International Journal of Advanced Smart Convergence Vol.9 No.4 149-155 (2020)
http://dx.doi.org/10.7236/IJASC.2020.9.4.149

IJASC 20-4-19

Best Practice on Software Traceability Environment based on PaaS Cloud Service

Woo Sung Jang[1], Janghwan Kim[2], R. Young Chul Kim[3†]

[1] PhD, Software Engineering Laboratory, Department of Software and Communication Engineering, Hongik University, Korea
[2] M.S., Software Engineering Laboratory, Department of Electronics and Computer Engineering, Hongik University, Korea
[3] Professor, Software Engineering Laboratory, Department of Software and Communication Engineering, Hongik University, Korea
{jang[1], janghwan[2], bob[3]}@selab.hongik.ac.kr

Abstract

In the software industry of Korean Small and Medium-sized Enterprise(SME)s, the development process is often not mature. This may lead to failures in quality control and output management. As a result, the quality of the software can be degraded. To solve the problem, the software visualization technique, which is from the National IT Industry Promotion Agency Software Engineering Center can be applied. We have experienced with mentoring not only the visualization of software development process, but also various visualization process of SMEs. However, the existing software visualization method was difficult to install environment and its time cost was high. This paper proposes a software visualization environment through a cloud service along with a case of building a software visualization environment. We expect that this method will make it easier to build a visualization environment and improve the quality of SME software.

Keywords: *Software Visualization, Cloud, PaaS, Process Visualization, Architecture Visualization, Automatic Document Generation*

1. Introduction

Recently, the roles and functions of software are increasing. Korean large-sized enterprise has excellent software quality management process. However, the software development environment of Korean SMEs finds it difficult to manage software quality because high cost and difficult maintenance of quality management tool [1,2]. Software visualization methods [3] can solve this problem. Software visualization visualizes the whole process from software requirements to development and testing. However, the existing visualization environment is difficult to install. A one-click installer [4] is provided to SMEs, but it may not be installed depending on the computing environment.

We propose a cloud environment-based software visualization method. The advantage of this method is:

Manuscript Received: November. 9, 2020 / Revised: November. 13, 2020 / Accepted: November. 27, 2020
Corresponding Author: bob@selab.hongik.ac.kr
Tel: +82-44-860-2477
Professor, Software Engineering Laboratory, Department of Software and Communication Engineering, Hongik University, Korea

1) Easily check and manage the quality of software through Internet service. 2) The time cost for building the environment is low. 3) Build and use a visualization environment without being limited by the existing computing environment.

The structure of this paper is as follows. Chapter 2 describes related studies. Chapter 3 describes the design of a cloud-based visualization environment. Chapter 4 describes the implementation of a cloud-based visualization environment. Chapter 5 describes the experimental results. Finally, the conclusion is mentioned.

2. Related Studies

2.1 Software Visualization

The software visualization method [2] focuses on visualization of software and automatic documentation. This method minimizes the burden of manpower and cost, and supports software development quality management. The advantage of this method is: 1) Requirements, development, and testing process can be traced. 2) The relationship between requirements, design and source code can be detected during development. 3) Provides a quantitative software quality score measurement environment. As a result, it can relieve the burden of SMEs' software quality management costs.

The software visualization method consists of visualization of the development process, visualization of the source code, documentation of the source code, and documentation of the development process. Each visualization method is not dependent on a specific algorithm or tool. The software visualization method should design and create an optimized solution through analysis of the culture of SMEs.

2.2 Platform as a Service(PaaS)

PaaS is one of the types of cloud-based services. PaaS provides the main infrastructure for software execution. PaaS service consists of server, storage, network and database management system [5]. The use of PaaS reduces the purchasing cost and managing cost of software licenses and basic application infrastructure. PaaS users can buy the necessary resources from cloud service providers, and can access them through a secure internet connection [6].

