The Customization of the ISACA's Framework as an Audit Model for Large Scale (Enterprise) Web Applications

Gede Karya, Veronica S. Moertini
Department of Informatics, Faculty of Information Technology and Science, Parahyangan Catholic University
Bandung, Indonesia

Keywords:
Enterprise web audit model
ISACA's Framework
Non-functional requirements
IWTE

ABSTRACT

The applications of information technology, specifically web applications, in the business sector, have led to changes in paradigms, techniques and tools used in performing audits. The applications of large scale of (enterprise) web applications in the business organizations have also been increasing the failure risks that depend on non-functional requirements, such performance (concurrency, response time, capacity/throghput) and reliability. Audit of enterprise web applications become an important part of the implementation process of large scale of web applications in organizations. In this paper, we propose a model to audit enterprise web applications. The model is adopted from Information System Audit and Control Association (ISACA's framework) and software quality model (ISO/IEC 25010) with some customizations. The main contribution of this research is to cover the weakness of ISACA's audit framework, specifically on the lack of attentions to non-functional aspects of large scale web applications. We also propose Integrated Web Test Environment (IWTE) tool that can be used to automate the testing process of non-functional aspects for the web applications.

Corresponding Author:

Gede Karya,
Department of Informatics, Faculty of Information Technology and Science,
Parahyangan Catholic University (UNPAR),
Jalan Ciumbuleuit No. 94, Gedung 9 Lantai 1, Bandung, Indonesia.
Email: gkarya@unpar.ac.id

1. INTRODUCTION

The applications of information technology, specifically web applications, in the business sectors have led to changes in paradigms, techniques and tools used in performing audits. Furthermore, the applications of large scale of (enterprise) web applications in the business organizations have been increasing the failure risks that depend on non-functional requirements, such as capacity, concurrency and performance requirements. Audit is a formal method to obtain reasonable assurance about whether the fulfillment of the rules or the requirements of a web application. Therefore, audits of enterprise web applications become an important part of the implementation process of enterprise web applications in organizations.

A number of researches in the fields of applications audits have been conducted. Some of the results are as follows. Z. Rezaee, et al. (2001) [1] has observed that by the time the e-commerce technologies and Internet were adopted by organizations, there had been some changes in business practices and the process of storing and processing business transactions. Therefore, the auditors need to perform audits continuously, to ensure that the applications performed as expected and to conduct internal controls. Abu Musa (2004) [2] stated that the audit of e-business had been a challenge for external auditors. Then, Junaid M. Shaikh (2005) [3] also discussed that by the time an organization's business relies on information technology, the auditors should apply electronic auditing (EA) framework associated with the technologies adopted. EA implementation is part of the computer-assisted auditing techniques (CAATs).

The implementations of e-business using a web application have prompted auditors to anticipate risks and controls, so as to ensure the safety and effectiveness of the business. S. Kennedy (2005) [4] has identified that there are 10 risks of web applications, which are the risks of: (1) authentication, (2) session security and IDs, (3) SQL injection, (4) buffer overflows, (5) cross-site scripting (XSS), (6) error handling,

(7) remote administration, (8) denial of service, (9) storage and (10) web application testing. Best practices to ``control each risk have been defined. To resolve risk (10), it is suggested to use the integrated test environment to perform testing on a regular basis and whenever changes are made.

For the security aspects, Popa (2009) [5] has discussed how to detect vulnerabilities in the web applications. This activity is important, because the utilizations of complex web applications increase the security risks of business. To deal with the security governance, C. Watson (2009) [6] has proposed methods for implementing Control Objective for Information and related Tehcnology (COBIT) Security Baseline in the web-based business applications. This study also concluded that COBIT can guide the auditors to prioritize the controls which can lead to reducing the security risks of web applications.

In terms of performance and load, Minesce (2002) [7] has proposed a model of web load test using the concept of peak condition emulations by considering users behavior. This model is widely adopted by industries to make their proprietary products. Seng (2009) [8] has stated that some models, which were used in the industrialized world benchmark to measure performance applications such as TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS3AP, OO1, OO7, XOO7, are too specific to particular systems only such that these are not easily applied to general web based applications. Therefore, he has developed "generic construct based workload model", which is derived from the requirements specification.

