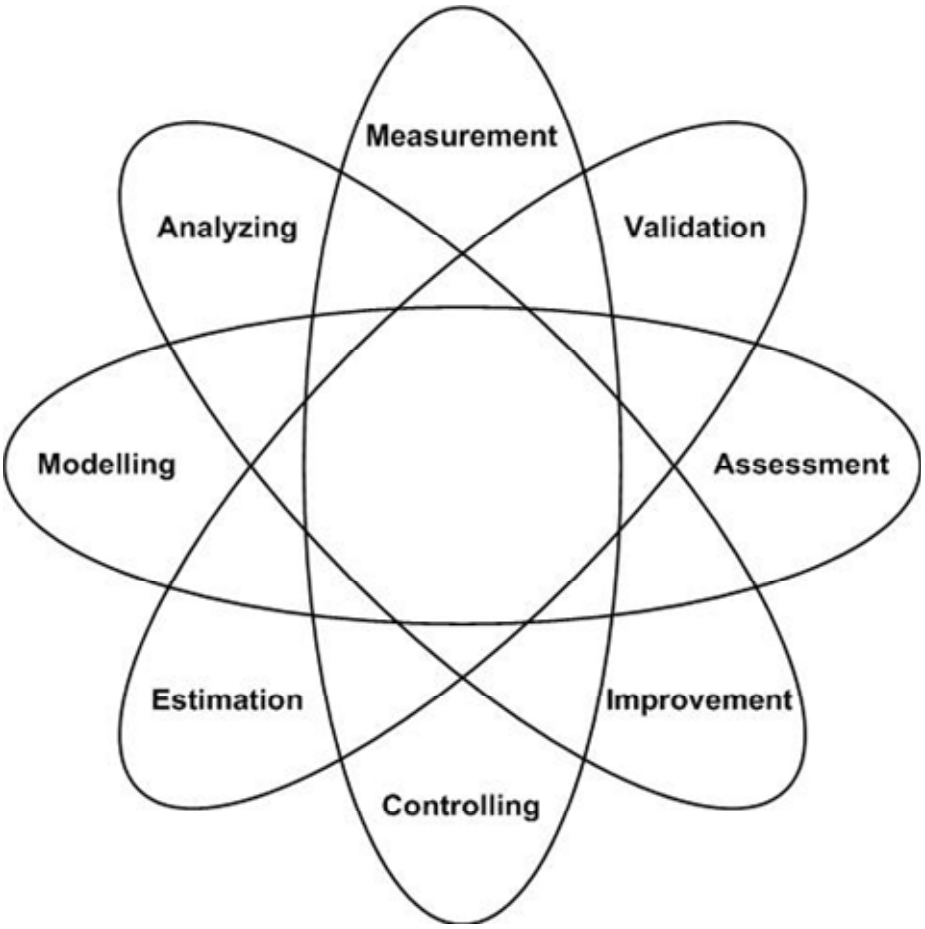
Volume 17, Number 1, March 2012

SOFTWARE MEASUREMENT NEWS

Journal of the Software Metrics Community



Editors:

Alain Abran, Günter Büren, Reiner Dumke, Christof Ebert, Cornelius Wille

The *SOFTWARE MEASUREMENT NEWS* can be ordered directly from the Editorial Office (address can be found below).

Editors:

Alain Abran
Professor and Director of the Research Lab. in Software Engineering Management
École de Technologie Supérieure - ETS
1100 Notre-Dame Quest,
Montréal, Quebec, H3C 1K3, Canada
***Tel.:** +1-514-396-8632, **Fax:** +1-514-396-8684*
aabran@ele.etsmtl.ca

Günter Büren
Vice Chair of the DASMA
Büren & Partner Software-Design GbR
Thurn-und-Taxis-Str. 12, D-90411 Nürnberg, Germany
***Tel.:** +49-911-5195511, **Fax:** +49-911-5195555*
gb@bup-nbg.de
http://www.dasma.org

Reiner Dumke
Professor on Software Engineering
University of Magdeburg, FIN/IVS
Postfach 4120, D-39016 Magdeburg, Germany
***Tel.:** +49-391-67-18664, **Fax:** +49-391-67-12810*
dumke@ivs.cs.uni-magdeburg.de

Christof Ebert
Dr.-Ing. in Computer Science
Vector Consulting GmbH
Ingersheimer Str. 24, D-70499 Stuttgart, Germany
***Tel.:** +49-711-80670-175*
christof.ebert@vector-consulting.de

Cornelius Wille
Professor on Software Engineering
University of Applied Sciences Bingen
Berlinstr. 109, D-55411 Bingen am Rhein, Germany
***Tel.:** +49-6721-409-257, **Fax:** +49-6721-409-158*
wille@fh-b1ingen.de

Editorial Office: Otto-von-Guericke-University of Magdeburg, FIN/IVS, Postfach 4120, 39016 Magdeburg, Germany

Technical Editor: Dagmar Dörge

The journal is published in one volume per year consisting of two numbers.

KONFERENZBERICHT ZUR METRIKON 2011

18.-20. NOVEMBER 2011, FRAUNHOFER IESE KAISERSLAUTERN

Konferenzbeschreibung

Die MetriKon 2011 von der Deutschsprachigen Anwendergruppe für Software-Metrik und Aufwandschätzung (DASMA) gemeinsam mit der ***GI-Fachgruppe 2.1.10 Software-Messung und Bewertung*** und dem Common Software Measurement International Consortium (COSMIC) organisiert und durchgeführt.

Mit der Austragung am Fraunhofer IESE in Kaiserslautern waren wieder ausgezeichnete Konferenzbedingungen gegeben.

Die interessierten Teilnehmer aus Hochschul- und vor allem industriellen Einrichtungen gaben dem Konferenzverlauf eine lebhafte und konstruktive Form der Diskussion und des wissenschaftlichen Meinungs-streites und demonstrierte sehr eindringlich das wachsende Interesse an dieser wichtigen Thematik für den Erfolg bzw. die Verbesserung grundlegender IT-Prozesse sowie der Software-Systementwicklung überhaupt. Der Fokus der Konferenz lag vor allem auf die erfolgreiche Installation von Messprozessen bzw. den vielfältigen Methoden und Messtechnologien zur Unterstützung von Produkt- und Prozessbewertungen, insbesondere für die agile Entwicklung und für intelligente Cloud-Architekturen.

Konferenzinhalt

Andreas Zeller (Uni Saarbrücken): **(Hauptvortrag)** Lernen aus Software

Anja Fiegler (Telekom Magdeburg):
Measurement of favorable characteristics in SOA and Cloud Computing

Thomas Fehlmann (Euro Project AG Zürich):
Understanding Business Drivers for Software Products from Net Promoter Score Serveys

Michael Klaes (BMW München):
Defect Flow Modeling – Fehlermanagement und Verbesserung in der Praxis

Frank Elberzhager (IESE Kaiserslautern):
Using Early Quality Assurance Metrics to Focus Testing Activities

Andrea Herrmann (TU Braunschweig)*:*
Wiederholbarkeit von Projektrisiken

Roland Neumann (Uni Magdeburg):
Vermeidung nur der teuren Fehler: Aufwandsgewichtete Fehlerprognosen

Konstantina Richter (Uni Magdeburg):
Flure Mode and Effect Analysis for the software team capabilities

Stavros Pechlivanidis (IBM Deutschland):
Qualitätsverbesserung von Expertenschätzungen

Thomas Fehlmann (Euro Project AG Zürich):
COSMIC Functional Sizing based on UML Sequence Diagrams

Reiner Dumke (Uni Magdeburg):
Aspect-Oriented vs. Holistic Measurement Frameworks

Michael Stupperich (Daimler AG):
Messen und Bewerten beim Entwickeln von Embedded Software: Erfahrungen aus der industriellen Praxis

Christof Ebert (Vector Consulting Stuttgart):
Estimation Tools – An Overview

Andreas Schmietendorf (HWR Berlin):
ERP-Festpreisprojekte im Kontext einer zunehmend industrialisierten Bereitstellung von IT-Lösungen

Binish Tanveer (FZI Karlsruhe):
Dynamic Identification, Extraction and Reuse of Software Components in Distributed Development Scenarios

Martin Kunz (Bild.de):
Metrikenbbasierter Reviewprozess in der agilen Softwareentwicklung

Martin Kowalczyk (IESE Kaiserslautern):
Alligning Software Porcesses with Organizational Purpose

André Janus (Telekom Stuttgart):
Continuous Integration, Continuous Measurement, Continuous Improvement – Wie Metriken helfen, die interne Qualität in agilen Wartungs- und Weiterentwick-lunsgprojekte sicherzustellen

Wolfgang Kuhl (Cassidian/EADS):
Einsatz von Deployment Metriken im Umfeld des Continuous Delivery Prozesses

DASMA Diplomarbeiten-Preisträger

Tobias Barz (HWR Berlin*):*
Entwicklung einer Methode zur toolgestützten Identifikation von Geschäftsobjekten aus vorhandenen IT-Systemen am Beispiel der Volkswagen AG

Tagungsband

Der Tagungsband ist beim Shaker-Verlag unter dem Titel *Büren et al.: Metrikon 2011 – Praxis der Software-Messung* erschienen.

Reiner Dumke, Stellvtr. Sprecher der GI-FG 2.1.10

Magdeburger Schriften zum Empirischen Software Engineering

Hrsg: Günter Büren, Büren & Partner Software-Design, Nürnberg
Prof. Dr.-Ing. habil. Reiner R. Dumke, Universität Magdeburg
Prof. Dr. Jürgen Münch, Universität Helsinki

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

Fakultät für Informatik
Institut für Verteilte Systeme
Arbeitsgruppe Softwaretechnik

MetriKon 2011

Praxis der Software-Messung

Tagungsband des DASMA Software Metrik Kongresses
17.-18. November 2011, Kaiserslautern

DASMA Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandsschätzung

GI-Fachgruppe 2.1.10
Software Messung und Bewertung

Otto-von-Guericke-Universität Magdeburg
Software Measurement Laboratory (SMLab)

SHAKER VERLAG

Conference Report of IWSM/MENSURA 2011

2.-4. NOVEMBER 2011, NARA, JAPAN

Conference Informations

Our international IWSM/Mensura 2011 conference took place in Nara, Japan, organized from the local Japanese software engineering community in NAIST (Nara) and IPA (Tokyo), the Common Software Measurement International Consortium (COSMIC) and the GI-Fachgruppe 2.1.10 Software-Messung und Bewertung.

The conference location was very excellent and there was more than 200 participants from 15 countries world-wide.

Our conference has facilitate the exchange of software measurement experiences between theory and practice in software measurement, analysis and controlling in different industrial environments and projects..

Conference Presentations

Keynote: *Business Analytics and Optimization in Software Development: Experience at IBM Rational* (Tsutomu Kamimura)

Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed using the Simulink Model
(Hassan Soubra, Alain Abran, Stern Sophie and Amar Ramdane-Cherif)

An Analysis of Gradual Patch Application - A Better Explanation of Patch Acceptance
(Passakorn Phannachitta, Pijak Jirapiwong, Akinori Ihara, Masao Ohira and Kenichi Matsumoto)

Enabling Analysis and Measurement of Conventional Software Development Documents Using Project-specific Formalism
(Taiga Nakamura, Hironori Takeuchi, Futoshi Iwama and Ken Mizuno)

Caching Highly Compute-intensive Cloud Applications: An Approach to Balancing Cost with Performance
(Robert Neumann, Eric Göltzer, Andreas Schmietendorf and Reiner Dumke)

Evaluation of Understandability of UML Class Diagrams by Using Word Similarity
(Yuto Nakamura, Kazunori Sakamoto, Kiyohisa Inoue, Hironori Washizaki and Yoshiaki Fukazawa)

Growth- and Entropy-based SOA Measurement - Vision and Approach in a Large Scale Environment
(Anja Fiegler and Reiner R. Dumke)

Bidirectional Influence of Defects and Functional Size
(Sylvie Trudel and Alain Abran)

Aligning Software Projects with Business Objectives
(Adam Trendowicz, Jens Heidrich and Katsutoshi Shintani)

Evidence-Based Evaluation of Effort Estimation Methods
(Reiner Dumke, Cornelius Wille, Anja Fiegler and Robert Neumann)

Tool-support for a Model-Centric Quality Assessment: QuaTALOG
(Benoît Vanderose and Naji Habra)

Internal and External Software Benchmark Repository Utilization for Effort Estimation
(Ozden Ozcan Top, Baris Ozkan, Mina Nabi and Onur Demirors)

Improve Tracking in the Software Development Projects
(José L. Cuadrado-García, Juan J. Cuadrado-Gallego, Miguel A. Herranz-Martínez and Pablo Rodríguez Soria)

Analyzing Involvements of Reviewers Through Mining A Code Review Repository
(Junwei Liang and Osamu Mizuno)

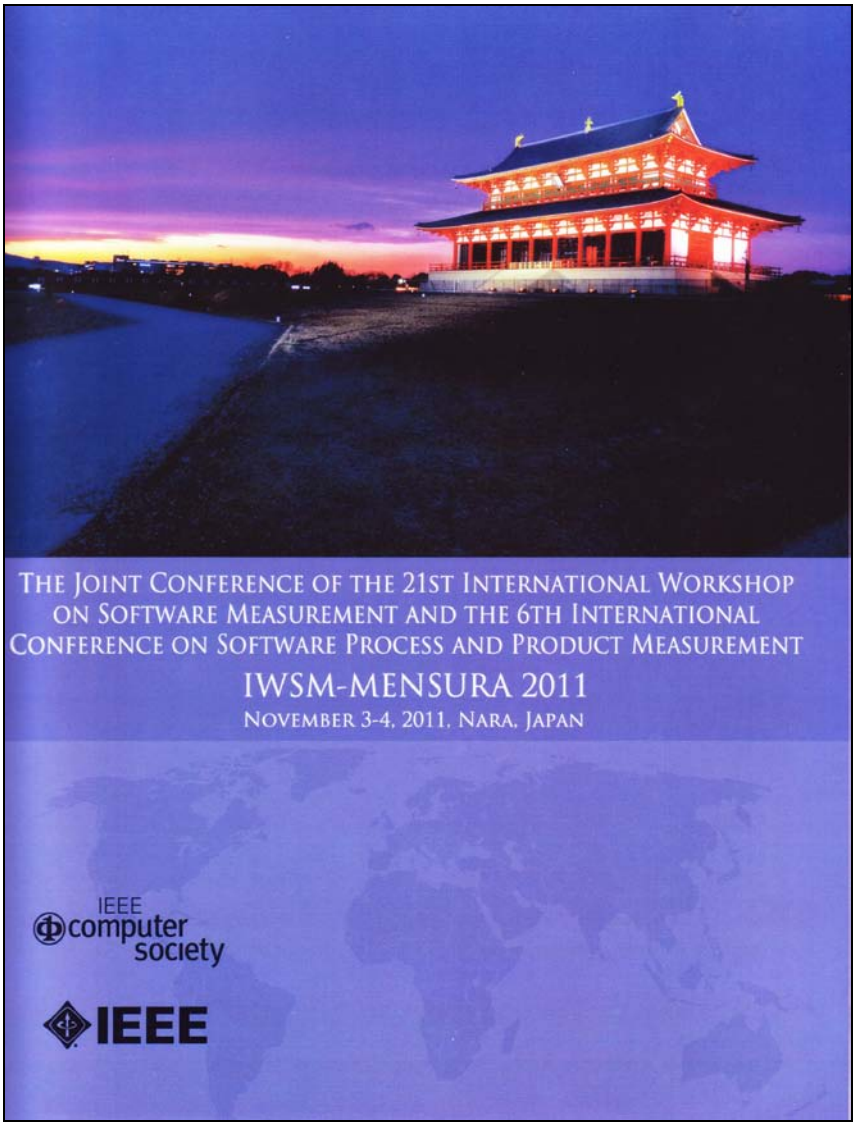
Service Oriented Framework for Mining Software Repository
(Shinsuke Matsumoto and Masahide Nakamura)

and any other more.