3. Design of Visualization Environment based on PaaS Cloud Service

3.1 Design of Overall Structure

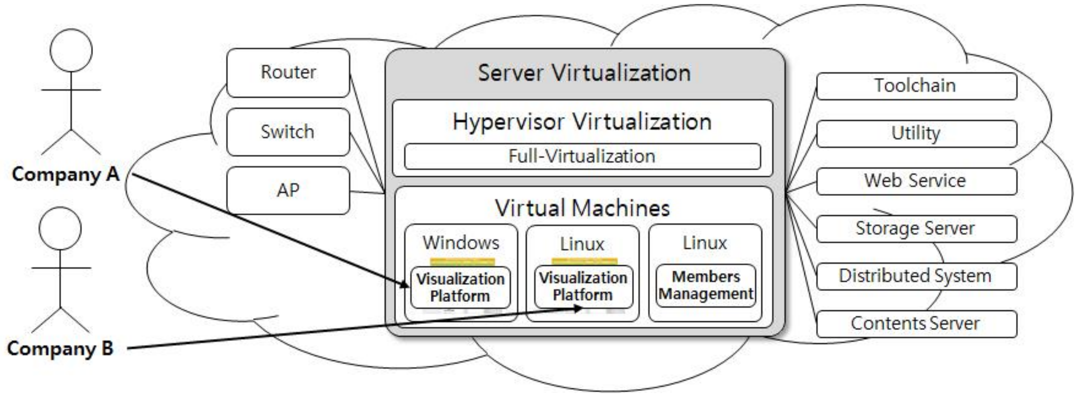


Figure 1. Design of software visualization environment based on PaaS cloud service

We propose a PaaS cloud-based software visualization environment implementation and service method. When a user requests a service provider to provide a visualization environment, the service provider provides a new virtual machine including an operating system and a visualization platform to the user. The detailed structure is shown in Figure 1.

The visualization platform is an environment that includes tools for Process Visualization, Architecture Visualization, and Automatic Documentation. Users can easily visualize a software by using the visualization platform.

3.2 Design of Visualization Platform

Visualization Platform consists of Process Visualization, Architecture Visualization, and Automatic Documentation as shown in Figure 2. Process Visualization visualizes the software development process. This method can easily see the progress of software development. Architecture Visualization visualizes the architecture of software. This method can easily see the quality of the software. Automatic Documentation automatically generates documentation from development work products. This method can reduce the cost for development document generation that developers dislike.