For web application testing, Sampath (2012) [9] has investigated on how to improve the effectiveness of web application testing using user-session-based. This study concluded that sequencing can reduce the tests suitable to improve the effectiveness of the tests.

In another research, T.W. Singleton (2012) [10][11] has formulated a framework for applications audits consisting of 10 steps, which are: (1) plan the audit, (2) determine the audit objectives, (3) map the data systems and flows, (4) identify key controls, (5) understand application's functionality, (6) perform applicable tests, (7) avoid/consider complications, (8) include financial assertions, (9) consider the beneficial tools, and (10) complete the report.

In our previous works, we have developed a tool to automate the processes of large-scale tests for applications. The tool, which specifically used to test the performances and system functions, adopt the concept of intelligent agents distributed on distributed computer network environment [12, 13, 14]. The tool is an Integrated Web Test Environment (IWTE), which can be access in http://webtest.unpar.ac.id. Here, the load test sub-system model was adopted from [7]. It has implemented user-session-based [9] to improve its effectiveness in performing the tests.

Based on our literature study results, we conclude that so far there has no research results discussing the non-functional aspects in auditing large scale web applications. Therefore, in this paper, we propose a model to audit enterprise web applications. The model is basically adopted from ISACA's[1] framework and from other software quality standards with some customizations. The main contribution of this research is to cover the weakness of ISACA's audit framework, specifically on the lack of attention to non-functional aspects of large scale web applications, such as capacity, concurrency, response time, throughput, and performance requirements. We also enhance the IWTE tool such that it can automate the testing processes applied to non-functional aspects.

2. RESEARCH METHOD

This research aims to develop a model of enterprise web applications audit. We started the research by learning previous similar research results published in 2001 to 2012. Then, we focused on studying the auditing framework which's used by ISACA. We also studied non-functional aspects from software quality model based on ISO/IEC 25010. Having explored and had sufficient understanding of the related concepts, we then proposed an audit model which can be used to audit web applications. We then enhance the IWTE tool by adopting the proposed model and the tool is further validated by applying it in a case study. By the time this paper is being written, we are testing the enhanced IWTE with the case study of academic information system in Parahyangan Catholic University (UNPAR), specifically Student Portal application (sub-system).

3. RESULTS AND ANALYSIS

In this section, we will discuss the research results, which are standards found from literatures that will be adopted, the proposed model and the analysis for validating the model.

3.1. ISACA's Framework in Large Scale Web Applications

[1] In the area of information technology auditing professions, Information System Audit and Control Association (ISACA) is one of the leading associations. ISACA has issued standards and certification in the field of information technology audit.

ISACA Code of Professional Ethics defines 7 codes of conduct, code number 2 states that an auditor should carry out their duties objectively, conduct due diligences and maintain professionalism in accordance with professional standards. Therefore, all CISA certified auditors are required to meet professional standards, which is the Standards, Guidelines and Tools and Techniques of ISACA. It is also stated in [15] that the standards define mandatory requirements, where as the guidelines provide guidelines of how to implement these standards. Tools and techniques provide information on how to meet the standards at the time of audit and assurance work. All ISACA's standards, guidance, tools and techniques were codified as Information Technology Assurance Framework (ITAF) [16].

In the case of standards, audits of large scale web applications associate with standard S6 Performance of Audit Work, Reporting S7, S14 Audit Evidence. In the case of guidance, it is depicted in G2 Audit Evidence Requirement, G3 Use of Computer Assisted Audit Techniques (CAATs), Audit Documentation G8, G10 and G20 Reporting Audit Sampling.

S6 Performance of Audit Work states that to perform the audits, there are three requirements: (1) **Supervision**, information systems audit staff should be supervised to obtain reasonable assurance that audit objectives achieved by meeting professional standards; (2) **Evidence**, to carry out the audit, the auditor should obtain sufficient, reliable and relevant evidences to achieve the audit objectives; audit findings and conclusions must be supported by appropriate analysis and interpretation of the evidences; (3) **Documentation**, the audit process should be documented, and the document should describe the audit works performed and the audit evidences that support the findings and conclusions.