Proceedings

The Proceedings are available as online service CPS in the IEEE Xplore

Reiner Dumke, Stellvtr.Sprecher der GI-FG 2.1.10



Bewertungsaspekte serviceorientierter Architekturen (BSOA2011) - detaillierter Workshopbericht

Andreas Schmietendorf[+], *Frank Simon*[#]

[+]Hochschule für Wirtschaft und Recht Berlin
andreas.schmietendorf@hwr-berlin.de

[#]SQS Software Quality Systems AG
Frank.Simon@sqs.de

1. Motivation zum Workshop

Dem Hype der serviceorientierten Architekturen scheint in jüngerer Zeit ein ähnliches Schicksal wie der Titanic zu widerfahren (vgl. Aussage von Anne Thomas Manes – [Manes 2009]): Schon lange untergegangen, aber irgendwie spricht man doch immer wieder darüber. Es stellt sich die Frage nach den Ursachen und den heute benötigten Anforderungen bzw. Veränderungen bei modernen Integrationsarchitekturen:

Ist der „Geist“ der SOA heute unbrauchbar?

Da eine SOA weder als Produkt oder Lösung noch als Technologie verstanden werden kann, entpuppt sich eine allzu technologische Betrachtung häufig bereits als potentielle Ursache gescheiterter SOA-Ansätze. Im Kern geht es der SOA-Idee um die gezielte Restrukturierung bzw. geschäftsorientierte Ausrichtung gewachsener Anwendungslandschaften unter Verwendung interoperabel und lose gekoppelter Fachservices. Eine Aufgabe, die nach wie vor zu den größten Herausforderungen im IT-Umfeld gehört. Dabei handelt es sich aber weniger um eine Projektmanagement-Aufgabe, vielmehr fällt die Verantwortung dafür in ein entsprechendes Programmmanagement. Darüber hinaus existieren im Kontext der aufgezeigten Zielstellung vielfältige Implikationen zum Informations- und Geschäftsprozessmanagement, zum Enterprise Architektur Management oder aber zum betrieblich orientierten IT-Service Management.

Seit 5 Jahren beschäftigt sich die BSOA-Initiative mit der Analyse und Bewertung von serviceorientierten Architekturansätzen. Entsprechend den Zielstellungen des empirischen Software Engineerings geht es der Initiative um den sukzessiven Aufbau qualitativ und quantitativ gesicherter Erfahrungen im Umgang mit derartigen Integrationsarchitekturen. Ein besonderes Interesse gilt dabei der Verwendbarkeit existierender Modellansätze, Methodiken, Prozesse und Techniken bzw. Technologien. Ebenso stellen sich Fragen nach den Einflüssen neuer Entwicklungstendenzen, wie z.B. dem Cloud Computing, der agilen Softwareentwicklung, semantisch orientierten Beschreibungsansätzen wie der Verwendung von Ontologien oder Fragen des Compliance Managements. Im Rahmen des 6. BSOA-Workshops am 15. November 2011 bei der SQS AG in Köln nutzten mehr als 40 Praktiker und Forscher die Möglichkeit zu einem Erfahrungsaustausch.

2. Beiträge des Workshops

Aus den eingesandten Vorschlägen wurden 6 Paper für eine Präsentation während der Workshopsitzungen und 3 für eine Posterdarstellung während der Pausenzeiten ausgewählt.

Darüber hinaus konnte mit Herrn Priv. Doz. Dr.-Ing. habil. Robert Scholderer (TU Ilmenau) ein international erfahrener Referent als Keynote gewonnen werden. In seinem Vortrag ging er auf die besonderen Herausforderungen des Service Level Managements beim industriellen Einsatz von Cloud-Services ein. Im besonderen Maße bedarf es bei Cloud-Lösungen standardisierter SLA-Ansätze, da hier eine Integration verschiedener Anbieter angestrebt wird. Entsprechende Interaktionsketten der beteiligten Lieferanten bedürfen einer differenzierten Betrachtung benötigter Vertragsvereinbarungen, weshalb auch von so genannten SLA-Arten gesprochen wird. In diesem Zusammenhang wird z.B. der Bedarf an generischen SLAs (gSLA) und spezifischen SLAs (sSLA) aufgezeigt. Die gSLA bieten eine standardisierte Klassifikation bzw. einen Ordnungsrahmen benötigter

Vereinbarungen. Bei den sSLA handelt es sich um konkrete Vereinbarungen zu speziellen Cloud-Angeboten. In jedem Fall gilt es, Festlegungen innerhalb von SLAs messbar zu gestalten.

Im Folgenden findet sich eine inhaltliche Zusammenfassung der weiteren, auf dem Workshop gehaltenen Vorträge, die korrespondierenden Artikel finden sich im Tagungsband zum Workshop [Schmietendorf 2011].

Harry M. Sneed: Eine Evolutionsstrategie für Service-orientierte Architekturen

Serviceorientierte Architekturen und die darin enthaltenen Services sind einem ständigen Wandel unterworfen. Aufgrund der Abbildung einer dynamischen Dienstleistungswelt kann eine SOA nie fertig sein. Zur Qualitätssicherung einer derartigen SOA sieht der Autor die Möglichkeit, auf selbst adaptierende Services zurückzugreifen (aktueller Forschungsansatz) bzw. benötigte Veränderungen durch den SOA-Benutzer manuell durchführen zu lassen. Unabhängig von der Art der Serviceanpassung benötigen SOA-Betreiber eine Evolutionsstrategie, die Aufgaben bzw. Dienste und organisatorische Fragen konzeptionell festlegt.

Victor Czenter: Vom SOA-Performancetesten zum Cloud-Performancetesten

Im Mittelpunkt des Beitrags stehen die Möglichkeiten von Performancetests innerhalb einer SOA bzw. die Besonderheiten derartiger Tests im Cloud-Umfeld. Einführend wird eine Übersicht zur benötigten Testinfrastruktur, Lastgeneratoren und einsetzbaren Monitorwerkzeugen gegeben. Die Zielstellungen von Performancetests können sehr unterschiedlich ausfallen und sich z.B. auf die Antwortzeiten und Durchsätze bei einem gegebenen Lastszenario oder aber die Lastgrenzen bzw. die Stabilität des Systems beziehen. Eine besondere Herausforderung im SOA/Cloud-Umfeld stellt die Lastmodellierung bzw. Lastgenerierung dar.

Frederik Kramer, Naoum Jamous: Towards ontology driven information systems – the OEPI example

Die wesentlichen Herausforderungen zur erfolgreichen Implementierung serviceorientierter Lösungen sehen die Autoren bei den beteiligten Menschen bzw. Organisationen in Bezug auf ihre Denkweise und Herangehensweise. In diesem Zusammenhang verweisen Sie auf den Bedarf eines ganzheitlichen Ansatzes und der Verwendung von Ontologien zur Implementierung serviceorientierter Informationssysteme. Auf der Grundlage formal beschriebener Daten und Regeln zu möglichen Beziehungen zwischen diesen sollen semantische Interpretationen und damit die für serviceorientierte Lösungen essentielle Interoperabilität ermöglicht werden.

Anja Fiegler: Bewertung von SOA Qualität anhand wachstums- und entropiebasierter Messmethoden

Die Verwendung wachstums- und entropiebasierter Bewertungsansätze, bekannt aus der Idee wachsende IT-Landschaften mit digitalen Ökosysteme zu vergleichen, wurde hier auf serviceorientierte Architekturen übertragen. Unter Verwendung einer SOA-Referenzarchitektur werden zunächst Hypothesen zur Qualitätsbewertung der involvierten SOA-Ebenen aufgestellt. Nach der theoretischen Erläuterung der verwendeten Mess- und Bewertungsansätze wurden diese innerhalb eines realen Industrieprojekts verwendet. Die dabei erreichten Ergebnisse wurden im Kontext der aufgestellten Hypothesen diskutiert.

David Endler: Release Management von Cloud-Lösungen unter Berücksichtigung unterschiedlicher Infrastrukturen und Architekturen

Cloud-Lösungen implizieren besondere Anforderungen an das Release Management nach ITIL. Die klassischen Anforderungen sieht der Autor im Zusammenhang mit technischen und organisatorischen Sachverhalten. Im Weiteren geht der Beitrag auf unterschiedliche Architekturansätze zur Realisierung von Softwarelösungen ein und vergleicht diese einführend mit Cloud-Architekturen. Die aufgezeigten Architekturansätze sind dann Gegenstand von Komplexitätsabschätzungen, wofür entsprechende Formeln angegeben werden. In der koordinierenden Rolle des Release Management wird eine Möglichkeit zur Komplexitätsbewältigung gesehen.

Florian Muhss: Standardisierung von Cloud Services

Die Vielfältigkeit angebotener Cloud-Services nimmt ständig zu. Sollen diese erfolgreich eingesetzt werden, bedarf es zunehmend grundlegender Festlegungen in Bezug auf funktionale und nicht funktionale Eigenschaften angebotener Servicefunktionen. Mit dem Beitrag geht der Autor auf die speziellen Anforderungen an Cloud-Service Standards im Kontext der Suche,

Bewertung, Bereitstellung, Benutzung und Aussonderung ein. Im Rahmen einer Analyse wird der aktuelle Stand entsprechender Initiativen (z.B. NIST, DMTF) untersucht.

Neben den dargestellten Vollbeiträgen gab es die folgenden Posterpräsentationen, die insbesondere jungen Absolventen und Doktoranden vorbehalten waren:

Marcus Zieger: Möglichkeiten ereignisorientierter Ansätze im Umfeld serviceorientierter Lösungen

Mandy Mälzer: Produkt- und Servicemodellierung mit Hilfe von Ontologien innerhalb der Förderdomäne

Naoum Jamous, Frederik Kramer et al.: Deploying OEPI ontology into the "LWC-EPI" system

3. Ergebnisse der Diskussionsrunde

3.1 Diskussionsthemen

Die Tradition einer moderierten Diskussionsrunde wurde auch beim vergangenen Workshop erfolgreich fortgesetzt. Die dafür angesetzte BoF-Session (Ad hoc–Meeting) stand unter dem Motto „*Quo vadis SOA?*". Zur Anregung potentieller Diskussionen wurden im Vorfeld des Workshops die folgenden Themenbereiche publiziert:

- Agile Vorgehensweise und SOA – ein Oxymoron!?
- Synchronisation zwischen fachlicher und technischer SOA
- SOA ready Services vs. Cloud-Lösungen
- Enterprise Architektur Management und Serviceorientierung
- SOA als Unwort – Bedürfnisse nach neuen Hypes

Darüber hinaus konnten die Teilnehmer eigene Themenvorschläge unterbreiten. Wie bereits im vergangenen Jahr konnte für die Moderation der BoF-Session Herr Dr. Frank Simon (SQS AG) gewonnen werden. Mit der provozierenden These, dass sich SOA und Agilität zueinander wie Feuer und Wasser verhalten, startete er die Diskussion.

3.2 Ausgewählte Ergebnisse

Übereinstimmung gab es bei der Diskussion zu einer SOA-Landkarte (serviceorientierter Bebauungsplan), welche auf keinen Fall agil implementiert werden kann. Im Gegensatz dazu können einzelne Services für eine SOA durchaus agil implementiert werden. In diesem Zusammenhang gilt es festzuhalten, dass es sich bei einer SOA um ein Architekturparadigma handelt. Bei der agilen Vorgehensweise zur Softwareentwicklung geht es um eine leichtgewichtige Prozessgestaltung zur Überwindung einer bürokratischen Softwareentwicklung.

- Eine SOA kann als Enabler für Cloud-Lösungen fungieren
- SOA-Ansätze sind allerdings auch ohne Clouds denkbar
- Eine Cloud kann Laufzeitumgebung für eine SOA sein
- Cloud-Lösungen sind allerdings auch ohne SOA nutzbar

Hinsichtlich des letzten Aspekts entwickelte sich eine Diskussion zur Klassifikation der Cloud-Nutzung. Auf der einen Seite finden sich Endnutzer-zentrierte „ad hoc" Clouds (z.B. der Dropbox-Service), auf der anderen Seite „enterprise" Clouds (z.B. CRM-Anwendungen). Letztere werden eher im Kontext strategischer Überlegungen und damit prozessbezogen zum Einsatz gebracht. Weiterführende Informationen speziell zu diesem Sachverhalt finden sich auch unter [Schmietendorf/Simon 2012].

4. Weitere Informationen

Der Tagungsband wurde innerhalb der Schriftenreihe „Berliner Schriften zu modernen Integrationsarchitekturen" (ISBN 978-3-8440-0503-5) publiziert.


Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Berliner Schriften zu modernen Integrationsarchitekturen

Hrsg.: Andreas Schmietendorf, HWR Berlin
Frank Simon, SQS AG Köln

Hochschule für Wirtschaft und Recht Berlin

Fachbereich II
Wirtschaftsinformatik – Systementwicklung

BSOA 2011

6. Workshop Bewertungsaspekte serviceorientierter Architekturen

15. November 2011, Köln

SHAKER VERLAG

Abbildung 1: Tagungsband zum BSOA-Workshop des Jahres 2011

Weitere Informationen zur BSOA-Initiative, wie z.B. der Call for Paper für den kommenden BSOA-Workshop, finden sich unter folgender URL im Internet:

http://ivs.cs.uni-magdeburg.de/~gi-bsoa

Als Gastgeber für den diesjährigen BSOA-Workshop, am 15. November 2012, konnte die T-Systems Multimedia Solutions GmbH in Dresden gewonnen werden. Informationen zum Tagungsort demnächst auch unter:

http://www.t-systems-mms.com

5. Quellenverzeichnis

[Manes 2009] Manes, A. T.: SOA is Dead – Long Live Services, http://itrepublik.de/business-technology/news/SOA-ist-tot-%96-es-lebe-Service-Orientierung-046793.html, Burton Group

[Schmietendorf 2011] Schmietendorf, A.; Simon, F. (Hrsg.): BSOA 2011 - 6. Workshop Bewertungsaspekte serviceorientierter Architekturen, in Berliner Schriften zu modernen Integrationsarchitekturen, Shaker-Verlag, Aachen – Nov. 2011

[Schmietendorf/Simon 2012] Schmietendorf, A.; Simon, F.: Ad-hoc Clouding: Es gibt keinen Weg zurück!, Publikation der SQS AG Köln – in Vorbereitung 2012

Dank

Ohne vielfältige Unterstützung ist die Durchführung eines solchen Workshops nicht denkbar. Ein besonderer Dank geht an den Gastgeber des diesjährigen Workshops, Herrn Dr. Frank Simon von der SQS AG. Für die organisatorische Unterstützung bedanke ich mich ganz herzlich bei Herrn Dmytro Rud von der InterComponentWare AG und Herrn Henry Frankenberger von der HWR Berlin.

Ein spezieller Dank gilt den Sponsoren, allen voran der SQS AG (Köln), der Software AG (Darmstadt) und der T-Systems International GmbH (Magdeburg). Allen Partnern der BSOA-Initiative (siehe Seite i im Anhang) danke ich gleichfalls für ihr vielfältiges Engagement, insbesondere der ceCMG für die erneute Übernahme der Schirmherrschaft sowie der Hochschule für Wirtschaft und Recht Berlin für die Unterstützung der Tagung.

Bedanken möchte ich mich auch bei Frau Leany Maaßen vom Shaker Verlag Aachen für ihre gewohnt schnelle und unkonventionelle Unterstützung bei der Erstellung dieses Tagungsbandes.