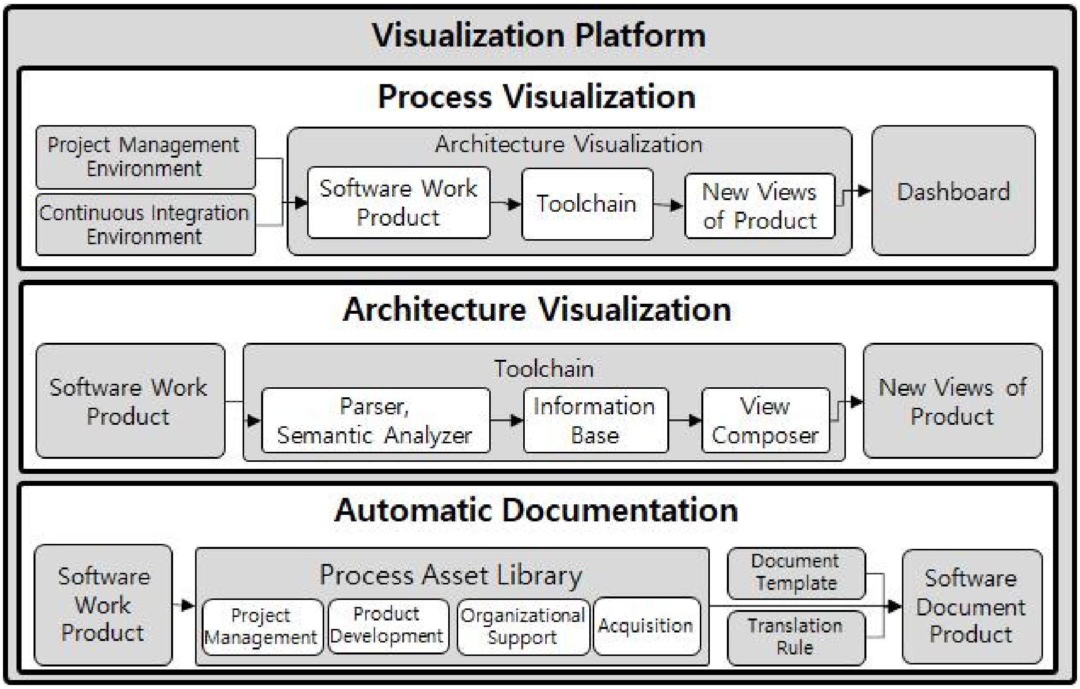


Figure 2. Design of Visualization Platform

The structure of Process Visualization is shown in top of Figure 3. The Project Management Environment stores information about the software development process. Continuous Integration Environment builds the developed source code and manages the source code build version. Architecture Visualization measures the score of software and draws diagrams using Work Product of Project Management Environment and Work Product of Continuous Integration Environment. Dashboard displays the measured score and diagram on the screen.

The structure of Architecture Visualization is shown in middle of Figure 3. Software Work Product is a Work Product of Project Management Environment and Continuous Integration Environment. New Views is a picture of software score or a picture of software structure. Toolchain analyzes Software Work Product and creates New Views. Toolchain consists of Semantic Analyzer (Parser), Information Base, and View Component. Semantic Analyzer formats informal Software Work Product information, and calculates software score using formatted information. Information Base stores formatted information and scores. View Composer visualizes information of Information Base using diagrams.

The structure of Automatic Documentation is shown in bottom of Figure 3. Process Asset Library (PAL) classifies and stores product information into four elements. Document Template is a document template for automatic document generation. Translation Rule is a rule that automatically generates a document by using PAL information and Document Template information. Software Document Product is an automatically generated document.

4. Implements of Visualization Environment based on PaaS Cloud Service

4.1 Implements of Visualization Platform

The implemented Process Visualization is shown in Figure 3. The Project Management Environment was configured by installing the Redmine tool [3]. The Continuous Integration Environment was configured by installing Jenkins tool [3]. Integrated Login Page is implemented for integrated login of Redmine and Jenkins. The work products of Redmine and Jenkins are analyzed by Architecture Visualization, and the analyzed information is displayed on the Dashboard.

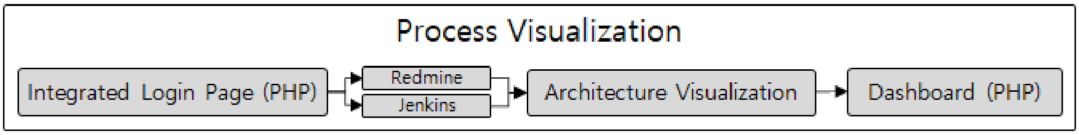


Figure 3. Implements of process visualization

As shown in Figure 4, web-based tools are installed and provided to users. Redmine, Jenkins, Personal Home Page Tools(PHP), and etc. can be replaced by other tools at the request of the user. That is, it can be Plug & Play.