In performing the audit work, there is a guide to the use of CAATs (G3) where it states that the CAATs can be used to perform audit procedures such as: (1) Testing details of transactions and balances; (2) analytical review procedures; (3) ommon control compliance tests; (4) compliance test of application controls; (5) penetration testing. In this context, IWTE can be used in a compliance test of the application controls and penetration testings.

The things that need to be reported are described further in the S7 Reporting. Guidances of how to make an audit reports are described in G20 Reporting. S14 Audit Evidence obliges the auditor to: (1) Obtain sufficient appropriate evidences to draw reasonable conclusions to support the audit results. These evidences include the procedures performed, work results, source documents (either in electronic or paper format), records and other information to support the audit. In addition, the findings and results should demonstrate that they are comply with the laws, regulations and policies that apply to the organization being audited (the auditee). (2) Evaluating the adequacy of the audit evidences obtained during the audit. The evaluation of the evidences can be done through the following processes: inspection, observation, inquiry and confirmation, reperformance, calculation, analysis procedures, and other methods that are commonly used. More on audit evidences are described in the Audit Evidence Requirement G2 and G8 Audit Documentation.

The detailed discussion of audit tools and techniques can be found in ITAF [16]. Some excerpts are as follows. IT Assurance Processes, which is associated with large scale web application audit and particularly in the group of section 3600 IT Audit and Assurance Process and Auditing Application Controls of section 3650. This section describes the process for: (1) Section 3653, traditional auditing application controls, which typically uses a structured approach such as SDLC (System Development Life Cycle) and also includes: (a) authorization input, (b) control and batch balancing, (c) input and editing process, (d) rejecting/ suspended of transactions, (e) batch integrity in online or database system, (f) processing procedures and controls, (g) output control, (h) application access, (i) log management, (j) end-user computing application, and (k) business intelligence. (2) Section 3655, auditing ERP systems. (3) Section 3657, alternative auditing software development strategy, such as portals, web services, service oriented software, UML and other strategies. (4) Section 3660, Specific Audit Requirements, which is divided into related specific criteria applied for the governmental institutions, and specific criteria related to the industries. (4) Section 3670, Audit with CAATs. In implementing the IWTE, the ITAF is associated with this clause, especially for the purposes of system-testing.

In addition, in implementing the section 3610 Using COBIT in IT Assurance Process, it is stated that the COBIT control practices can be adopted to conduct the audit, particularly relating to the limits of coverage and the corresponding situation.

Detailed guidance on **how to audit** the **application programs in general** are described in the Generic Application Audit/Assurance Program [17]. In these guidance, there are 8 activities defined for auditing applications, which are: (1) planning and scoping the audit, (2) planning the audit application, (3) source data preparation and authorization, (4) source data collection and entry, (5) checking the accuracy, completeness and authenticity, (6) processing integrity and validity, (7) output review, reconciliation and error handling, (8) validating the transaction authentication and integrity. Activities no 3-8 are adopted from COBIT's application controls (AC). [17] and [18] also discusses the maturity assessment matrix of 6 ACs.

This maturity measurement was first proposed by Eric Guldentops (2003) [19]. The detail definitions of each maturity level are outlined in [17].

By reviewing the standards, guidelines and techniques used by the ISACA that has been discussed, it can be concluded that ISACA seems to be more focused on the governance and functionality aspects. Non-functional aspects, such as capacity, concurency, scalability and reliability, that reflect the nature of a large-scale (enterprise) web applications are not explicitly discussed. Therefore, to include the non-functional aspects in conducting audits, these will be adopted from ISO/IEC 25010 [20].

3.2. Large Scale Web Application Software Quality Based on ISO/IEC 25010

A large-scale web application is developed based on user requirements. There are two type of requirements, which are functional and non-functional requirements. The degree of fulfilling the functional and non-functional requirements determines the quality of the application soft wares.