Ein Dank gilt auch allen Mitwirkenden im Programmkomitee und - last but not least – den Autoren, die sich mit ihrem Beitrag an der Agenda des Workshops beteiligen und damit wesentlich zum Gelingen beitragen.

Organisation

Veranstaltet wurde der Workshop in Kooperation zwischen der Hochschule für Wirtschaft und Recht Berlin, dem Forschungszentrum Informatik Karlsruhe und der Otto-von-Guericke-Universität Magdeburg (Softwaremesslabor) unter der Schirmherrschaft der ceCMG (Central Europe Computer Measurement Group). Darüber hinaus erfährt die BSOA-Initiative Unterstützung durch die GI (Gesellschaft für Informatik - Fachgruppe Softwaremessung- und Bewertung), die DASMA (Deutschsprachige Interessensgruppe für Softwaremetrik und Aufwandsschätzung) und durch die ASQF (Arbeitskreis Software-Qualität und –Fortbildung).

An Enhanced Security Approach for Securing and Validating Enterprise Business Processes

Ahmed A. Hussein

College of Computers and Information Sciences, King Saud University, Saudi Arabia

ahussein@ksu.edu.sa

Abstract. *Modern enterprises have their own business activities represented as business processes in an IT environment. The dynamic nature of these processes leads to a problem in providing their security and adaptation. To face this problem we propose a security enhancement approach to guarantee obtaining secured BPEL based on defined enterprise security ontology criteria. The proposed approach introduces a framework that provides an enhanced securely adapted business processes at pre-production phase and checks for their security at production phase. The framework composes of two layers; the first layer is the validation and adaptation of the BPEL security which comes into place at the pre-production phase while the second layer is the Security Enhancement Engine (SEE) that is used at both phases. The main function of the first layer is to validate the securely adapted BPEL against the enterprise security ontology criteria; this is done by providing a security token that is used to represent certain infrastructure quality of service criteria within business processes of an enterprise. A developed algorithm using whirlpool in conjunction with NMACA approach is introduced and used by the second layer to fully secure and validate the BPEL file, its corresponding WSDL files and the token during both phases. We applied our framework to a real case study of a banking environment.*

Keywords: *BPEL, Security Ontology, hash algorithm, Whirlpool, Web Service, Business process.*

1 Introduction

Different security ontology policies are used to fully secure the enterprise internal business processes. Currently, those policies consider the infrastructure; and web services security (WSS). At the message level SOAP extensions are defined to implement, message integrity through XML signature, client authentication and message confidentiality through XML encryption. Such policies provide a basic security that avoids the details of business processes and the relationships within them [5].

The Business processes are the core representation of an efficient, secure implementation and illustration of organization's main services. A BPEL (Business Process Execution Language) is used to represent a business processes schema where detailed descriptions of services are described as web services. It allows, using XML schema standards, defining new internal relationships and security tags. The attributes and the behavior of the web services referenced in BPEL file are presented in an XML standard format(WSDL)Services in which defines the business modules and application functionalities are considered as the main components of SOA. Using standard (XML-based) protocols such services are programmed and implemented based on a self-describing open standards interface.[3, 4, and 7].The combination and invocation of web services through service orchestration is performed by BPM engine executing activities described using the Web Services Business Process Execution Language (WS-BPEL2.0) [8, 9,10,13]

Digital signature and the message authentication code are the major two methods to apply data integrity and authentication. In digital signature method, public key cryptography is used, while in the Message Authentication Code (MAC), a shared secret key is used instead of the private key[14,16].

Our main motivation was to secure the business process within the enterprise and to perform a periodic security check on them to prevent/detect any alteration or modification that could occur.
The rest of this paper is organized as follow. In Section 2, the proposed approach, its layer structure and developed hashing algorithm are described. The applicability of the proposed framework using a

real case study in a banking environment is described in Section 3. Section 4 illustrates the related works. Finally, In Section 5, conclusions and future work are discussed.

2 Proposed Approach

This section introduces the layers of the framework and their functionalities. Figure (1) illustrates the layers' components. The first layer consists of two main engines, the validating and security ontology engines. The inputs to the first layer consist of the *WSS policy* which reflects the standard WSS (Web Service Security) the enterprise imposes. The second input is the interval values of the *Enterprise-based infrastructure security coefficients*, those coefficients are based on the security (QoS) provided by the enterprise infrastructure security components (Firewalls, Wireless security and Intrusion detection prevention system –IDP). The third input is the enterprise BPEL file and its corresponding WSDL files, that are needed to be validated. The function of the first layer is to produce a securely adapted validated or a rejected BPEL and WSDL at the pre-production phase. Figure (2) illustrates a snap-shot of validated WSDL as an output from the first layer where the attribute "wsTokenValue" presented in the BPEL file is reflected in its corresponding WSDL files. Algorithm used to produce the output along with the layers' components and their inputs are illustrated in details in our previous work. [1]

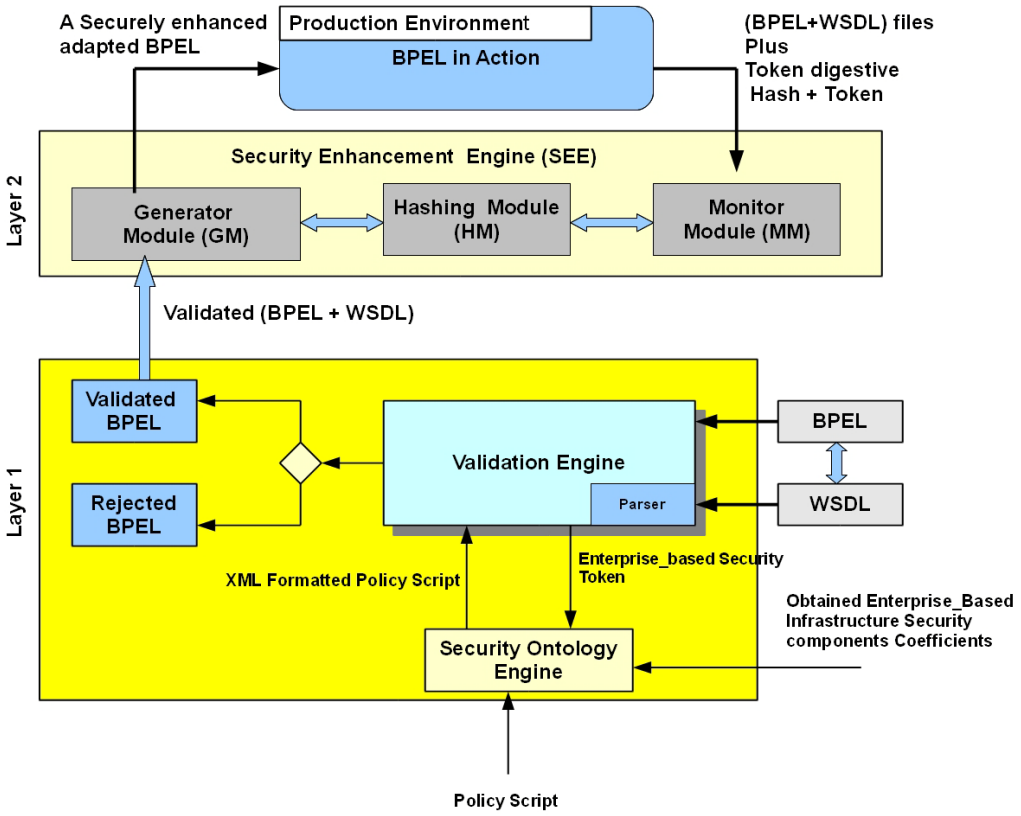


Figure 1. Proposed approach layers

The second layer is the Security Enhancement Engine (SEE) which is designed as a separate layer to provide fully customizable and expandable pool in which can contain any newly introduced security ontology criteria in business process. It is composed of three components; the first one is the Generator Module (GM), the second one is the Monitoring Module (MM) and the third one is the Hashing Module (HM). The input to the second layer is the output from the first layer, and the output of the second layer is a securely enhanced adapted BPEL and its corresponding WSDL files. At the pre-production phase the GM adds two digestive hash values (one is for the token value and one for the file contents) as new tags inside both the BPEL and its corresponding WSDL files as shown in

<DigestiveTokenValue>, <DigestiveFileValue> attributes respectively at figure(2); this hash is generated by a developed fusion hash algorithm provided by the Hash Module (HM), and then the GM produces a securely enhanced adapted files. At the production phase, the Monitoring Module (MM) monitors and verifies that for a specific BPEL file; being called for periodic security checks and/or on-demand security checking; its content or its web services files' contents were not altered or modified. This is done by checking the token digestive hash values inside these files along with checking the digestive hash values for their contents. Figures (3, 4) illustrate the processing for GM & MM at both the pre-production and productions phases.

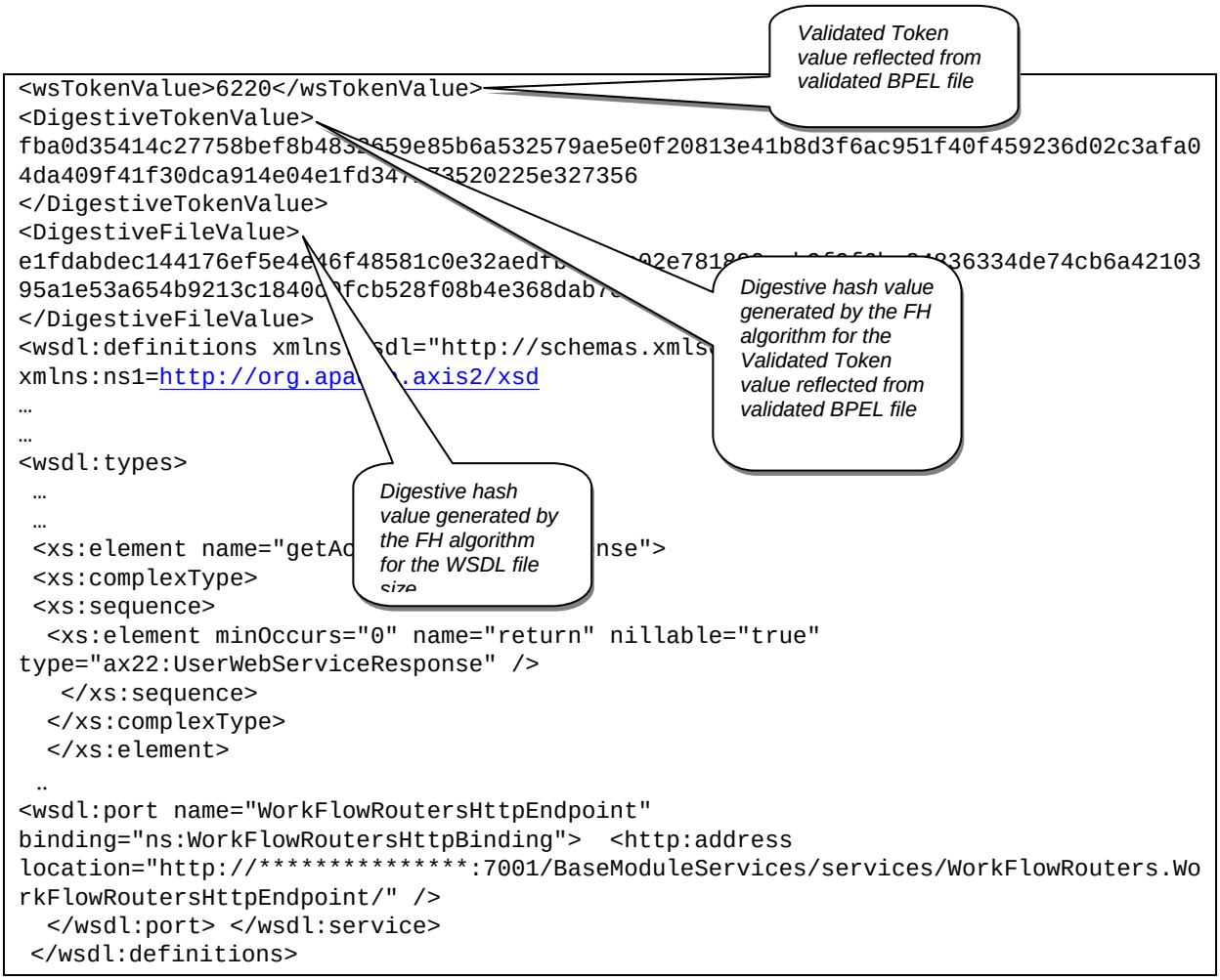


Figure 2. Snapshot of Securely Enhanced adapted WSDL file

The Hashing Module produces the digestive hash for both the token values and files' contents, and provides it to both the GM and the MM at pre-production and production phases respectively. We developed hashing algorithm – Fusion Hashing (FH) – to be used by the Hashing Module (HM) using a combination of the Whirlpool hashing algorithm and the NMACA approach [2]. This will overcome the use of common non-secret key message authentication techniques such as MD5, SHA-1, and Whirlpool where an alteration of both token value, token's digest value and file contents won't be detected by the MM. Therefore; the Fusion Hashing (FH) algorithm detects any alteration either in the token value, digestive hash value, file contents and/or all. The FH algorithm is explained in details below.

Pre-production phase:

#Generator Module

INPUT:
Validated BPEL + Validated WSDL
BEGIN:
Call getHashForToken(Token Value) # Hashing Module
Call addTokenDigestiveValue( )
Call getHashForfilecontent(file) # Hashing Module
Call addfileDigestiveValue( )

//
for example:
<DigestiveTokenValue>
fba0d35414c27758bef8b4832659e85b6a532579ae5e0f20813e41b8d3f6ac951f40f459236d02c3afa04da409f41f
30dca914e04e1fd347173520225e327356</DigestiveTokenValue>
//

//
for example:
<DigestiveFileValue>
e1fdabdec144176ef5e4e46f48581c0e32aedfb7c8ce02e781800acb9f9f6bc84836334de74cb6a4210395a1e53a6
54b9213c1840c9fcb528f08b4e368dab7c2d</DigestiveFileValue>
//

OUTPUT:

An Securely enhanced BPEL & WSDL

END

Figure 3. Steps performed by GM at pre-production phase

production phase:

#Monitoring Module

INPUT:
Token digestive hash & Token
BEGIN:
Call getHashForToken(Token Value) # Hashing Module
Call getHashForFile(File) # Hashing Module
Call CompareTokenagainstitsHash(inuputtokenandhash,generatedhash)

CompareTokenagainstitsHash(inuputtokenandhash,generatedhash)
CompareexistingFilehashagainsgeneratedHash(File,filegeneratedhash)

BEGIN # *CompareTokenagainstitsHash*

If (inputtokenandhash EQUAL generatedhash)

```
            No alteration occurred
        ELSE
            Alteration occurred , pause BPEL at production
        EndIF

END # CompareTokenagainstitsHash

BEGIN  # CompareexistingFilehashagainsgeneratedHash

         If (generatedFilehash EQUAL existinghash)
            No alteration occurred for the file
        ELSE
            Alteration occurred , pause BPEL at production
        EndIF

END # CompareexistingFilehashagainsgeneratedHash
```

Figure 4. Steps performed by MM at production phase

The Fusion Hashing (FH) algorithm: The FH technique is an updated version of the whirlpool technique, where the basic steps of whirlpool are used in conjunction with the NMACA approach. In FH algorithm, a message of arbitrary length and a secret key of 512 bits are used to produce a 512- bit Message Authentication Code (MAC). Based on the features of Whirlpool technique and the additional features of NMACA approach, it is computationally infeasible to have two different messages with the same MAC, or to have any message that has a pre-specified MAC.