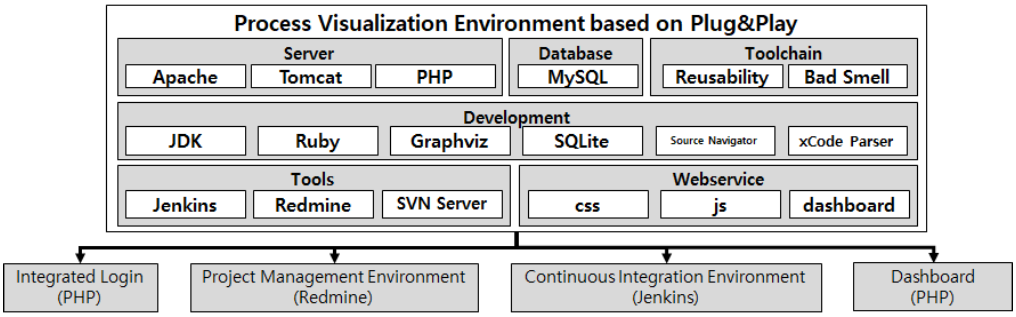


Figure 4. Results of process visualization

The implemented Architecture Visualization is shown in Figure 5. Parser uses Source Navigator. The Information Base uses SQLite (relational database). View Component uses GraphViz. Toolchain measures the score of the software using the source code information analyzed by the Source Navigator, and measures the project progress using the information from Redmine. The types of software scores are Coupling[7], Cohesion[7], Reusability[8], Bad Smell[9], etc. The measured information is stored in SQLite. The information stored in SQLite is automatically converted to DOT script by the toolchain. DOT script is a graph description language. Various graph generation programs support DOT script. The generated DOT script is drawn as a diagram by GraphViz.

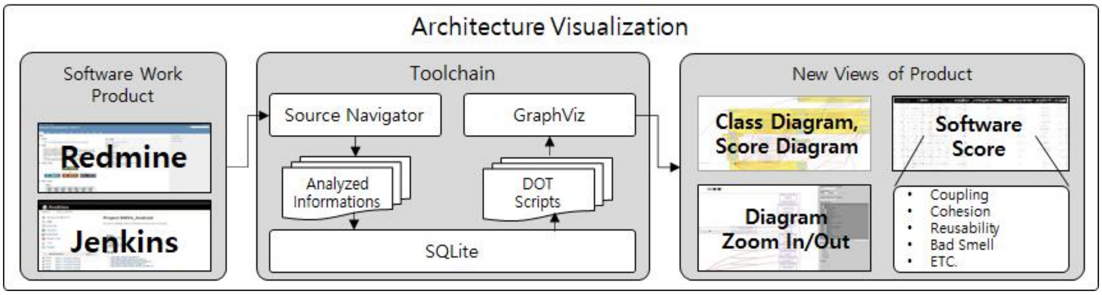


Figure 5. Implements of architecture visualization

Figure 6 is an example of automatic generation of diagrams based on Architecture Visualization. Various diagrams can be automatically generated using the identified information in the source code.

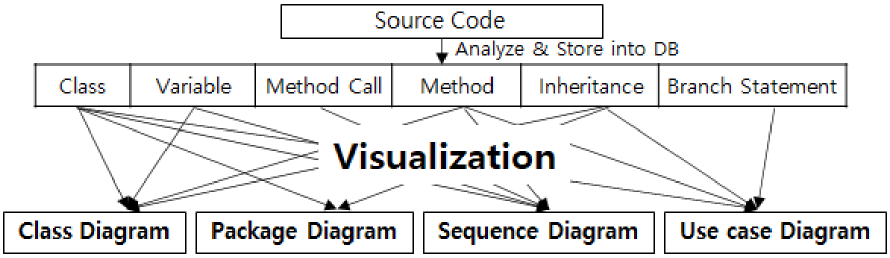


Figure 6. Results of architecture visualization

The implemented Automatic Documentation is shown in Figure 7. The PAL stores information using MySQL. The stored information is automatically converted to an Extensible Markup Language(XML) format file. Document Template and Automatic Document Translation Rule are implemented in Extensible Stylesheet Language Transformations(XSLT). XSLT is a file containing transformation algorithms and Hypertext Markup Language(HTML) templates. Automatically converts XML file information into HTML document.

Redmine, Jenkins information is automatically stored in PAL, and information stored in PAL is automatically converted to XML file. The XML file data is automatically converted into an HTML document by XSLT file.

