Service Oriented Architecture Quality Model for Software Security

J.Avesh Gopal[1], DR.P.G.V Suresh Kumar[2], Mohammed Kemal[3], M Sajeeva Reddy[4], Nune Sreenivas[5]

Abstract— The paper presents an approach to locating security aspects in the Service Lifecycle and Service Oriented Architecture (SOA) quality model. The first part of the paper focuses on the quality of SOA and security measures and investigates some functional and non-functional requirements for security measurement. The general discussion about SOA quality and security measures have been summarized by the proposition of the multi-agent architecture for SOA systems security level evaluation in the second part of the paper.

Index terms— Service Oriented Architecture, Software Quality, Software Security.

1 Introduction

Quality of SOA (Service Oriented Architecture) means usually more then only reducing defects. It has to be connected with the requirements of its users, not only for today but into the future as well. When business and IT expectations are mixed, defect-free product is a necessary, but not sufficient. The real challenge with SOA software is in guaranteeing that the application meets the all (business and technological) requirements set out for it. One of such important thing is **security assurance**. SOA is an approach to designing, implementing, and deploying product (service) as a 'puzzle' it is created from the set of components implementing discrete business functions. These components may be distributed across the world but for the user they have to be secure. The problem is how to evaluate and confirm a security level of SOA product.

2. Quality In Service Lifecycle Management

The ability to effectively use of available methods of QA (Quality Assurance) in the lifecycle of services is fundamental to achieving success within SOA. The simplest model of Service Lifecycle may include: governance, delivery, execution and measure (fig. 1).

Nevertheless the main aim for each stage it is important to think about quality analysis and improvement.

- Author name is J Avesh Gopal currently working as a Asst. Professor, Department of Computing, Adama University, Ethiopia. E-mail: avesh-gopal9@gmail.com
- Co-Author name is DR.P.G.V Suresh Kumar currently working as a Professor, Department of ITSC, AAiT, Addis Ababa University, Ethiopia. E-mail: pendemsuresh@gmail.com
- Co-Author name is Mohammed Kemal currently working as a.HOD in Department of computing, Adama University, Ethiopia. E-mail: yekin-99a@yahoo.com.
- Co-Author name is M.Sajeeva Reddy currently working as a. Asst. Professor, in Harshavardan P.G College of Computer Science, Charukupalli, Guntur, AP, India E-mail:sajeeva.reddy@gmail.com
- Co-Author name is Nune Sreenivas currently working as aAsst. Professor, School of Electrical & Computer Engineering, AAiT, Addis Ababa University. E-mail: ns_maruthi@yahoo.com

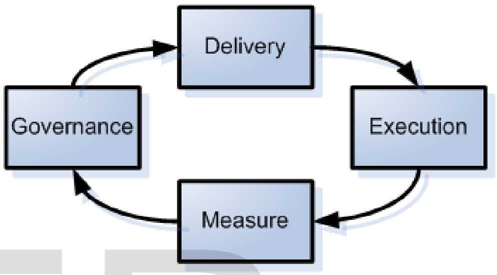

Fig. 1. Typical service lifecycle.

TABLE 1.

Quality Assurance In The Service Lifecycle

phase	key achievement	quality assurance
Governance	1.committing to a strategy for SOA within the overall IT strategy explicitly determining the level of IT and SOA capabilities articulating and refining the vision and strategy for SOA 2. reviewing current governance capabilities and arrangements 3.developing a governance plan	1. reviewing of quality aspects 2. setting the quality expectations (levels) 3. developing a quality assurance plan
Delivery	1. establishing or refining a SOA Center of Excellence (COE) 2. defining addi-	1. defining quality characteristics 2. developing quality metrics 3.deploying rules

	tional capabilities required, such as upgrades to the IT infrastructure 3. agreeing on policies for service reuse across lines of business 4. putting funding mechanisms in place to encourage this reuse 5. establishing mechanisms to guarantee service levels .	of result's interpretation 4. founding procedures for testing 5. establishing system of work's documentation
Execution	1. deploying new and enhanced governance arrangements 2. deploying technology to discover and manage assets 3. communicating and educating expected behaviors and practices within both the business and IT decision-making communities 4. enabling the policy infrastructure 5. executing the service	1. deploying quality assurance model 2. using external tools and application to service quality analysis .
Measure	1. monitoring compliance with policies and governance arrangements, such as service level agreements (SLAs), reuse levels, and change policies 2.analyzing IT effectiveness metrics	1. monitoring and analyzing values of quality metrics 2.communicating within team to improve service quality

.