In FH algorithm, we use a more restricted approach to build MAC that uses a secret key K to enhance the MAC compression function, where K is involved in all iterations. The use of K in the NMACA approach provides an additional protection to overcome weaknesses that could be arisen in the used hash function. In the Whirlpool algorithmic steps, instead of just using the key as the initial chaining values to the algorithm, K is used to determine the access order for message words and to determine the shift amounts in the distinct rounds. The proposed technique is as robust as the Whirlpool technique, where its performance is slightly slower than that of the Whirlpool technique.

As a start, the Whirlpool technique takes as an input a message with a maximum length of less than 2256 bits and generates a 512-bit message digest. The input is processed in 512-bit blocks as a starting step; the message is padded to produce a message with a multiple of 512 bits that consists of the main message, its length and some padding bits. The padding process is done as follows:

The message is padded by a single 1-bit followed by the necessary number of 0-bits, so that its length in bits is an odd multiple of 256. Padding is always added, even if the message is already of the desired length. The length in bits of the original message (before padding), with a length of 256 bits is appended to the message. The produced message is a multiple 512 bits in length message that is represented by a sequence of 512-bit blocks b1, b2, . . . , bm, Each of these blocks is viewed as an 8X8 matrix of bytes. Similarly, the 512-bit key is depicted as an 8X8 matrix of byte. The diagram of the Whirlpool technique is given in figure (5), where its main layers are

- Add Key (AK)
- Substitute Bytes (SB)
- Shift Columns (SC)
- Mix Rows (MR),

The technique consists of a single application of AK followed by 10 rounds that involve all four functions.

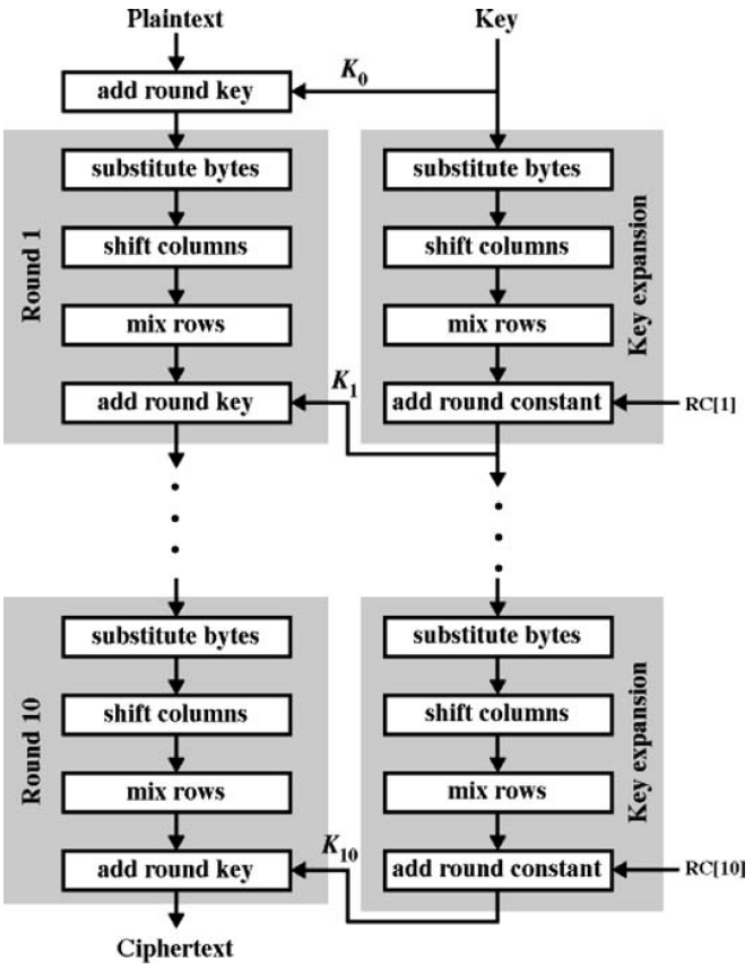


Figure 5. The Whirlpool Hash Function Diagram

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	18	23	C6	E8	87	B8	01	4F	36	A6	D2	F5	79	6F	91	52
1	60	BC	B	8E	A3	0C	7B	35	1D	E0	D7	C2	2E	4B	FE	57
2	15	77	37	E5	9F	F0	4A	CA	58	C9	29	0A	B1	A0	6B	85
3	BD	5D	10	F4	CB	3E	05	67	E4	27	41	8B	A7	7D	95	C8
4	FB	EE	7C	66	DD	17	47	9E	CA	2D	BF	07	AD	5A	83	33
5	63	02	AA	71	C8	19	49	C9	F2	E3	5B	88	9A	26	32	B0
6	E9	0F	D5	80	BE	CD	34	48	FF	7A	90	5F	20	68	1A	AE
7	B4	54	93	22	64	F1	73	12	40	08	C3	EC	DB	A1	8D	3D
8	97	00	CF	2B	76	82	D6	1B	B5	AF	6A	50	45	F3	30	EF
9	3F	55	A2	EA	65	BA	2F	C0	DE	1C	FD	4D	92	75	06	8A
A	B2	E6	0E	1F	62	D4	A8	96	F9	C5	25	59	84	72	39	4C
B	5E	78	38	8C	C1	A5	E2	61	B3	21	9C	1E	43	C7	FC	04
C	51	99	6D	0D	FA	DF	7E	24	3B	AB	CE	11	8F	4E	B7	EB
D	3C	81	94	F7	B9	13	2C	D3	E7	6E	C4	03	56	44	7F	A9
E	2A	BB	C1	53	DC	0B	9D	6C	31	74	F6	46	AC	89	14	E1
F	16	3A	69	09	70	B6	C0	ED	CC	42	98	A4	28	5C	F8	86

Table 1. Whirlpool S-box

The whirlpool algorithm Steps:

Step 1:
The Add Key layer (AK), where the key is used as input to the initial AK function. The key is expanded into a set of 11round keys. The Add Key Layer AK is used to perform the bitwise XOR operation on the 512 bits of round message block with the 512 bits of the round key.

Step2:
The substitute byte layer (SB) is used to provide a nonlinear mapping of the 8X8 byte matrix of the input text produced from the Add Key layer. The mapping uses a 16X16 matrix of byte values, called an S-box (Table 1) that contains a permutation of all possible 256 8-bit values. Each individual byte of input text is mapped into a new byte in based on the values of the leftmost 4 bits of the byte; as a row value, and the rightmost 4 bits; as a column value. The row and column values are used as indexes into the S-box to select a unique 8-bit output value.

Step 3:
The Permutation layer or the Shift Columns layer (SC) performs a circular downward shift of each column (except the first column) in the 8X8 matrix of bytes produced from step 2. For the second column, a 1-byte circular downward shift is performed, for the third column, a 2-byte circular downward shift is performed; and so on.

Step4:
The Diffusion layer or the Mix Rows layer (MR) performs diffusion within each row, by mapping each byte of a row into a new value that is a function of all eight bytes in that row. The transformation can be defined by the matrix multiplication: B = A. C, where A is the input matrix, B is the output matrix, and C is the transformation matrix:

01	01	04	01	08	05	02	09
09	01	01	04	01	08	05	02
02	09	01	01	04	01	08	05
05	02	09	01	01	04	01	08
08	05	02	09	01	01	04	01
01	08	05	02	09	01	01	04
04	01	08	05	02	09	01	01
01	04	01	08	05	02	09	01

The FH algorithm

INPUT: bit string X of arbitrary bit length b ≥ 0 and 512- bit key K.
OUTPUT: 512-bit hash-code of X.

Use secret key:

1) Define order for accessing words in the input 8X8 bytes matrix A, by using the secret key K (64 bits of K for each row i; 0 ≤ i ≤ 7, in A)
z[i*64::i*64+63] = Permutation P using the ith 64 bits of K,
P : {0, 1, …, 63} → {Oi | 0 ≤ Oi ≤ 63}],

2) Define the number of positions downward shifts (rotates) by using the secret key K (8 bits for each column j; 0 ≤ j ≤ 7, in A)
s[i*8::i*8+7] = Permutation P using the ith 8 bits of K,
Ps : {0, 1, …, 7} → {Oi | 0 ≤ Oi ≤ 7}],

3) Define the matrix C in the Mix Rows step using the secret key matrix.

The diagram of the Fusion Hashing algorithm is given in figure (6), where it has the same layers of the Whirlpool technique with the following modifications in steps 1-4.

The FH algorithm Steps:

Step 1:
The Add Key layer (AK), where the key is the input secret key K that is used as input to the initial AK function. The key is expanded into a set of 11round keys. The Add Key Layer AK is used to perform the bitwise XOR operation on the 512 bits of round message block with the 512 bits of the round key.

Step2:
As an addition to the substitute byte layer (SB) that is used in the original Whirlpool, a subsequent substitute layer that based on the P permutations using the secret key K.

Step 3:
The Permutation layer or the Shift Columns layer (SC) performs a circular downward shift of each column (except the first column) in the 8X8 matrix of bytes produced from step 2. The permutation will be based on the permutation mapping Ps.

Step4:
The Diffusion layer or the Mix Rows layer (MR) performs diffusion within each row, by mapping each byte of a row into a new value that is a function of all eight bytes in that row. The transformation can be defined by the matrix multiplication: B = A. C, where A is the input matrix, B is the output matrix, and C is the 8X8 bytes secret key matrix:

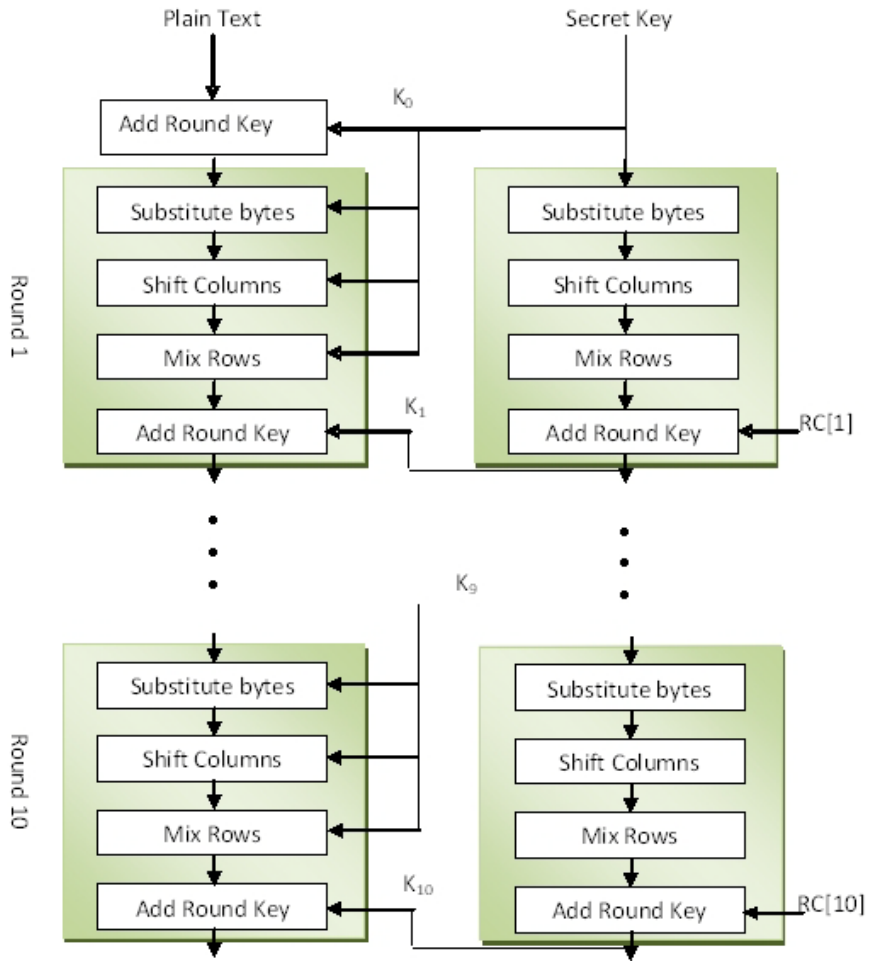


Figure 6. The FH algorithm Hash Function Diagram

3 Case Study

We've implemented the framework in a banking environment that has its own automated business processes in BPEL format generated by oracle BPA Suite; its network has infrastructure security components which include a SRX 3600 Juniper firewall, a Wireless security component for Wireless access points Cisco1520, and a juniper IDP250 Intrusion Detection and Prevention Appliance, the basic configuration of the server that was used is 2xQuad Core 3.2 GHZ with 64 GB RAM running windows 2008 R2 with SP2. The implementation has been performed as follow:

At Pre-production Phase:

1. Obtaining the bank security ontology criteria, Policy Script.
2. Obtaining the bank BPEL file that is needed to be validated.
3. Obtaining the values for the (QoS) throughput coefficients.
4. Apply correct WSS security policy for the enterprise to this file, adding different correct token values for BPEL and WSDL.
5. The above inputs are then fed into layer 1 of the framework to be processed and obtain a validated BPEL files each with validated tokens and its corresponding validated WSDL files' sizes.
6. The validated BPEL and WSDL were fed to the GM at layer 2 to produce securely enhanced BPEL with digestive hash value for each token and file contents using proposed FH algorithm.
7. We have measured the time consumed by whirlpool and FH algorithm to generate the digestive hash. . The time complexity of the FH algorithm at this phase is O (n) where n is the file size.
8. In the preparation step of the FH algorithm the input is padded to have blocks of 512bits and since the maximum expected token length is within the range of 0-256 bits, then the padded input will always have one block of 512 bits. As a result of that the time consumed by the FH algorithm to generate the digestive hash for tokens is fixed and was measured as ~0.002 microsecond.
9. For BPEL and its corresponding WSDL files, table (2) and Figure (7) illustrates the time in T*10000 unit (T=10 microsecond); consumed by whirlpool & FH; and different validated file sizes in KB {200-10000}. Figure (7) shows the comparison in time between applying whirlpool to create the files' contents digestive values and applying FH algorithm to create the same values. The result shows there is a tiny difference in time, and suggests that using our developed FH algorithm provides enhanced security without a noticeable degrading in performance.