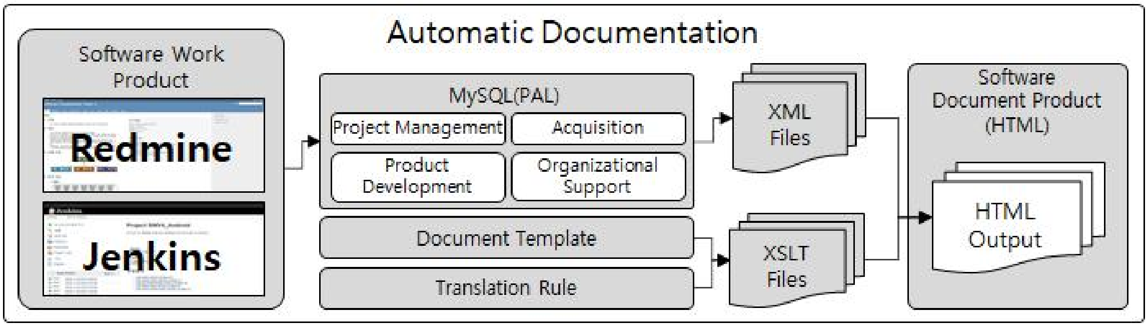
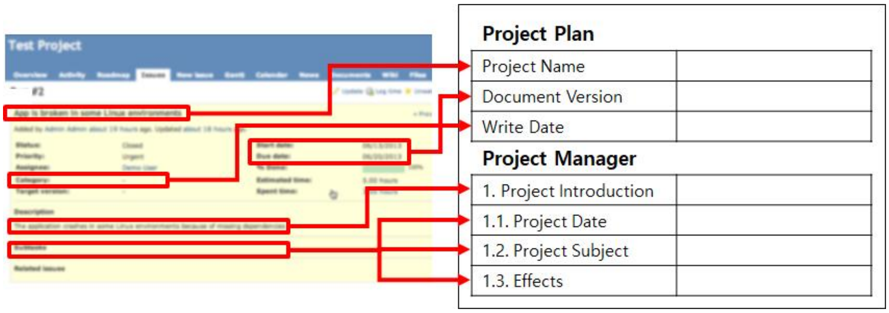


Figure 7. Implements of automatic documentation

If we use automatic documentation, we can easily automatically generate documentation from the requirements management tool, as shown in Figure 8.



Project Plan

Project Name	
Document Version	
Write Date	

Project Manager

1. Project Introduction	
1.1. Project Date	
1.2. Project Subject	
1.3. Effects	

Figure 8. Document Generation using Requirement Management Tool

4.2 Implements of Software Visualization Environment based on PaaS Cloud Service

The structure of the implemented cloud server is shown in Figure 9. The cloud server operating system is ESXi[10]. ESXi is a virtual machine management operating system. The cloud server generates and manages the Management Server virtual machine and the Storage Server virtual machine. The Management Server includes user information and login functions. Storage Server includes toolchain related tools and document templates. Because the router, AP, and firewall functions are paid functions of ESXi, we used physical modules. Because Switch is a free functions of ESXi, we used ESXi's Virtual Switch.

The following is the operation method of the cloud service. 1) A user (company) asks a new visualization environment(Linux OS type or Windows OS type) for a service provider. 2) The service provider is 2-1) Copy the requested type of virtual machine (operating system and visualization platform installed). 2-2) Connect the external port of the virtualization server and the visualization platform service. 2-3) Provide virtual machines to users. 3) Users can access the visualization platform through the external port of the virtualization server. All tools in visualization platform can be serviced through external ports.

