms responded that they had a plan to introduce software framework within a year.

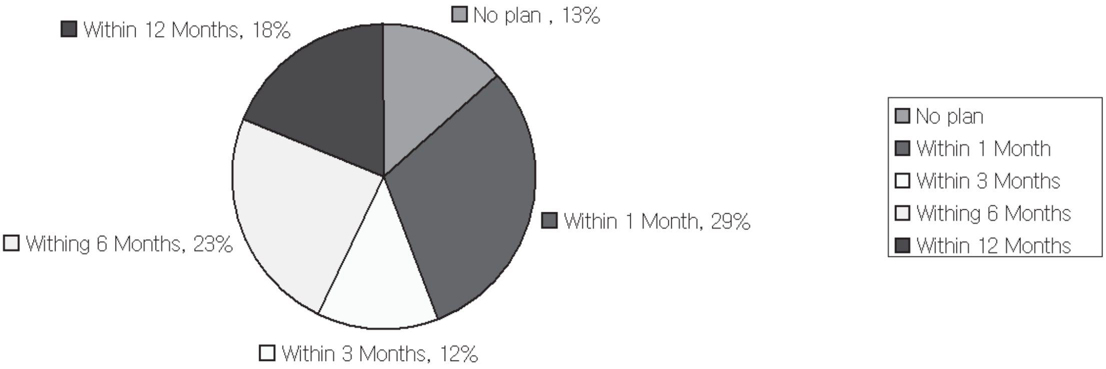


Fig. 9. Survey on intension to apply eGovframe

6.3 Incremental use of eGovframe for building e-Government applications:

eGovframe has so far been applied to seventy-six e-Government projects: national portal systems, e-Authentication, system for sharing administration information, tracing of imported beef distribution channels, advanced civil service systems, local administrative information systems, systems for boosting corporate competitiveness, ubiquitous port systems, and integrated management systems for providing firms with policy information.

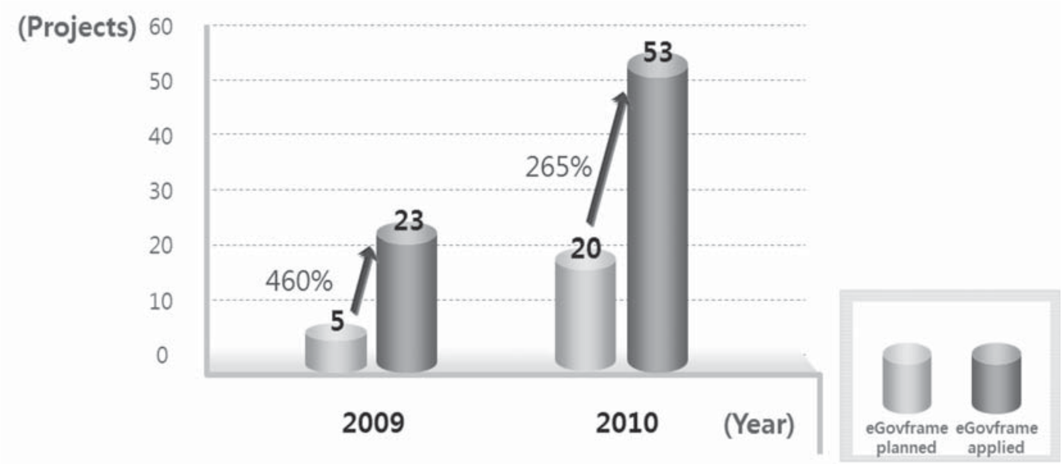


Fig. 10. Incremental use of eGovframe

6.4 Incremental number of common components:

219 common components were defined and developed for one hundred and twenty nine common technological services and ninety elementary technological services.

Table 2. List of common components

Component Type		Component
Common Technological Service (129)	Security (8)	services including Authentication, permission administration, encryption/decryption
	User directory/authentication (3)	General Login, login with authentication token, login policy
	User support (56)	services including user administration, inquiry administration, questionnaire administration, FAQ, Q&A
	Collaboration (28)	services including Bulletin board, community, directory
	System administration (25)	services including common code, menu/log administration, institution code
	System/service interface (4)	services including Institution/Interface administration
	Statistics/reporting (5)	services including Statistics on posting, access, report
Elementary Technological Service (90)		services including calendar, format/calculation/conversion, validity check for format/calculation/conversion