File Size(KB)	Whirlpool algorithm	Fusion algorithm
200	6.467	22.4501
400	7.769	32.0727
600	10.565	42.7509
800	15.025	51.99
1000	18.531	63.2324

Table 2. Time values for applying whirlpool and FH to an experimental files' sizes (Pre-Production Phase)

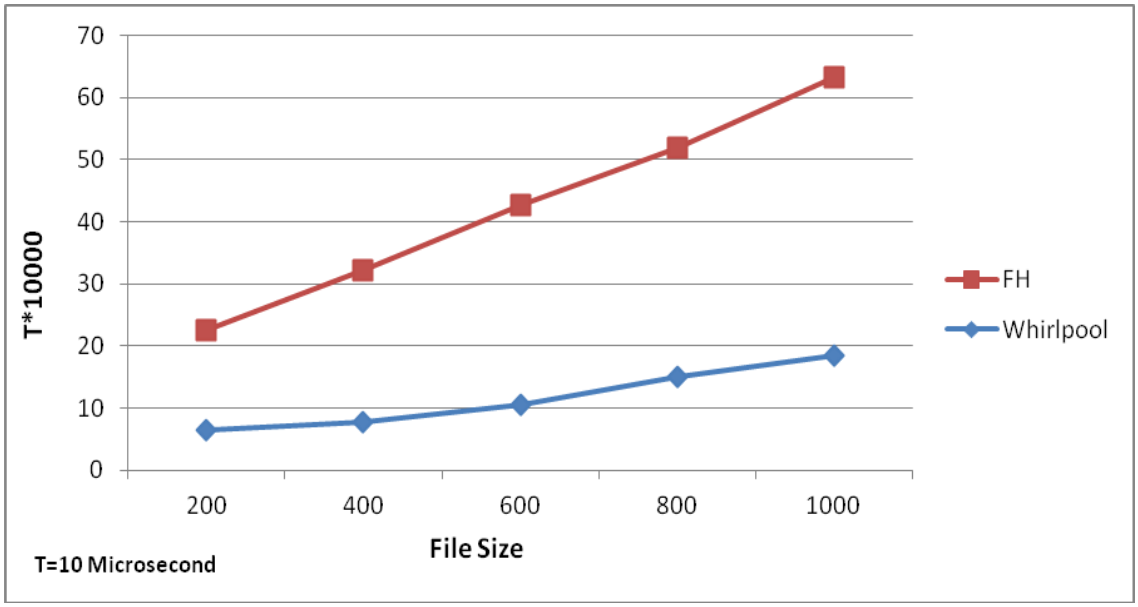


Figure 7. Comparing time values for applying Whirlpool and FH using files' sizes

At Production Phase:

1. A periodic security check and/or on-demand security check is called on the securely enhanced BPEL
2. The securely enhanced BPEL and its corresponding WSDL files are fed into the MM at layer 2.
3. The Token values, their digestive hash values, and files' contents digest values are extracted from the WSDL files by the MM.
4. The Token values and their digestive hash values are extracted from the WSDL files by the MM.
5. The MM calls the HM.
6. The FH algorithm is used by the HM to generate a corresponding digestive hash values for alteration checking of passed files and tokens. The time complexity of the FH algorithm at this phase is O (n) where n is the file size.
7. In the preparation step of the FH algorithm the input is padded to have blocks of 512bits and since the maximum expected token length is within the range of 0-256 bits, then the padded input will always have one block of 512 bits. As a result of that the time consumed by the FH algorithm to generate the digestive hash for tokens is fixed and was measured as ~0.002 microsecond.
8. For BPEL and its corresponding WSDL files, table (3) and Figure (8) illustrates the time in T*10000 unit (T=10 microsecond); consumed by GM and MM; and different file sizes in KB {200-1000}. Figure (8) shows the comparison in time between the FH used separately by the GM at pre-production phase and using FH in conjunction with a digestive hash comparison steps (CT) at production phase. The result shows there is a tiny difference in time, and suggests that using our developed FH algorithm provides enhanced security without a noticeable degrading in performance.

File Size(KB)	Fusion algorithm	Fusion algorithm + CT
200	22.4501	22.4551
400	32.0727	32.0777
600	42.7509	42.7559
800	51.99	51.995
1000	63.2324	63.2374

Table 3. Time values for using FH at pre-production and production phases respectively

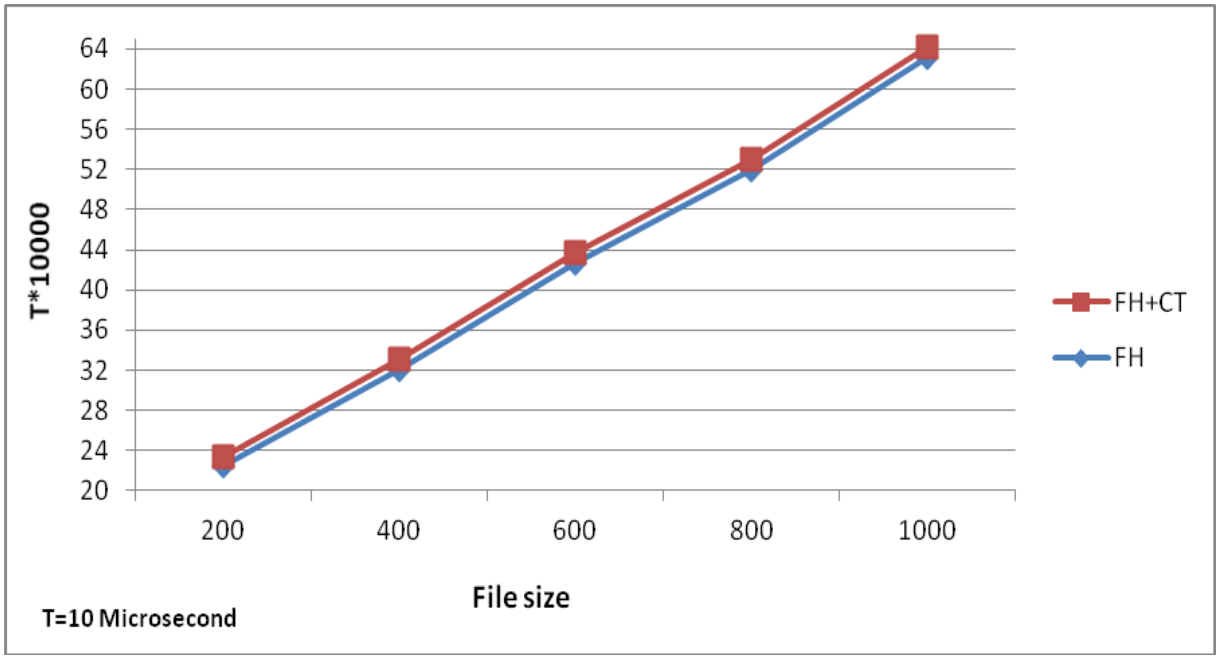


Figure 8. Performance of FH at pre-production and production phases.

4 Related Work

We have gone through reviewing two types of related work, one is concerned with security ontology frameworks and the other is concerned with message authentication codes and hashing algorithms.

A connectional security model presented in [7], which classified the security policies to the integration of dynamic selection of services provided by foreign organizations, and then the proposed model, determines an aggregation of security requirements in cross organizational service compositions. Huang [6] proposed a new ontology that represents the security constraints as a policy and then built the framework that uses the proposed ontology. The framework illustrates the integration workflow between business rules and non-functional description into policy specification. In [12] an approach is presented to improve end-to-end security by annotating service descriptions with security objectives between partners. A process model-driven transformation approach to concrete security implementations such as XACML or AXIS2 security configurations was introduced in [11].

Information integrity and authentication is considered vital for modern enterprises internal and external transactions [15]. Integrity and authentication are usually combined together by various algorithms, where integrity assures the transaction is delivered without alteration while authentication assures the identity of the sender [14]. Message Authentication Code algorithms can be constructed using Message Detection Codes in which raise a concern with MDCs properties [16].NMACA technique can work with wide range of message authentication codes, NMACA-MD5 was presented as a conjunction with MD5 hashing algorithm[2].

The previous related work did not address correlating the infrastructure security components performance with the BPEL through the new security Token tag. Moreover most of the authentication and integrated techniques didn't address the enterprise workflow security represented in business processes.

5 Conclusion and Future Work

In this paper, a framework with a security enhancement engine was introduced, it is composed of two layers, the first layer is to validate a BPEL and its corresponding WSDL files against the enterprise

security ontology criteria, and produce a validated BPEL. The second layer at the pre-production phase enhances the securely adapted BPEL by adding a digestive hash using the Generator Module. The digestive hash was generated by a Hashing Module that uses a developed Fusion Hashing algorithm. While at the production phase the second layer through the Monitoring module detects any alteration; if occurred; either in the token value, digestive hash value, file contents and/or all. We have implemented the framework in a real banking environment measuring the performance of the developed Fusion hashing algorithm at both the pre-production and production phases. In future work we are planning to model the internal behavior of the framework using PetriNet.

References

[1] Ahmed A. Hussein, Ahmed Ghoneim, and Reiner R. Dumke : An Approach for Securing and Validating Business Processes Based on a Defined Enterprise Security Ontology Criteria. The International Conference on Digital Information Processing and Communications. The ICDIPC2011 proceedings will be published in the “Communications in Computer and Information Science” (CCIS) Series of Springer LNCS).2011

[2] Khaled S. Alghathbar and Alaaeldin M. Hafez, “The Use of NMACA Approach in Building a Secure Message Authentication Code”, , International Journal of Education And Information Technologies Issue 3, Volume 3, 2009

[3] Erl, Thomass: Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice Hall, 2005

[4] Papazoglou, Mike P.: Service-Oriented Computing: Concepts, Characteristics and Directions, Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE’03), pp. 3, 2003

[5] Fragoso-Diaz, Olivia Graciela, -Salgado, Santaolaya René and Gyves-Avila, Silvana: Web Services for Software Development: the Case of a Web Service that Composes Web Services, The Third International Conference on Software Engineering Advances, pp. 31-36, OCT. 2008

[6] Dong Huang: Semantic Descriptions of Web Services Security Constraints, In SOSE '06: Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering. 2006. Page(s):81 - 84

[7] Michael Menzel; Christian Wolter; Christoph Meinel: Towards the Aggregation of Security Requirements in Cross-Organisational Service Compositions, 11th International Conference, BIS 2008, Innsbruck, Austria, May 5-7, LNCS 2008. Page(s): 297 - 308

[8] Dario Bruneo, Salvatore Distefano, Francesco Longo, Marco Scarpa: QoS Assesment of WS-BPEL Processes through non-Makrovian Stochastic Petri Nets, Proceeding of 2010 IEEE international symposium on Parallel & Distributed Processing (IPDPS), Atlanta, USA, April 19-23. ISBN: 978-1-4244-6442-5

[9] OASIS Web Services Business Process Execution Language (WSBPEL)TC. :Web Services Business Process Execution Language Version 2.0, OASIS, April 2007, **http://docs.oasis-open.org/wsbpel/2.0/wsbpelv2.0.html**

[10] Zayati, Lilia Sidhom, Youakim Badr, Frédérique Biennier, Mohamed Moalla: Towards Business Ontologies Matching for Inter-Enterprise Collaboration Platform in a Lean Manufacturing Strategy. PRO-VE 2010: 746-754

[11] Christian Wolter, Michael Menzel, AndreasSchaad, Philip Miseldine, Christoph Meinel: Model-driven Business Process Security Requirement Specification, Journal of Systems Architecture, Volume 55, Issue 4,Secure Service-Oriented Architectures (Special Issue on Secure SOA), April 2009, Pages 211-223, ISSN 1383-7621

[12] Youakim Badr, Frédérique Biennier, Samir Tata: The Integration of Corporate Security Strategies in Collaborative Business Processes, In IEEE Transactions on Services Computing, Issue 99, ISSN: 1939-1374, 2010

[13] Frédérique Biennier, Régis Aubry, Mathieu Maranzana: Integration of Business and Industrial Knowledge on Services to Set Trusted Business Communities of Organizations. PRO-VE 2010: 420-426

[14] Marc Stevens, Arjen Lenstra, and Benne de Weger,Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities,EUROCRYPT 2007 (Moni Naor, ed.), LNCS, vol.4515, Springer, 2007, pp. 1–22

[15] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu."Finding collisions in the full SHA-1." In VictorShoup, editor, Advances in Cryptology — CRYPTO 2005, volume 3621 of Lecture Notes in Computer Science, 2005

[16] "The Keyed-Hash Message Authentication Code (HMAC)", Federal Information Processing Standards Publication 198, March 2002

Der 3C-Ansatz für Agile Qualitätssicherung

André Janus

André Janus – IT Consulting / Universität Magdeburg

mail@andre-janus.de

Abstrakt: *In der Agilen Software-Entwicklung finden Methoden der klassischen Qualitätssicherung auf den ersten Blick keinen Platz. Dennoch helfen die Erkenntnisse der klassischen Qualitätsicherung auch in einem Agilen[1] Umfeld, wenn sie geschickt mit den verwendeten Agilen Praktiken kombiniert werden. Dieser Beitrag stellt den 3C[2]-Ansatz vor, der zeigt wie auf der Basis der Agilen Praktik Continuous Integration eine Agile Qualitätssicherung entwickelt werden kann. Die Erfahrungen stammen aus einem Agil durchgeführten Projekt der Firma T-Systems und beziehen sich auf erfolgreich durchgeführte Maßnahmen zur Sicherstellung und Verbesserung der Code-Qualität in einem Wartungs- und Weiterentwicklungsprojekt.*

Schlüsselbegriffe; *Agilität, Qualitätssicherung, Continuous Integration, Software-Wartung, Empirische Fallstudien*

Einleitung

Eine der bekanntesten Agilen Praktiken ist Continuous Integration (CI). CI hat einen großen Anteil daran, dass Agile Software-Entwicklungsprojekte es oft schaffen hochqualitative Software zu liefern [1]. Meist wird aber von einem "sauberen" Projektstart ausgegangen und der Projekt-Kontext, in der Praxis sehr oft Wartung und Weiterentwicklung, nicht ausreichend berücksichtigt. Zudem werden Metriken in der Agilen Community oft sehr skeptisch begegnet [2], obwohl der Nutzen von Software-Messungen prinzipiell unabhängig vom Vorgehensmodell ist.

Software-Wartung und -Weiterentwicklung

In der Praxis zeigt sich, dass die Code Qualität, vor allem bei sehr altem Source Code, oft zu wünschen übrig läßt. Gerade wenn sich zu sogenannten Brownfield-Projekten [3] noch Zeit- und Kostendruck gesellen, sind Qualitätsprobleme schwer in den Griff zu bekommen. Die Agile Software-Entwicklung bietet zwar viele Praktiken an, um das Entstehen solcher Qualitätsprobleme zu verhindern, aber wenn das Kind erst einmal in den Brunnen gefallen ist, greifen auch die Praktiken der Agilen Software-Entwicklung nicht mehr ausreichend, gerade wenn ein Projekt in die Wartungphase geht und erst hier die Agilen Praktiken zur Anwendung kommen. Auf der anderen Seite sind aus der klassischen Qualitätssicherung viele Metriken und Methoden bekannt um die Softwarequalität zu messen und Verbesserungspotential aufzuzeigen. Wie diese Probleme während einer laufenden Wartung und Weiterentwicklung zu beheben sind, erklären aber auch diese Verfahren nicht ausreichend.

Agilität, Messverfahren und Metriken

Hier kommt eine Kombination aus der Agilen Praktik Continous Integration und klassischer Qualitätssicherung ins Spiel. Wenn Metriken und Messverfahren in den Agilen Entwickungsprozess integriert werden, können aus den Ergebnissen kontinuierlich und ohne die normale Entwicklung zu stören Verbesserungs-maßnahmen abgeleitet werden.

[1] **a**gil: wortwörtl. "flink", "beweglich"; **A**gile: "dem Agilen Manifest folgend/entsprechend"
[2] **C**ontinuous Integration, **C**ontinuous Measurement, **C**ontinuous Improvement

erhält man einen Verlauf der Änderungen über die Zeit – das Controlling der Software-Qualität bekommt man damit quasi geschenkt. Neben der Verwendung von Code-Metriken, die sich auf die interne Qualität der Software beziehen, werden auch Metriken mit Bezug auf die externe Qualität erhoben. Die Code-Metriken reichen von den simplen Lines of Code (LOC) [4] über Kommentardichte bis zu den komplexeren CK-Metriken [5] für Objektorientierte Software. Aber auch aus dem Agilen Bereich stammende Metriken wie Testabdeckung [6] finden Verwendung. Welche Metriken genauer betrachtet werden, kann – ganz im Sinne des Agilen Manifests [7] – das Team entscheiden.