Functional requirements focus on **what** can be run by the product [20]. These requirements are made based on business processes that are automated by the application. A functional requirement specification should include: what is the process, who is the user and what data are entered, and what informations are produced, and how to present them.

Non-functional requirements focus on **how** well the function of the software will run [20] and the variables determining this include: (1) **Performance efficiency** [20, 21], such as: concurent user (based on capacity planning), response time (based on the adequacy of operational time), throughput (data and transactions). (2) **Reliability** [20], such as: availability, maturity, fault tolerance, recoverability. (3) **Maintenability**, such as: modularity, reusesability, analysability, modifiability, testability. (4) **Usability** [20], the ease of use of a web application. The factors determining this include the manual which should be easy to learn, the intuitive interaction scenarios and the elements that attracts users in using the application. (5) **Security** [20], related to confidentiality, integrity, non-repuidiation, accountability and authenticity. (6) **Compatibility** (co-existence and interopability). (7) **Portability** (adaptability, installability, replacebility), and (8) **functional suitability** (how does a function meet the needs of a user under specific conditions, does not mean the functional requirements).

Non-functional aspects are **closely linked to large scale web applications** are: (1) performance efficiency, such as: **concurrency, response time, capacity/throghput**, (2) reliability. These aspects can be assessed through **load testing** to ensure that the applications meet the capacity and performance requirements.

3.3. Proposed Model

Based on the discussions in Section 3.1 and 3.2, we propose an audit model for large-scale web application as shown in Figure 1.

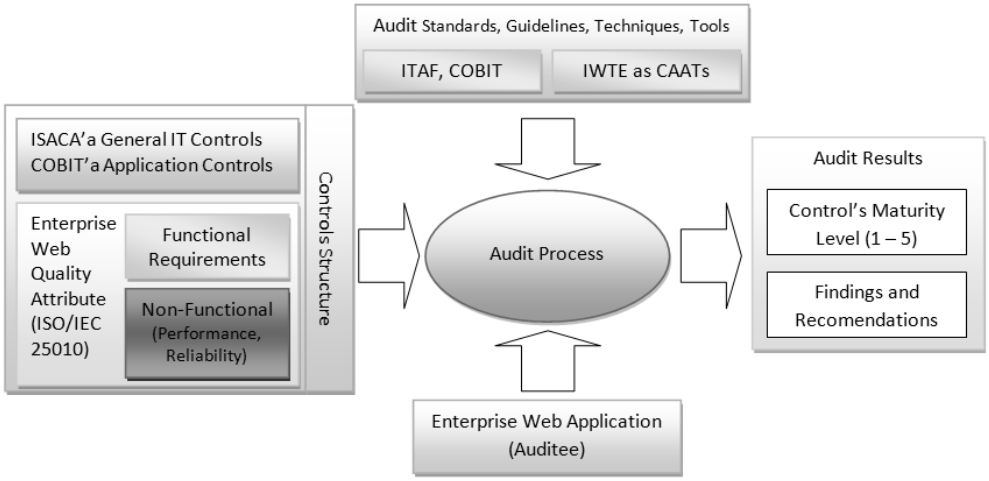

Figure 1. The Web Enterprise Audit Model

There are 3 inputs for the audit process, which are: (1) **Controls structures**, which are all of the enterprise web application controls which will be assesed by the auditor. These controls are derived from

ISACA General IT Controls, COBIT Application Controls, and software quality attributes adopted from the **ISO/IEC 25010**. There are two types of attributes, which are: (a) **functional requirements**, which is based on the business process that are automated by the web applications, and (b) **non-functional requirements**, which are related to the large-scale (enterprise) aspects of the web applications, specifically performance (concurerency, response time, and capacity) and reliability. (2) **Audit standards, guidance, techniques and tools**, which are based on the ITAF and COBIT framework, and IWTE as CAATs tools. (3) **Enterprise web application as auditee**, which is the application to be audited. The audit will be performed with the phases depicted in Figure 2.

The results of the audit process is stated as **audit results**. There are 2 types of audit results, which are: (1) **control's maturity level**, which is based on 6 COBIT's application controls described in Section 3.1, and (2) **findings and recomendations**, which will be provided based on the results of the assesment in fulfilling the functional and non-functional requirements (as the evidences of the controls structures measurements).