3 SOA Quality Model And Security Metrics

SOA quality model should address multiple aspects of service quality across SOA service implementations. In fact, two aspects seem to be the most important – software product and business process quality (fig. 2). Both of them determining final quality of the service, which depends on final user expectations and feelings. Companies to ensure top SOA quality of service have to meet customer's business requirements, improve their satisfaction and profitability as well as ensure the highest level of software reliability.

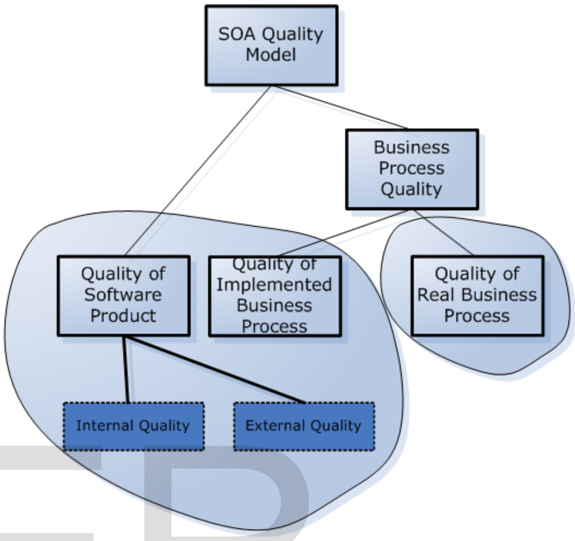
SOA Quality Model
Business Process Quality
Quality of Software Product
Quality of Implemented Business Process
Quality of Real Business Process
Internal Quality
External Quality

Fig. 2 SOA quality model

3.1 Quality of business process

One of the model for business process quality analysis was defined by A.Selcuk Guceglioglu and Onur Demirirs (fig. 3). It presents a complementary process-based approach and focusing on the quality aspect of the process [15]. The structure of the model is based on ISO/IEC 9126, so includes:

— categories (aspects) of quality;

— characteristics (functionality, reliability, usability and maintainability);

— subcharacteristics;

— metrics (to analyze quality attributes)

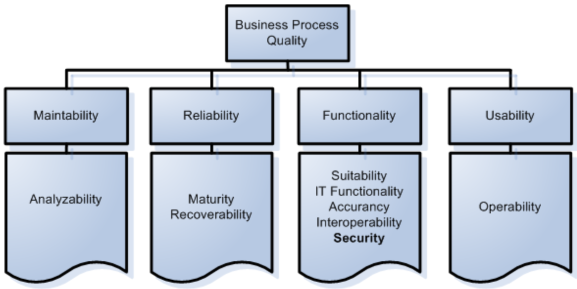

Fig. 3 Model for business process quality

In this model **security** (of the business process) is a part of its functionality and may be measured using *access auditability* metrics.

3.2 Quality of software product

One of the well-known model of standard for description the software product quality is *ISO/IEC9126 Software engineering — Product quality*. It defines six quality characteristics (subdivided into subcharacteristics) for internal and external quality and four characteristics for quality-in-use (fig. 4).

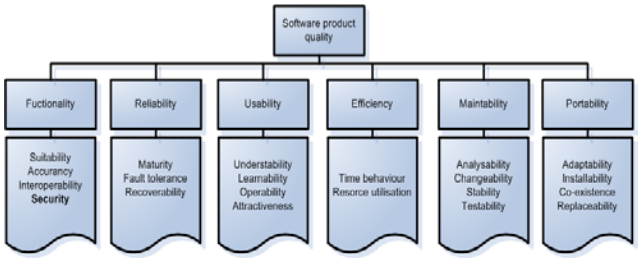

Fig. 4 ISO 9126 model for software product quality

ISO/IEC 9126 may be used to specify and evaluate software product quality from different perspectives. It is dedicated for users linked with analysis of requirements, development, evaluation, maintenance, use or audit software and typical examples of its use are to:

- validate the completeness of a requirements definition;
- identify software requirements;
- identify software design objectives;
- identify software testing objectives;
- identify quality assurance criteria;
- identify acceptance criteria for a completed software product.

Security according to ISO/IEC 9126 means —the capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them‖ [14]. It has several metrics, e.g. *access auditability*, *access controlability*, *data corruption prevention*, *data encryption*.