Continuous Integration

Die Agile Praktik Continuous Integration (CI) [Abb. 1] [8] wird heute im Allgemeinen mit einer automatisierten CI-Engine im Zusammenspiel mit einem Versions Control System (VCS) gleichgesetzt: Der Entwickler implementiert neue Funktionalitäten in seiner IDE und merged sie in die gemeinsame Code-Basis, die in einem VCS wie z. B. in diesem Projekt Subversion (SVN) [9] vorgehalten wird. Die CI-Engine – in diesem Projekt CruiseControl [10] im Zusammenspiel mit dem Build-Tool Ant [11] – prüft das VCS in regelmäßigen Abständen auf Änderungen. Falls Änderungen erfolgt sind, kompiliert und baut (build) die CI-Engine die Anwendung und führt (Unit-)Tests aus. Da es sich in diesem Projekt um eine auf Java-Technologie basierende Anwendung handelt, kommt JUnit [12] zum Einsatz. Wenn diese Schritte erfolgreich waren, gelten die Änderungen als voll integriert – falls ein Fehler auftritt, werden der entsprechende Entwickler und ggf. weitere Personen benachrichtigt. Die höchste Priorität hat nun das Finden und Beheben der Fehlerursache, um wieder ein voll integriertes System herzustellen. Die Continuous Integration bildet damit ein Quality-Gate (Q-Gate), das die externe Qualität durch die Tests (dynamische Code- Checks) absichert. Wie in der Agilen Software-Entwicklung üblich, gibt es weitere "manuelle" QGates in Form von Reviews und Pair Programming, bei denen ein weiterer Entwickler die Arbeit des implementierenden Entwicklers prüft. Zusätzlich gibt es Coding Standards, die als (Formatierungs-)Regeln in der IDE hinterlegt ein weiteres "automatisiertes" Q-Gate bilden. Auch die Agile Praktik des Refactoring findet an dieser Stelle Eingang.

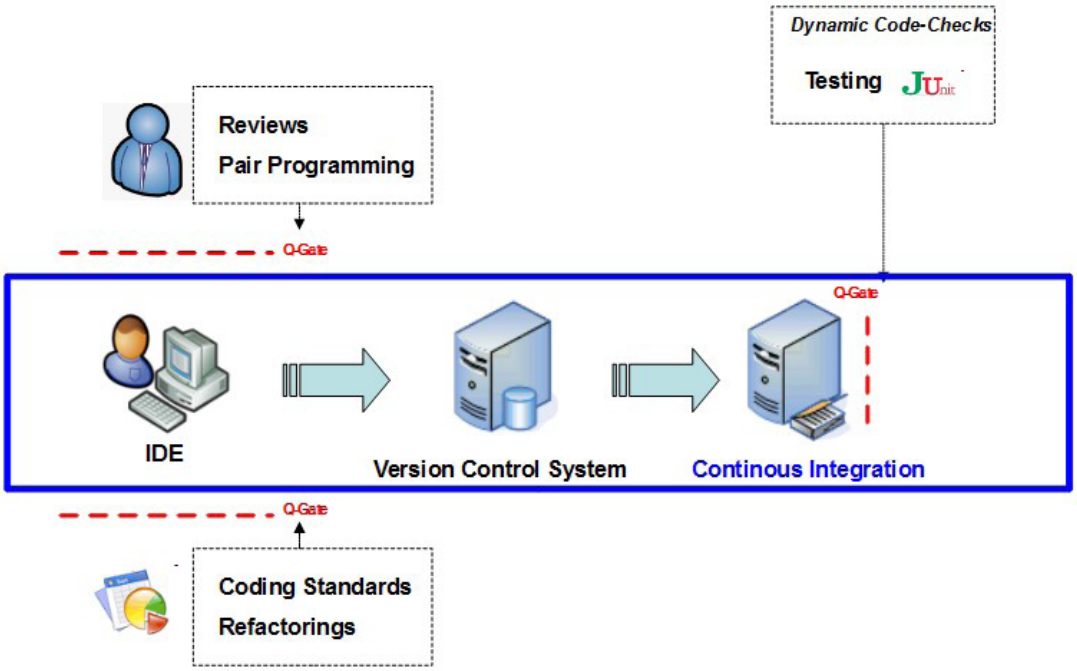


Abbildung 1: Continuous Integration

Continuous Measurement

CI-Engines und Build-Tools bieten vielfältige Möglichkeiten weitere Tools zur Überprüfung des Source Codes anzubinden, damit diese bei jedem Lauf der CI den Zustand des Source Codes messen und in entsprechenden Reports darstellen [Abb. 2]. Für diese statischen Code-Checks kommen im vorliegenden Projekt die Tools Findbugs [13], Checkstyle [14] und PMD [15] zum Einsatz. Da diese Tools erst im Nachhinein eingeführt wurden, enthalten die Reports viele Regel-Verstöße, die allerdings erst bewertet und vor dem Hintergrund einer schon seit Jahren weiterentwickelten und gewarteten Software gesehen werden müssen. Viele Verstöße stammen aus einer Zeit, in der mit der damals vorliegenden Java-Version noch keine andere Lösung möglich war. Ein Beispiel hierfür ist die Verwendung von Generics, bspw. eine Liste mit Strings: `List<String>`. Die Ergebnisse dieser Tools geben einen Blick auf die interne Qualität der Software. Auch die CIEngine selbst liefert viele Werte aus dem Compile- und Build-Prozess wie z. B. Informationen über
die erfolgten Änderungen oder verwendete Artefakte. Die Metrik LOC sowie Regel- erstöße der Tools werden in diesem Projekt mittels eines "Cockpit" genannten Diagramms übersichtlich dem Zeitverlauf nach dargestellt, so dass auf einen Blick eine Tendenz erkennbar wird. Ein weiteres hilfreiches Tool ist Cobertura [16], das im Java-Bereich die Testabdeckung mißt. Damit ist erkennbar, welche Bereiche des Source Codes während der Testausführung verwendet, d. h. getestet werden. Die genannten Tools stammen aus dem Java-Bereich. Für viele gibt es aber auch Varianten z. B. für das ebenfalls populäre .NET.

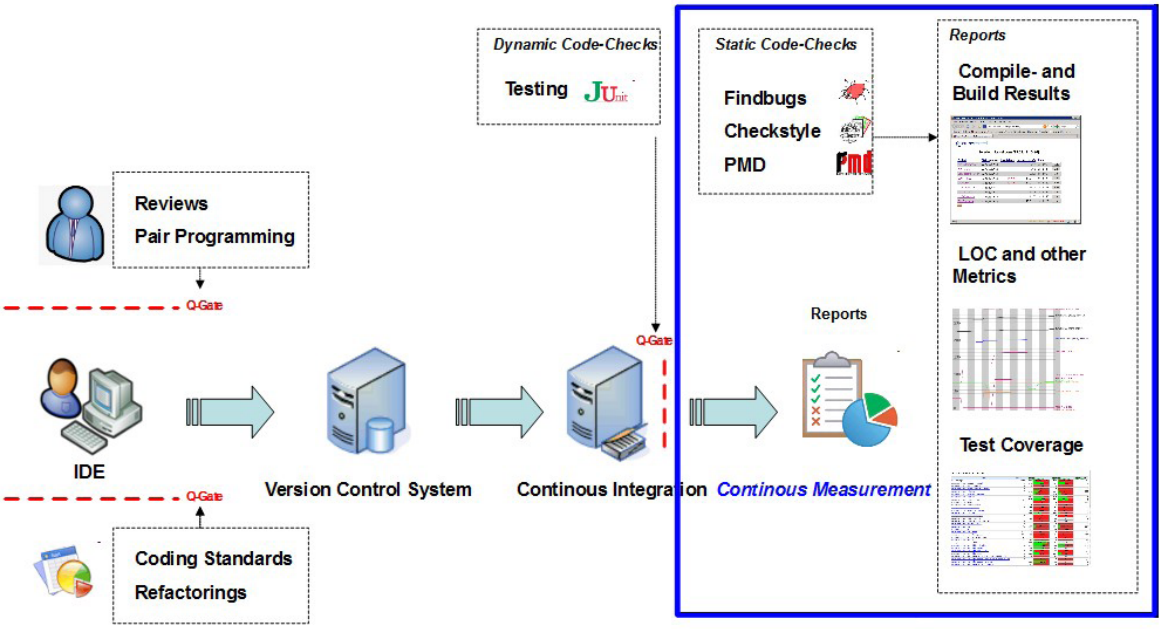


Abbildung 2: Continuous Measurement

Continuous Improvement

Der Measurement bzw. Messvorgang kann in die automatisierte CI integriert werden. Das Interpretieren der Messergebnisse und das Ableiten von geeigneten Massnahmen ist aber ein manueller Vorgang [Abb. 3] – der nichtsdestotrotz regelmäßig durchgeführt werden sollte! Diese Maßnahmen können Änderungen und Anpassungen an Coding Standards oder Coding Style Guide sein. Es kann sich auch um das explizite Einplanen von Refactorings in die normale Weiterentwicklung handeln, um die Struktur des Source Codes zu verbessern. Eine Möglichkeit ist die im Projekt durchgeführte Typisierung von untypisierten Generics, deren große Zahl durch die Reports der Tools entdeckt wurde. Die Interpretation und das Ableiten von Maßnahmen fällt der Rolle des Quality Managers zu, der aber im Sinne der im Agilen geforderten cross-functional teams aus dem Entwickler-Team stammen sollte. Der Charme dieser Lösung besteht darin, dass Vorgaben die aus dem Team selbst kommen, eher akzeptiert werden, zudem kennen sich die Entwickler am besten im Source Code

aus und können daher auch am ehesten die Ergebnisse in den Reports interpretieren. Eine weitere Möglichkeit zur Qulitätssicherung, mit Betonung auf "Sicherung", bietet sich dadurch, dass Regelverstöße, die durch eine Refactoring behoben wurden, als neues Abbruchkriterium für die CI-Engine definiert werden können. Somit wird sichergestellt, das einmal behobene Schwächen des Source Codes nicht über die Weiterentwickung wieder Einzug halten. Dafür sorgt die Vorgabe der Pratik CI, dass beim Fehlschlagen der Integration mit höchster Priorität an der Fehlerbehebung gearbeitet werden muss. Somit ist dieser Teil der Qualitätssicherung automatisiert; der weit größere Teil bleibt aber weiterhin manuell, da zu viele Abbruchbedingungen bzw. tatsächliche Abbrüche der CI die Akzeptanz im Team gefährden.Um die Akzeptanz im Team für die Qualitätssicherung weiter zu erhöhen, sowie den Spaß bei der Arbeit, was übrigens ein nicht zu unterschätzender qualitätsfördernder Aspekt ist, zu fördern, ist die Verwendung eines sogenannten "Extreme Feedback Device" ein sehr gute Möglichkeit. Dieses Device zeigt auf möglichst "spektakuläre" Weise den Status der CI(-Engine). Zwei Projektmitarbeiter waren hier kreativ und haben mittels einer IP-fähigen Steckerleiste eine Lampe an die CI-Engine angeschlossen. Schlägt eine Integration fehl, leuchtet die Lampe auf und projiziert ein eindeutiges Symbol an die Decke des Büros, das sich auch auf dem Symbol des Quality Managers in [Abb. 3] findet [17].

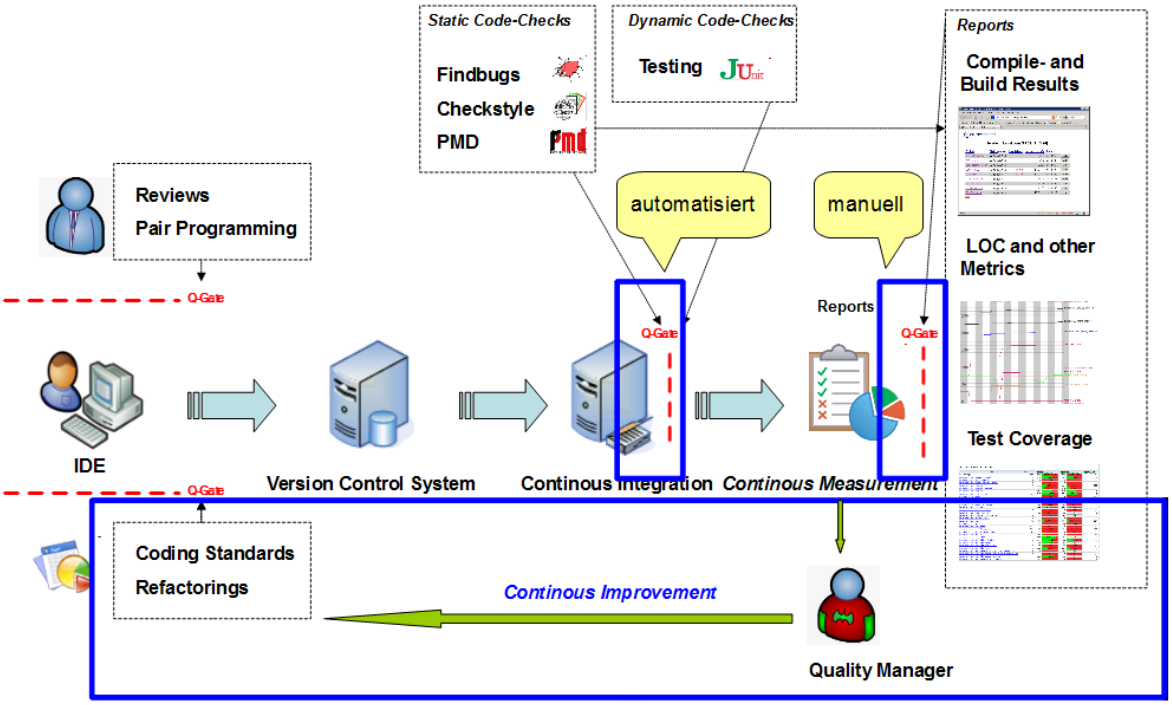


Abbildung 3: Continuous Improvement

Fazit

Die Erfahrungen aus dem Projekt zeigen, dass die Agile Praktik Continuous Integration eine nützliche Basis und die entsprechenden Tools eine gute Infrastruktur für einen kontinuierlichen Messprozess und somit ein Agiles Qualitätsmanagement bilden. Die Kombination von Agiler Software-Entwicklung und klassischen Qualitätsicherungsverfahren bieten einen guten Ausgangpunkt für ein "Continuous Improvement". Ein wichtiger Vorteil dieses Vorgehens ist, dass nicht nur punktuelle Messungen und Verbesserungen möglich sind, sondern dass sich die Verbesserungen auch über die Zeit durch automatisierte Q-Gates erhalten lassen. Die Herausforderung bei diesem Vorgehen bleibt die Interpretation und Analyse durch den Quality Manager. Dieser muss die Maßnahmen und Verbesserungen zielführend auswählen und außerdem darauf bedacht sein, dass die Maßnahmen vom Team akzeptiert und umgesetzt werden. Zusätzlich zeigt sich hier, dass die Agile Software-Entwicklung – obwohl auf das Software-Entwicklungsteam fokussiert – einen weiteren Schritt in Richtung Industrialisierung ist, indem Automatisierung (CIEngine, Messprozess) und Standardisierung (Coding standards, Coding Style Guide) umgesetzt werden.

Referenzen

[1] Paul M. Duvall, Steve Matyas, Andrew Glover. 2007.Continuous Integration: Improving Software Quality and Reducing Risk (Martin Fowler Signature Books). Addison-Wesley

[2] Jens Coldewey. OBJKEKTSpektrum 03/2010. Schattenboxen: Warum Metriken nicht funktionieren können – und es trotzdem tun

[3] Stefan Lieser, Ralf Westphal. 2009-2010. Herausforderung Brownfield, Heise Developer: **http://www.heise.de/developer/artikel/Clean-Code-Developer-in-Brownfield-Projekten-855114.html**

[4] S. H. Kan. 2002. Metrics and Models in Software Quality Engineering. 2nd Edition. Addison-Wesley

[5] Chidkamber & Kemere. 1993. A Metric Suite for Object-Oriented Programming. MIT Sloan School of Management

[6] Glenford J. Myers. 2004. The Art of Software Testing, 2nd edition. Wiley

[7] Manifesto for Agile Software Development, **http://www.agilemanifesto.org/**

[8] Simon Wiest. 2010. Continuous Integration mit Hudson/Jenkins: Grundlagen und Praxiswissen für Einsteiger und Umsteiger. Dpunkt-Verlag

[9] Subversion (SVN), **http://subversion.apache.org/**

[10] CruiseControl, **http://cruisecontrol.sourceforge.net/**

[11] Ant, **http://ant.apache.org/**

[12] JUnit, **http://www.junit.org/**

[13] Findbugs – Find Bugs in Java Programs, **http://findbugs.sourceforge.net/**

[14] Checkstyle, **http://checkstyle.sourceforge.net/**

[15] PMD, **http://pmd.sourceforge.net/**

[16] Cobertura, **http://cobertura.sourceforge.net/**

[17] eXtreme Feedback Device: The Batman Lamp: **http://www.jensjaeger.com/2010/04/extreme-feedback-device-the-batman-lamp/**

Quantitative Approach of IT Security Management Processes

Detlef Günther, Volkswagen AG, Wolfsburg

1 Introduction

In the Business world of today information have the same priority as a productivity factor as human resources, budget, time and money. The availability of up-to-date and consistent information is a strategic success factor for the majority of large enterprises worldwide, not only for their competitive ability but also for securing their further existence. More and more business processes e.g. logistic processes, production control, customer care and just in time delivery services, are supported by information systems based on information, manual inputted, scanned, measured, computed, transferred and stored.