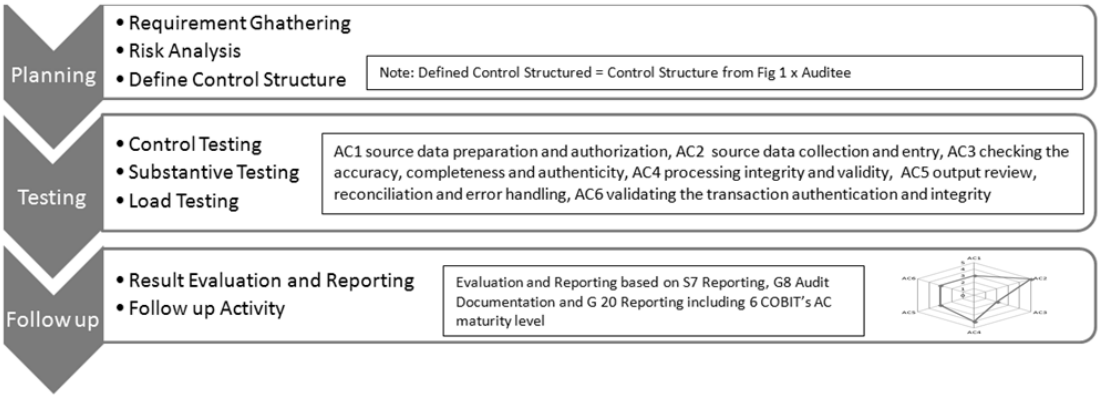

Figure 2. The phases of audit processes

Figure 2 shows that there are 3 phases in performing the audit, which are: (1) **Planning**, this phase covers activities that include: identification of requirements, risk analysis and scoping based on control structure and auditee. This phase produces the control structure to be audited. (2) **Testing**, includes: the process of gathering facts/proofs/evidences, and verifying the compliance, reliability and adequacy of the controls, as well as the substantive testing. In this phase, we can use the IWTE as CAATs to automate the substantive and load testing, such as performance and reliability test. (3) **Follow-up**, includes: evaluating the audit results and writing the recommendations and reporting, as well as reviewing the audit results with the auditee. All of the activities performed must comply with the standards and guidelines from ISACA's audit framework.

The highlights of this model: the non-functional aspects that are important in large scale web applications and which are not yet covered by ISACA's framework are included in the audit process. As has been discussed in Section 2, the IWTE tool has also been developed to adopt the model and is currently under testing.

So, using the proposed model for auditing large scale web applications, ones can ensure that all of the aspects concerning the web applications development have been verified and validated. This will lead to a certain confidence level of the applications that is adequate to ensure the success of large scale web application operations.

4. CONCLUSION

Based on the above discussions, we can conclude, that: (1) ISACA's Framework focus more on the governance aspects and functionality. Non-functional aspects that reflect the nature of large scale (enterprise) are not explicitly discussed. (2) To cover non-functional aspects, such as performance and reliability, which are important for large scale web applications, we adopt **ISO/IEC 25010**. (3) The proposed audit model that is developed by combining ISACA's framework and **ISO/IEC 25010** provides more comprehensive assurance, not only functional and governance aspects, but also non-functional aspects of large scale web applications. For further works, the model is needed to be validated by applying to case studies to ensure that it can be used to audit the web applications appropriately and generate results as expected.

ACKNOWLEDGEMENTS

We would like to thank to the LPPM of Parahyangan Catholic University who has funded this research. Hopefully, the research results will be useful for the society.