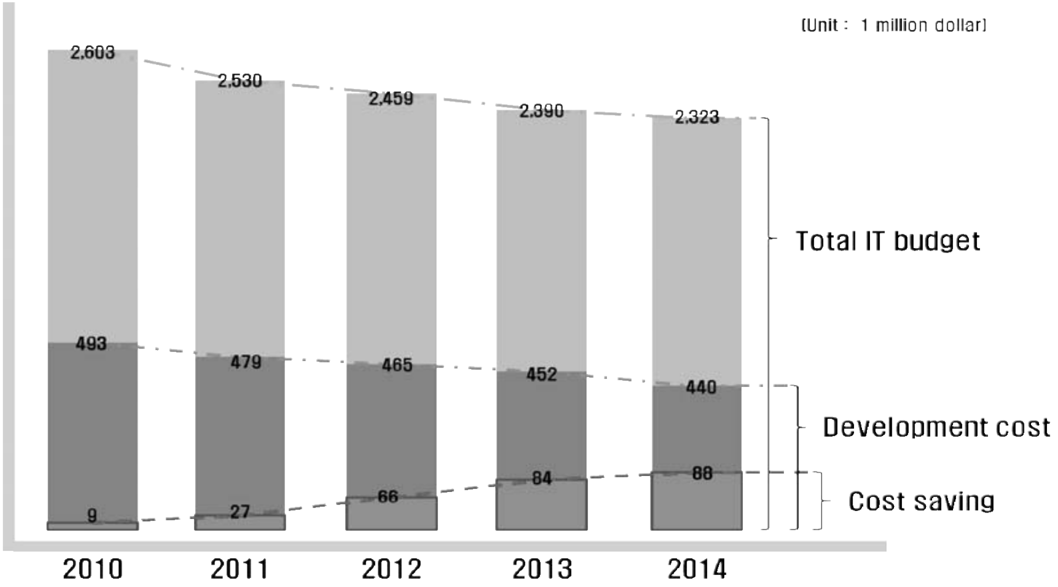
6.5 Return on investment:

By 2014, around twenty three million dollars of budget saving are expected by applying eGovframe to e-Government projects

Table 3. Estimated reduction of budget for information society

(Unit: 1 million dollar)

Section	2010	2011	2012	2013	2014	Total
Total IT Budget	2,603	2,530	2,459	2,390	2,323	12,304
Development Cost	493	479	465	452	440	2,329
Cost saving	9	27	66	84	88	274



6.6 Qualitative improvement:

e-Government services can be standardized and improved in quality, because it is possible to focus on developing task functions by using verified development form and pattern, instead of redeveloping similar or repeated common tasks.

Besides, opening technologies for development framework and providing them free of charge contribute to technology improvement in e-Government and to the elevation of open source software technology worldwide.

7 Conclusions

The Korean government has performed many e-Government projects and made a lot of e-Government applications. For those projects, software frameworks were applied to a considerable part of them because a software framework is a useful tool to increase the productivity and quality for application development. Nowadays software frameworks become a popular tool for developing e-Government applications, but there are some drawbacks originated from software framework. In order to overcome these problems, the Korean government tried to standardize a software framework, eGovframe. However, there were a lot of opinions and issues from many stakeholders, and in order to overcome these issues, we implemented the open innovation strategy with four phases. By the strategy, we achieve the standardization of e-Government framework and establish the open ecosystem of the eGovframe.

With the eGovframe, we formed a fair competition environment for SMEs that made SMEs to win sixty four percent among e-Government projects applied by eGovframe. And we accomplished the diffusion of standard framework to banking and e-business, etc. Also eGovframe has swept three awards at the annual FutureGov Awards in Kota Kinabalu. The E-government Standard Framework won the Government Organization of the Year, the Technology Leadership Award and the award for Government Transformation of the Year.

8 Acknowledge

This research was supported by the ICT Standardization program of MKE(The Ministry of Knowledge Economy)

9 References

[1] Jung-Hyup Kang. E-Government in Korea. Journal of E-Governance, Vol. 33, Number 3. 2010.
[2] The status of Frameworks applied in e-Government projects from 2004 to 2007.
[3] Chesbrough, H.W. Open Innovation: The new imperative for creating and profiting from technology. Boston: Harvard Business School Press. 2003.
[4] Chesbrough, H.W. The era of open innovation. MIT Sloan Management Review, 44 (3), 35-41. 2003.
[5] Wikipedis. Open Innovation. http://en.wikipedia.org/wiki/Open_innovation. 2011.