So business processes are direct or indirect depending on Information Technology (IT) and outcome of this are higher requirements for protection of information and information systems. In case of a dysfunction of information systems, business processes could be interrupted caused by a manipulation or loss of information.

Information is not only the basis for the development, production, marketing and sales of new innovative products but also for a successful management of an enterprise. For this reason business requirements of large enterprises are forcing new challenges upon the IT organization.

To respond to these challenges, the IT functions needs to strange standardization in

- Requirement engineering
- Modeling
- Processes

The following figure shows the integration of IT, from the technology level up to the requirements and model into the business strategy and business requirements of enterprise and the interconnection of applicable standards and best practices, which are covering different divisions of an IT organization.

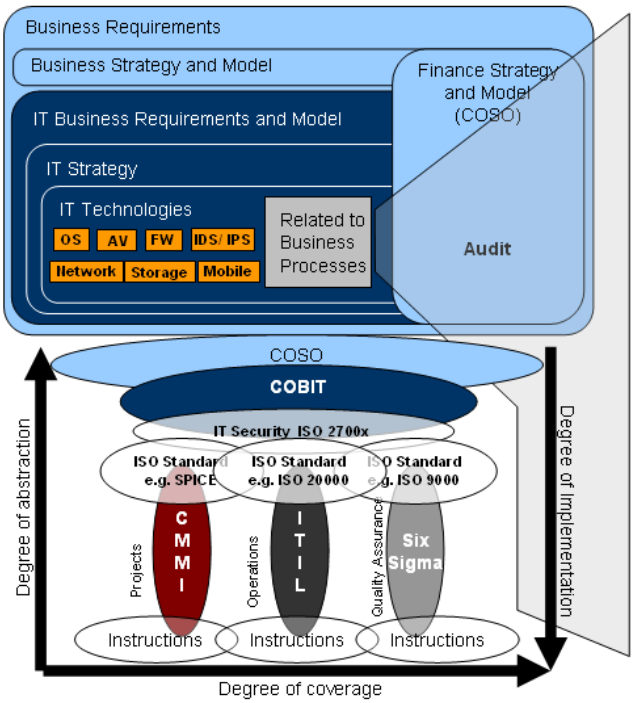


Figure 1. Business Requirement Engineering

From this perspective result the following needs, to provide a synergetic, complementary and aligned support for the business processes:

- Forceful application of standards in processes
- Measuring of quality/ maturity
- Audits to evaluate degree

2 Basics of Security in IT Infrastructures

An essential aspect of problems in new/modern software systems is the ***security.*** Methodologies and strategies in order to keep this aspect are summarized in ***security management*** and ***security engineering***. Typical areas in security engineering are described in following ([Anderson 2001], [Katzenbeisser 2000], [Oppliger 2000]) characterized as following [Günther 2002]:

Security policies: is "an expression of intent of a system's owner or operator within which the system should operate" [Phoha 2002]. Kinds of intentions could be any rules about authentication, access control, authorization, confidentially etc.

Security principles: Using modern We-based infrastructures, many people are scam victims. Stajano and Wilson identify the following seven principles against these threats [Stajano 2011]: *Distraction principle:* "While we are distracted by what grabs our interest, hustlers can do anything to us and we won't notice.", *social compliance principle:* "Society trains people to not question authority. Hustlers exploit this 'suspension of suspiciousness' to make us do what they want.", *herd principle:* "Even suspicious marks let their guard down when everyone around them appears to share the same risks. Safety in numbers? Not if they're all conspiring against us.", *dishonesty principle:* "Our own inner larceny is what hooks us initially. Thereafter, anything illegal we do will be used against us by fraudsters.", *kindness principle:* "People are fundamentally nice and willing to help. Hustlers shamelessly take advantage of it.", *need and greed principle:* "Our needs and desires make us vulnerable. Once hustlers know what we want, they can easily manipulate us.", *time principle:* "When under time pressure to make an important choice, we use a different decision strategy, and hustlers steer us toward one involving less reasoning.".

Security management: The general processes of security identifying, analysis and controlling are the essential aspects in security management [Rubin 1998]. Management approaches could be consists in the supporting the ISO/IEC 27001 based IT security self assessment [Khurshid 2009] or the support of metrics visualization in the automotive industrial environment [Zeiler 2010].

Security policy specification: "Internet and e-commerce applications typically involve many different organizations – manufacturers, merchants, network service providers, banks, and customers. Specifying and analyzing the security for such environments to determine who can access what resources or information can be extremely difficult" [Sloma 2005]. Security policy specification involves both: security management and policy-driven network management.

Security technologies: Basic security technologies are cryptographic techniques (like RSA, AES etc.), authentication techniques (digital signatures, watermarking, fingerprints etc.), authorization approaches (passwords, DSA, biometrics etc.), security protocols (SSH, SSL TLS etc.), and security network techniques (VPN, ISP, firewalls etc.) [Dumke 2003], [Oppliger 2000], [Piva 2002], [Tilborg 2001], [Vitek 1999].

Multilevel security: "Multilevel concepts were originally developed to support confidentiality in military systems, there are now commercial systems that use multilevel integrity policies" [Anderson 2001]. Model requirements depends on the security policy model, security target and protection profile over the different levels of responsibilities, organizational entities and/or technological infrastructures.

Organizational security: This kind of security is a special approach covering the organizational structure and could be addressed to the different parts of the organization reasoning in distributed development, outsourcing and offshoring [Singh 2005].

Distributed trust management: "The trust-management approach to distributed-system security was developed as an answer to the inadequacy of traditional authorization mechanism. Trust-management engines avoid the need to resolve 'identities' in an authorization decision. Instead, they express privileges and restrictions in a programming language. This allows for increased flexibility and expressibility, as well as standardization of modern, scalable security mechanisms" [Ioannidis 2005].

Digital rights management: "Digital Rights Management more commonly known as DRM, aims at making possible the distribution of digital content while protecting the rights of the parties involved (mainly the content owners and the consumers)" [Atallah 2005]. Open standards are used to support the DRM as XrML (from XEROX PARC) and ODRL (as Open Digital Rights Language). Typical protection mechanism are been used as watermarking, fingerprinting, code obfuscation, software aging, dongles and hardware-based integrity.

Transaction security: Transactions are one of the kernel operations in e-business, e-commerce and digital marketplaces. Essential principles on this area are the *ACID principles* (as atomicity, consistency, isolation and durability), the *Kerberos principle* of transactions, the *secure electronic transaction* (SET) approach, the *secure transaction technology* (STT), and the *management of certificates* [Dumke 2003], [Oppliger 2000], [Singh 2005]. Note that in Web services the 'A' and the 'I' of the ACID principle are not fully implemented [Papazoglou 2008].

Electronic payment systems: This techniques or technologies play an important role in the acceptance of any e-commerce systems. Typical solutions on this area are *electronic cash systems* (like eCash, CAFÉ, NetCash etc.), *electronic checks* (as NetBill, PayNow, NetCheque etc.), *electronic credit card payments* (like CyberCash, STT and SET), and *micropayment systems* (like Millicent, PayWord and MiniPay) [Anderson 2001], [Dumke 2003], [Oppliger 2000].

Business privacy protection and anonymity services: Typical technologies in order to implement privacy protection are *content hiding, content deniability, association hiding,* and *association deniability.* Considering the service anonymity, special techniques are helpful like anonymous browsing, onion routing, crowds, anonymous publishing and personalized web assistants [Anderson 2001], [Hassler 2001], [Oppliger 2000].

Emission and intrusion detection: The emission detection try to avoid any attacks in e-business and e-commerce systems. Typical examples of these security problems are *spoofing, phishing* and *spamming*. The intrusion detection helps to identify any trials of attacks as soon as possible [Anderson 2001], [Dumke 2003], [Brukczynski 2003].

Information security: "A complete solution to the information security problem must thus meet the following requirements: *secrecy of confidentiality,* which refers to the protection of data against unauthorized disclosure, *integrity,* which means the prevention of unauthorized or improper data modification, and *availability,* which refers to the prevention of and recover from software errors and from malicious denials making data not available to legitimate users" [Bertino 2005]. A helpful technology in the field of information hiding is well-known as *steganography* [Katzenbeisser 2000].

E-Commerce system protection: The protection of e-commerce try to avoid failures such as misconfigured access control, implementation blunders, theft of network services and inapproriate use of cryptology [Anderson 2001].

The following figure characterizes any kinds of securities considering a business-oriented software system.

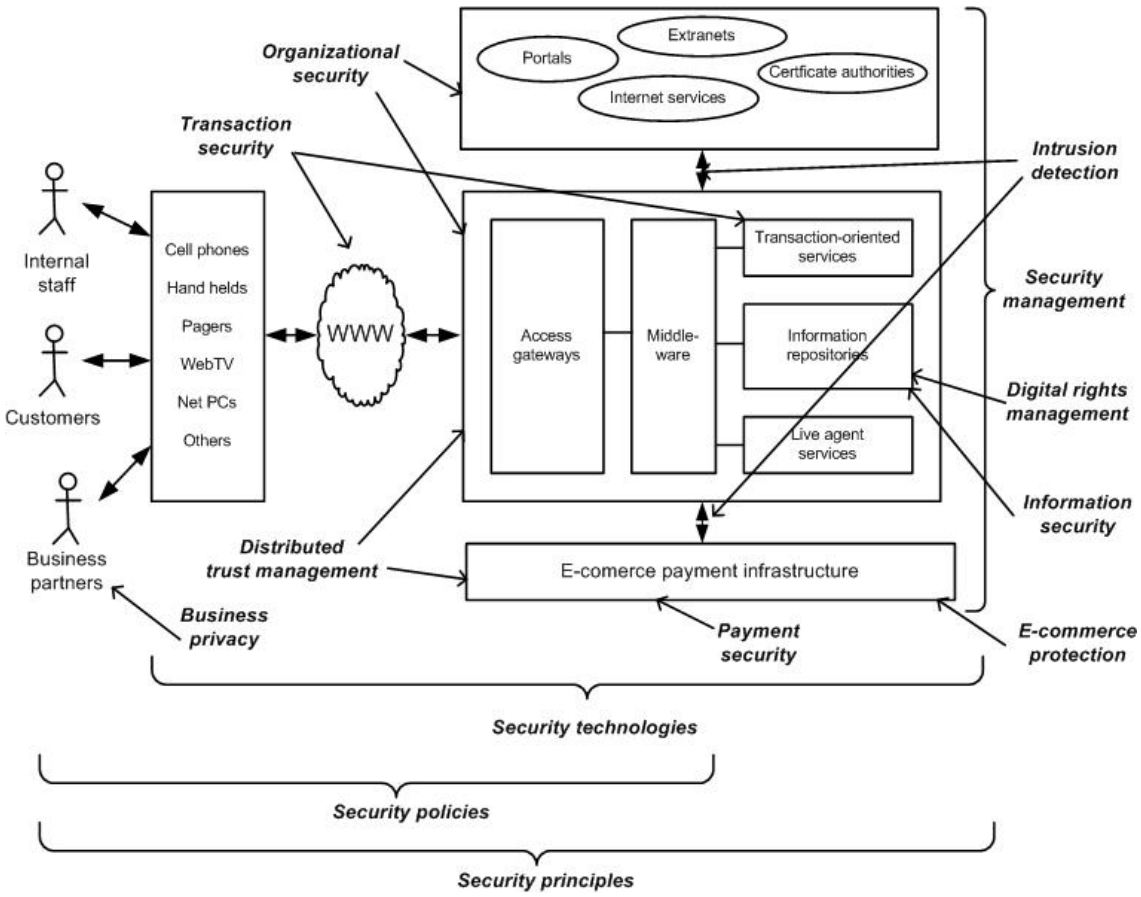


Figure 2. Securities in business-oriented software systems

Further considerations are addressed to the *internet-based lowing prices* [Lesk 2011], to the problems of *nudging privacy* in personal information [Acquisti 2009], to the *trust and privacy in cloud computing* [Ghosh 2010], [Takabi 2010], to the *formal methods in security and privacy* [Datta 2010], and to the *privacy in internet* in general [Kaufman 2011].

3 Challenges for IT Security Management Processes

Enterprise Information Security Management is not trivial. Heterogeneous infrastructures make a management of security technologies difficult. Decentralized distributed responsibilities for planning and operations, basic services and application management are not easy to control. Hype technologies that are not “Enterprise ready” like iPhones and iPads create not only needs but greediness especially in the top management. These technologies let security architects no choice for alternatives and give very often not enough time for a safe implementation in the enterprise architecture. Upcoming challenges not only for Information Security Managers but also for Auditors and Forensic Investigators are in addition to new technologies also threats.

Threats of the real world like misuse, theft, extortion, espionage and fraud are mostly the same in the digital world but using modern information systems. Product piracy increased and supported by information leakage is one of the most important threats for automotive enterprises these days. Implementation of security technologies helps to reduce the risk related to these threats, but presenting not a complete picture of Information Security Management.

Efficient, effective and secure IT-Processes are the basis for reliable Business Processes. The following figure shows the IT integration in the business processes of an production enterprise including the interaction of the business processes with the supporting IT systems or applications down

to the interconnection network level, the process and application strategy with the need of standardization in an IT - Master Construction Plan and last but not least the IT Security processes.

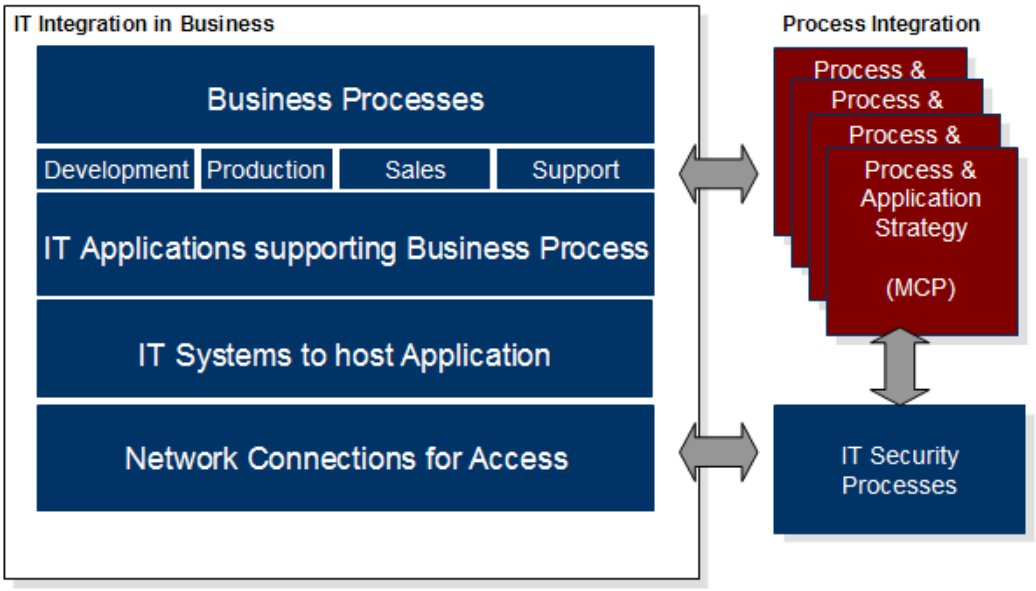


Figure 3. Business Processes dependent on IT Security

4 The New Approach: Business Process Integration

The new approach of security-based business process improvement considers the different IT security standards (ISO 2700x, CobIT (new Version 5.0), ITIL and BSI) Baseline Protection Manual) as a first step shown in the following figure.