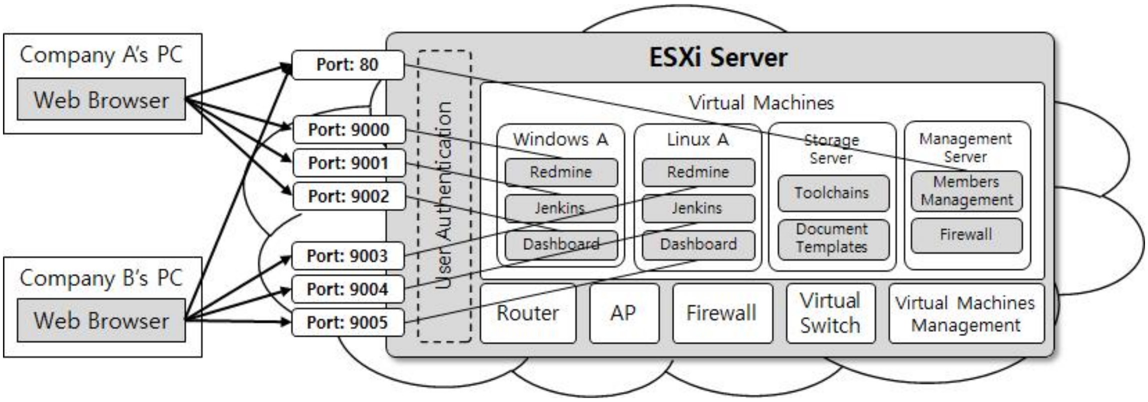


Figure 9. Implements of software visualization environment based on PaaS cloud service

5. Experiment Result

Table 1 shows a comparison of the cloud-based service provision time and the existing installation time. The cloud-based service provision time is within 10 minutes, and the probability of installation failure is 0%.

Table 1. Visualization environment Installation time

	User installs themselves	One-click installer	Cloud service
Installation time	1 week ~1 months	1 hours	10 minutes
Installation failure	Often	Sometimes	None

6. Conclusion

We propose a PaaS cloud-based software visualization environment implementation method. The software visualization environment is a good way to increase the quality of software, but it is difficult to install the environment and the time cost is high. Cloud services can dramatically reduce the time to install a software visualization environment and easily provide a visualization environment to users. The installation time by the user needs 1 week to 1 month, but the installation time using the cloud service requires 10 minutes.

However, the virtual machine-based service provision method consumes a lot of server resources. As a future study, we will study cloud services that consume less server resources.

Acknowledgement

The research was supported by the Ministry of Trade, Industry and Energy (MOTIE), KOREA, through Education Program for Creative and Industrial Convergence (Grant Number N0000717) and supported by the BK21 plus program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.

References

[1] B.K. Park, H.E. Kwon, H.S. Son, Y.S. Kim, S.E. Lee, R.Y.C. Kim, "A Case Study on Improving SW Quality through Software Visualization", Korean Institute of Information Scientists and Engineers, Vol.41, No.11, pp.935-942, Nov 2014.
http://dx.doi.org/10.5626/JOK.2014.41.11.935.

[2] B.K. Park, S.Y. Moon, C.Y. Seo, R.Y.C. Kim, "A Constructive Practice of Software Visualization Mechanism for Code Quality Improvement on the Army's Software Resource Management System", Korean Institute Of Information Scientists and Engineers, pp. 161-163, Dec 2017.

[3] NIPA Software Engineering Center, "SW Development Quality Management Manual", Dec 2013.

[4] W.S. Jang, H.S. Son, R.Y.C. Kim, "A Constructive Practice of the Visualized Development Process of Military Human Resource Management for High Quality Software", Korean Institute Of Information Scientists and Engineers, pp.140-142, Dec 2017.

[5] J.H. Lee, " A Study on Requirement Traceability Matrices based on Automatic Software Process ", Hongik University, Korea, 2018.

[6] S.J. Hur, "A Design and Development of n Large Scale On-Line Game Server Based on PaaS", Journal of the Korea Academia-Industrial cooperation Society, Vol. 9, No. 4, pp. 1006-1011, 2008.

[7] Microsoft Azure, https://azure.microsoft.com/ko-kr/overview/what-is-paas/.

[8] H.E. Kwon, H.S. Son, C.Y. Seo, Y.S. Kim, B.H. Park, R.Y.C. Kim, "A study on Comparing Object Oriented Paradigm with the Cohesion and Coupling mechanism between Traditional modules", Korean Institute Of Information Scientists and Engineers, pp.556-558, 2014.06.

[9] E.Y. Byun, R.Y.C. Kim, "Visualizing Reusable Code patterns based on Object-Oriented Reusability Metrics", 2018 World IT Congress, Feb 2018.

[10] J.H. Park, H.S. Son, R.Y.C. Kim, "Developing an Automatic Tool for Visualizing Source Code against Bad Smell Patterns", Global Conference on Engineering and Applied Science, Jul 2017.

[11] VMware ESXi, https://www.vmware.com/kr/products/.