REFERENCES

[1] Z. Rezaee, R. Elam, *et al*., "Continuous auditing: The Audit of The Future", *Managerial Auditing Journal*, vol. 16, Iss: 3, pp.150 – 158, 2001.
[2] A. Abu-Musa, "Auditing E-business: New Challenges for External Auditors", *Journal of American Academy of Business*, Cambridge, vol. 4, Iss: 1, pp. 28 – 41, 2004.
[3] J. M. Shaikh, "E-Commerce Impact: emerging technology – electronic auditing", *Managerial Auditing Journal*, vol. 20, Iss: 4, pp.408 – 421, 2005.
[4] S. Kennedy, "Common Web Application Vulnerabilities", *Information System Control Journal*, vol. 4, 2005.
[5] M. Popa, "Detection of the Security Vulnerabilities in Web Applications, *Informatica Economica*, vol. 13, No. 1, pp. 127-136, 2009.
[6] C. Watson, "COBIT Security Baseline Applied to Business Web Applications", *Information System Control Journal*, vol. 4, 2009.
[7] A.D. Minesce., "Load Testing of Web Sites", *IEEE - Internet Computing Journal*, July 2002.
[8] J.L. Seng, I. Ko and B. Lin, "A Generic Construct Based Workload Model for Web Search", *Information Processing & Management*, vol. 45 Issue 5, p529-554, Sept 2009.
[9] S. Sampath and R. C. Bryce, "Improving the Effectiveness of Test Suite Reduction for User-Session-Based Testing of Web Applications", *Information & Software Technology*, vol. 54 Issue 7, p724-738, Jul 2012.
[10] T. W. Singleton, "Auditing Applications Part 1", *Information System Control Journal*, vol. 3, 2012.
[11] T. W. Singleton, "Auditing Applications Part 2", *Information System Control Journal*, vol. 4, 2012.
[12] G. Karya, "Perangkat Lunak Uji Performansi Dan Kapasitas Situs Web Terotomasi Multi Agen", Konferensi Nasional Sistem Informasi (KNSI) 2012, STMIK STIKOM Bali, Denpasar, Feb 2012.
[13] G. Karya, "Automation of Web Enterprise Functional Test Using User Interface-Based Intelligent Agent", *The First International Conference on Computational Science and Information Management*, Parapat-Toba, Indonesia, 3-5 Dec 2012.
[14] G. Karya & Elisati H., "Integrated Web Test Environment", *Research Report*, LPPM – Unpar, 2012.
[15] ISACA, "IT Audit and Assurance Standards and Guidelines", http://www.isaca.org/Knowledge-Center/Standards/Documents/ALL-IT-Standards-Guidelines-and-Tools.pdf, 2013.
[16] R. G. Parker, "ITAF – A Professional Practices Framework for IT Assurance", ISACA, 2008.
[17] N. Kelson, "Generic Application Audit/ Assurance Program", ISACA, 2009.
[18] M. Adler, *et al*., "Cobit 4.1: Framework, Control Objective, Management Guidelines, Maturity Models", IT Governance Institute, 2007.
[19] E. Guldentops, "Maturity Measurement: First the Purpose, Then the Method", *Information System Control Journal*, vol. 4, 2003.
[20] ISO/IEC 25010, "SquaRE – System and Software Quality Models", ISO, 2011.
[21] J. Radatz, *et all.*, "Std. 610.12 Standard Glossary of Software Engineering Terminology" IEEE, 1990.

BIBLIOGRAPHY OF THE AUTHORS

Gede Karya was born in Buleleng, Bali on March 15, 1975. He completed informatics engineering undergraduate in 1997 and master of software engineering in 2002 at Bandung Institut of Technology (ITB). He also hold Certified Information System Auditor (CISA) from ISACA in 2011. Curently, Gede is a lecturer in the Department of Informatics, Parahyangan Catholic University (Unpar) on subjects: Information Systems Auditing, Business Process Analysis, Telematics Application System, and Mobile Programming. He also serve consultancy in the field of information systems, particularly enterprise information system, business intelligence and audit. His research interests include software testing and audit, enterprise application integration (EAI), business intelligence (BI) and telematics.

Veronica S. Moertini is a lecturer in the Informatics Department, Parahyangan Catholic University (Unpar), Bandung, Indonesia. She is also a system analyst and an information system project manager. She completed her doctorate degree from Bandung Institute of Technology in August 2007. She has led teams in developing several medium and large information systems. Her research interests include enterprise information systems, e-commerce, database and data mining. Moertini has published several papers in the international conferences and journals in these research areas.