4 Security Level Evaluation For Software Quality Measurement

The IEEE standard for a Software Quality Metrics Methodology describes software quality as the degree to which software possesses a desired combination of quality attributes [2g]. In this context the crucial element for the quality measurement are quality attributes which are also called quality characteristics. Quality attributes can be classified into two main categories: execution qualities - such as security and usability, which are observable at run time, and evolution qualities - such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system [8]. According to ISO software quality model [3], security is a component of functionality category which is one of the six categories of quality characteristics that has been defined within this model. Definition of security proposed in this document is —the capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them‖. These two documents devoted to software quality measurement are the main set of guidelines to elaborate the security level evaluation framework for SOA systems.

4.1 Security Requirements and Security Measures for Software Development

There are several well known problems and controversies while defining the exact meaning of the security metrics [1,6,12]. There is no one metrics that is acceptable and applicable in context of all possible systems and situations. The security metrics are highly context dependant, so the final shape of the metrics is related to a situation and target depend on security goals, technical, organizational, and operational needs, available resources, etc. At the other hand, metrics are essential in measuring the goodness of target system and it is also true in the context of security quality evaluation, so there is continuous need for security metrics definition.

As a system activity cannot be managed well if it cannot be measured, metrics provide the manager with instruments which enable to characterize, to evaluate, to predict and to improve process execution. Security metrics and measurements can be used for decision support, especially in assessment and prediction. Exemplary security metrics for security assessment include [10]:

- Risk management activities in order to mitigate security risks,
- Comparison of different security controls or solutions,
- Obtaining information about the security posture of an organization, a process or a product,
- Security assurance of a product, an organization, or a process,
- Security testing (functional, red team and penetration testing) of a system,
- Certification and evaluation (e.g. based on Common Criteria) of a product or an organization, and
- Intrusion detection in a system,
- Other reactive security solutions such as antivirus software.

For example, to predict the security behavior of an organization, a process or a product in the future some metrics using mathematical models and algorithms can be applied to collect and analyze measured data (e.g. regression analysis).

A security metric can be qualified as objective or subjective, quantitative or qualitative, dynamic or static, relative or absolute, and direct or indirect [13]. The ISO/IEC 9126 propose external, internal and indirect metrics categories. The internal metrics are understood as static measure products. This metrics measure quality only indirectly. It has also been assumed that at the early stages of development only resources and process can be measured [11]. The external metrics of this standard measure code when during execution. The indirect metrics is a metric of quality in use. According to SOA systems characteristics there are some specific requirements that should be taken into account while considering security measurement.

4.2 Security Assessment in SOA

The security evaluation process should be based on some formal prerequisites. This means that the security evaluation must be objective to guarantee the repeatability and universality of the evaluation results. So, there must be defined notion of the security measure. There are some confusion about this notion. The first problem is that the security measure does not have any specific unit. The other difficulties are: security level has no objective grounding but it only in some way reflects the degree in which our expectation about security agree with reality, security level evaluation is not fully empirical process, etc.

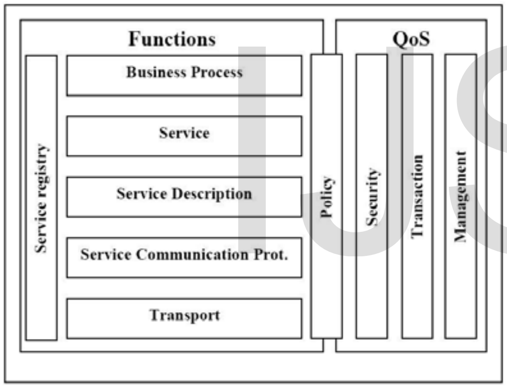

Fig. 5 The layered model of Service Oriented Architecture

As the SOA system can be defined by its five functional layers (fig.5) the correspondent definition of SOA security requirements for security evaluation process should address the specific security problems within each layer. Some elements from a set defining security requirements for the SOA layers has been presented in Table 1. describing functional and non-functional security evaluation requirements for each of the SOA functional layers (selection). The complete list can be found in

TABLE 2

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS FOR SECURITY MEASUREMENT

SOA Layer	Evaluate/verify/test	SOA Layer
Policy and Business Processes	Policy consistency Policy completeness Trust management Identity management	Policy and Business Processes
Service	Identification of the services Authentication of the services Management of security of the complex services	Service
Service Description	Description completeness Availability Protection from attacks	Service Description
ervice Communication Protocol	Confidentiality Authentication Norms compliance	Service Communication Protocol
Transport	Availability Protection from attacks Integrity	Transport

The most important functionality related to the SOA security level evaluation architecture is description of the all components, mechanisms and relations that are necessary to precisely evaluate the security level of the particular SOA system. As it was described in this section the problem of security evaluation is very complex and there exist more than one solution that could be acceptable within a context of a particular system and its environment. We propose some general idea about SOA security level evaluation in a relation to requirements listed in the table 1g. The central part of the proposition is a multi-agent architecture presented in the fig. 2g. The multi-agent architecture is composed of three types of agents: monitoring agents that tests the various security parameters related to particular SOA layer, superior agents that manage the activity of monitoring agents, managing agents that are responsible for all superior agents and for communication with service consumer agents. This type of architecture was selected according to its correspondence to SOA characteristic. Within the architecture monitoring agents are responsible for performing the task related to security assessment using the selected metrics while the managing agent should provide the final results. The next step of the research will be devoted to the problem of selection or elaboration of appropriate for SOA systems security measures for monitoring agents and data fusion methods for managing agent.

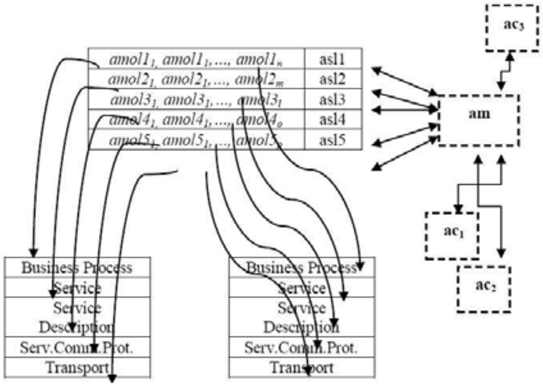

Fig. 6 The architecture of the multi-agent system for SOA security level evaluation

Where:

– AMOL – SOA functional layer monitoring agents

– ASL – SOA functional layer superior agents

– AM – SOA managing agents

– AC – the agents of consumers of SOA system services

5 Conclusions

Improved interoperability is one of the most prominent benefits of SOA. This type of the systems allows service users to transparently call services implemented in disparate platforms using different languages. However, one of the challenges of eliciting quality requirements for a system is that it may not be possible to know all the collaborating parts. This is especially true in SOA-based systems that provide public services and/or search for services at runtime.

A quality measures allows to judge quality of the systems. Quality requirements, such as those for performance, security, modifiability, reliability, and usability, have a significant influence on the software architecture of a system. The use of a service-oriented approach positively impacts some quality attributes, while introducing challenges for others. This paper presented the impact of SOA characteristic on different quality measures. The first part of the paper discussed quality of SOA and security measures and some functional and non-functional requirements for security measurement and then the proposition of the novel multi-agent architecture for SOA systems security level evaluation has been presented in the second part. The future work will concentrate on the evaluation of the proposed architecture for security assessment and refinement of the security measures.

References

[1] ISO/IEC_9126-1, Software engineering — Product quality — Part 1: Quality model. 2001

[2] ISO/IEC_9126-2, Software engineering — Product quality — Part 2: External metrics. 2003.

[5] ISO/IEC_9126-3, Software engineering — Product quality — Part 3: Internal metrics. 2003

[3] Kaner C., Bond W.P., 10th International ―Software Engineering Metrics: What Do They Measure and How Do We Know?‖, Software Metrics Symposium, METRICS 2004

[4] Kolaczek G., Multi-agent Security Evaluation Framework for Service Oriented Architecture Systems, to appear in Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, (2009)

[5] Matinlassi, M. and Niemelä, E. The impact of maintainability on component-based software systems. in Proceedings of 29th Euromicro Conference: New Waves in System Architecture. 2003. Belek-Antalya, Turkey: IEEE Computer Society.

[6] NIST 800-55, Swanson M., Nadya B., Sabato J., Hash J., Graffo L., ―Security Metrics Guide for Information Technology Systems‖, National Institute of Standards and Technology Special Publication #800-26, (2003).

[7] Savola, R. Towards a Security Metrics Taxonomy for the Information and Communication Technology Industry. In Proceedings of the international Conference on Software Engineering Advances (August 25 - 31, 2007). ICSEA. IEEE Computer Society, Washington (2007)

[8] Savolainen, P., Niemela, E., and Savola, R., A Taxonomy of Information Security for Service-Centric Systems. In Proc. of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications EUROMICRO. IEEE Computer Society, Washington, DC, 5-12 (2007)

[9] Vaughn, R.B.J., Henning, R., and Siraj, A. Information assurance measures and metrics - state of practice and proposed taxonomy. in Proceedings of 36th Hawaii International Conference on System Sciences (HICSS03). 2003.