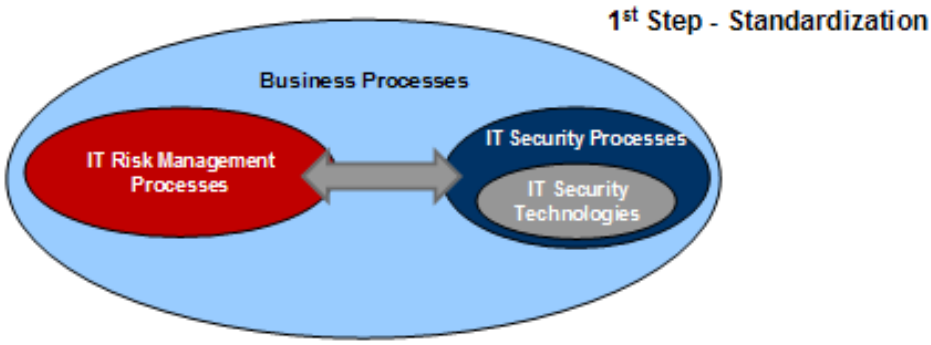


Figure 4. Risk Management and IT Security Processes

Furthermore, essential problems must be solved as different approaches, difficult for IT security audits, translation matrices needed, decision for one applicable standard, and the minimum requirements for further and secure existence of processes. These are the next steps in order to keep process efficiency and consistency for improving the world-wide business process security requirements and solutions.

References

[Acquisti 2009] Acquisti, A.: *Nudging Privacy – The Behavioral Economics in Personal Information.* IEEE Security and Privacy, Nov./Dec. 2009, pp. 82-85

[Anderson 2001] Anderson, R.: *Security Engineering – A Guide to Building Dependable Distributed Systems.* John Wiley & Sons Publ., 2001

[Atallah 2005] Atallah, M. et al.: *Digital Rights Managemant.* In: Singh, M. P.: *The Practical Handbook of Internet Computing.* Chapman & Hall CRC Publ., 2005, pp. 21-1 – 21-16

[Bertoni 2005] Bertoni, E.; Ferrari, E.: *Information Security.* In: Singh, M. P.: *The Practical Handbook of Internet Computing.* Chapman & Hall CRC Publ., 2005, pp.26-1 – 26-18

[Brukczynski 2003] Brukczynski, B.; Small, R. A.: *Reducing Internet-Based Intrusion: Effective Security Patch Management*. IEEE Software, February 2003, S. 50-57

[Datta 2010] Datta A.: *Formal Methods in Security and Privacy.* IEEE Security and Privacy, Nov./Dec. 2010, pp. 86-89

[Dumke 2003] Dumke, R.; Lother, M.; Wille, Z.; Zbrog, F.: *Web Engineering.* Pearson Education Publ., Munich 2003

[Ghosh 2010] Ghosh, A.; Arce, I.: *In Cloud Computing We Trust - But Should We?* IEEE Security and Privacy, Nov./Dec. 2010, pp. 14-16

[Günther 2002] Günther, D. Mews, G.: *Security Engineering for E-Commerce Applications(German).* Diploma Thesis, University of Magdeburg, Dept of Computer Science, 2002

[Hassler 2001] Hassler, V.: *Security Fundamentals for E-Commerce*. Artech House Publ., 2001

[Ioannidis 2005] Ioannidis, J.; Keromytis, A. D.: *Distributed Trust.* In: Singh, M. P.: *The Practical Handbook of Internet Computing.* Chapman & Hall CRC Publ., 2005, pp. 47-1 – 47-16

[Katzenbeisser 2000] Katzenbeisser, S.; Petitcolas, F. A. P.: *Information Hiding - techniques for steganography and digital watermaking*. Artech House Publ., 2000

[Kaufman 2011] Kaufman, G.: *How Private is the Internet?* IEEE Security and Privacy, Jan./Febr. 2011, pp. 73-75

[Khurshid 2009] Khurshid, S.: *Tool Support for ISO/IEC 27001 based IT Security Self Assessment*. Master Thesis, University of Magdeburg, Dept. of Computer Science, 2009

[Lesk 2011] Lesk, M.: *What is Internet Worth?* IEEE Security and Privacy, Jan./Febr. 2011, pp. 88-90

[Oppliger 2000] Oppliger, R.: *Security Technologies for the World Wide Web.* Artech House Publ., 2000

[Papazoglou 2008] Papazoglou, M. P.: *Web Services: Principles and Technology*. Pearson Education Publ., Harlow, 2008

[Phoha 2002] Phoha, V. V.: *Internet Security Dictionary.* Springer Publ., 2002

[Piva 2002] Piva, A.; Bartolini, F.; Barni, M.: *Managing Copyright in Open Network*. Internet Computing, June 2002, S. 18-26

[Rubin 1998] Rubin, A. D.; Geer, D. E.: *A Survey of Web Security*. IEEE Computer, September 1998, S. 34-41

[Singh 2005] Singh, M. P.: *The Practical Handbook of Internet Computing.* Chapman & Hall CRC Publ., 2005

[Sloma 2005] Sloma,M.; Lupu, E.: *System Management and Security Policy Specification.* In: Singh, M. P.: *The Practical Handbook of Internet Computing.* Capman & Hall CRC Publ., 2005, pp. 46-1 - 46-20

[Stajano 2011] Stajano, F.; Wilson, P.: *Understanding Scam Victims: Seven Principles for System Security.* Comm. of the ACM, 54(2011)3, pp. 70-75

[Takabi 2010] Takabi, H.; Joshi, J. B. D.; Ahn, G.: *Security and Privacy Challenges in Cloud Computing Environments.* IEEE Security and Privacy, Nov./Dec. 2010, pp. 24-31

[Tilborg 2001] Tilborg, H. C. A. van: *Fundamentals of Cryptology – A Professional Reference and Interactive Tutorial*. Kluwer Academic Publ., 2001

[Vitek 1999] Vitek, J.; Jensen, C. D.: *Secure Internet Programming*. LNCS 1603, Springer Publ., 1999

[Zeiler 2010] Zeiler, F.: *Supporting Metrics-based Analysis of IT Security Processes in the Automotive Industrial Environment.* (German) Bachelor Thesis, University of Magdeburg, Dept. of Computer Science, 2010

Efficiency in Integrated Legacy Systems based-SOA

Ayman Massoud

Otto-von-Guericke University, Magdeburg Germany

Abstract. *Many of the quality properties have considered during the systems integration process. Verification and validation (V&V) is one of these properties that shape the complexity and the efficiency of the integration framework. The architecture in SOA Migration has to implement some sort of mechanisms and web services to be able to check the V&V property and to be sure that the integrated legacy systems have got the expected goals and benefits, and also to confirm that the integration process has completed successfully from the providers and the consumers' services point of view.*
The following sections illustrate how to deploy the mentioned property in the integration architecture by providing a proposal framework for integrating legacy systems based SOA. The function and the behavior of the proposal framework and its web service(s) is simulated using an example of selected integrated business processes between ERP and CMMS legacy systems as a case study.

Introduction

Bulk of the business processing today is still carried out by legacy systems and large packaged applications especially in the real and critical business, like Enterprise Resource Planning ERP, Computerized Maintenance Management Systems CMMS, and so on. These kinds of systems are working in siloed nature environments which faced several challenges that making them lost to security, automation, data integrity and cooperation. The cost-effective approach to overcome this island behavior is to keep these systems running and to base new solutions on the existing applications portfolio and leverage integration as a mechanism for accessing the existing capabilities. Migrate these systems to work together under Service-Oriented Architecture SOA can move the enterprise forward toward its business goals, benefits from the loose–coupling and flexible architecture, and to adapt easily with the new business strategies of the organization to provide a competitive products and/or services to its customers.

The efficiency of the ***migration process*** depends on the quality properties considered by the migration architecture including the maintainability, governance, validation and verification, security, and so on.

Formally, we will define the software product as a (software) system as [Skyttner 2005]

$$SP = (M_{SP}, R_{SP}) = (\{programs, documentations, data\}, R_{SP})$$

In order to develop a software product we need resources such as developer, CASE tools and variants of hardware. Therefore, we define the software development resources *SR* as following

$$SR = (M_{SR}, R_{SR}) = (\{personnelResources, softwareResources, platformResources\}, R_{SR})$$

So, we can define the software process *SD* as following (including the essential details of every development component)

$$SD = (M_{SD}, R_{SD}) = (\{developmentMethods, lifecycle, softwareManagement\} \cup M_{SR}, R_{SD})$$

After the software development, the software product goes in two directions: at first (the original sense of a software product) to the software application *SA***,** second in the software maintenance *SM.* We define the different aspects in following

$$SA = (M_{SA}, R_{SA}) = (\{applicationTasks, applicationResources, applicationDomain\} \cup M_{SP}, R_{SA})$$

The different aspects and characteristics of the software maintenance are summarized by the following formula

$$SM = (M_{SM}, R_{SM}) = (\{maintenanceTasks, maintenanceResources\} \cup SP)$$

Considering ***legacy software systems*** *SS* we can describe the following characteristics

$$\begin{aligned} SS_{legacySystems} = (M_{SS}, R_{SS}) &= (\{ components, userDocs, dataBases\} \\ &\cup \{ businessFunctions, businessRules, governance\}, R_{SS}) \\ &= (\{LS_i\} \cup \{BS_j\}, R_{SS}) \end{aligned}$$

where *LS* stands for legacy system and *BS* for business logical service. Furthermore, we describe with *DB* as data basis

$$\begin{aligned} r^{(CMMS)}_{legacySystem} \in R_{SS}: \; & maintenanceModule \times procurementModule \\ & \times invoiceModule \times DB \rightarrow workflowEngine \end{aligned}$$

and

$$\begin{aligned} r^{(ERP)}_{legacySystem} \in R_{SS}: \; & DB \times accountPayableModule \times generalLedgerModule \\ & \rightarrow inboundOutboundInterfaceService \end{aligned}$$

Considering ***service orientation***, we can establish different *service views* as [Papazoglou 2008]: *Customer view, technological view, provider view, business view.* The general areas in service development can be characterized as [Alonso 2004]:

- General approach: *service oriented system engineering (SOSE) based on service oriented software (SOS) and service oriented computing (SOC)*
- General aspects: *evolution, complexity, self regulation, organizational stability, compatibility, continuous extension, decreasing quality security*
- Service modelling: *service composition, service interoperability,* π *calculus, ontologies (OWL-S, WSMO)*
- Basic services: *payment, contracts, negotiation, information, receipt, trust, reputation, discovery, safety transactions, security, large transactions*

Especially, the SOA notion could be explained as following by [Marks 2006]: “SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable service on an IT network. ‘Services’ in an SOA are modules of business or application functionality with exposed interface, and are invoked by messages.”

Usually, the SOA is divided in different layers such as *user interface, presentation layer, process layer, application layer* and *data base layer.*

From a business point of view, we differ between *enterprise/application layer, business layer, utility layer* and *basic layer.*

The system view of service concepts could be defined as observed behaviour of a system (as provider) as a set of interactions and their relationships between the system and the environment [Masak 2009, p. 15 ff.]. From a system theoretical point of view, the service concepts involves

- Basic service characteristics: *autonomy, interfaces, composability, behaviour* and *non functional aspects*
- Non functional service characteristics: *name of provider, service time/duration, language(s), quality of service* (QoS)*, security, locality, availability, prices/payment, penalties*
- SOE evolutions: *creativity, hierarchy, delegation, coordination, collaboration, enterprise network*
- SOE dimensions: *competency, efficiency, flexibility, motivation, coordination*

- SOE definition: as *temporary and dynamic collaboration between autonomous services for consumers in a changed environment as tuple*

  SOE = (*goals, entities, environment, lifecycle*)

- SOE agilities: *as customer agility, partner agility* and *organizational agility*

The SOA center approach motivates for centralization of (Web) services of any large company in order discover the service composition and management in an user-friendly manner [Schmietendorf 2007]. Based on a SOA strategy defined as

$$SOA_{strategy} = (SOA_{strategy}, R_{SOA}) = (\{SOA_{process}, SOA_{layer}, SOA_{standards}, SOA_{evaluation}, SOA_{information}, SOA_{organization}\}), R_{SOA})$$

we can characterize this center approach including the main SOA involvements as

$$r_{SOA}^{center} \in R_{SOA}): SOA_{information} \times SOA_{process}, \times SOA_{evaluation} \times SOA_{organization}$$

The SOA maturity model of Welke et al. [Welke 2011] are based on the consideration of SOA drivers as IT-related drivers such as infrastructure efficiency, reuse and application/data composition and integration, and enterprise-related drivers such as business analytics and processes, organizational flexibility and agility and enterprise transformation. The different maturity levels in this approach are

1. *Initial maturity:* “At the initial maturity level, most firms adopt SOA for IT-related reasons and seek to develop finer-grained software components, largely from existing APIs, to achieve greater interoperability and platform efficiency”.
2. *Managed maturity:* “As they defined and create more and finer-grained services, organizations need to employ common service definitions and some form of registry to manage service policies”.
3. *Defined maturity:* “A fundamental shift occurs as firms move simple beyond the simple, IT-focused management of services toward actual definitions of new services drive by business requirements and defined in business functionalities terms”.
4. *Quantitative managed maturity:* “The redesign of business process lets organizations be more agile and flexible but to quantitatively manage all of the services, they must create an ESB and define services in terms of business needs and in advance of their use in processes, mashups, and so on”.
5. *Optimized maturity:* “At the final level of optimized maturity, firms define, develop, and implement business processes that are themselves services”.

An efficient process effect should be given combining these levels with general process standard/approach levels like CMMI etc.

SOA quality of service modelling is a wide area and was considered later in this technical report. Now, we want to indicate shortly any typical approaches on this research field. The first consideration is a classification of the QoS areas including some technologies as [Hilari 2009] (see also [Johnson 2007] and [Papazoglou 2008])

- *QoS Approaches*: *Q-WSDL, WS-QDL, UDDI QoS, QoS Broker*
- *QoS in WS composition*: *WSFL, BPEL, Petri nets, QoS aggregation, optimization*
- *QoS monitoring*: *self healing (MAESoS), discovery (WSSQoS) etc.*

Web service reputation modelling by Kalepu et al. is based on the consideration of *compliance* and *verity* including the service attributes in order or estimate the service reputation [Kalepu 2004]. The QoS intention of [Gangadharan 2011] considers the *copyrights* and *moral rights* in service-based software. This investigation includes the analysis of language support like Web Service Level

Agreement (WSLA), Web Service Offering Languages (WSOL), SLA languages like SLAng, Web Service Policy Language (WSPL), Collaboration Protocol Profile and Agreement (CPP/CPA) and Open Digital Rights Language for Services (ODRL-S). The *service-based framework for flexible business processes* of Ardagna et al. uses *quality gates*, *quality annotations* and *QoS matchmarker* in order to keep SLA requirements and involvements [Ardagna 2011].

General QoS approaches and solutions: The service development must consider the existing SOA-infrastructure from the customer side. That means that the service should be usable within the established runtime-environment. The quality of service (QoS) was defined in the manner the "QoS refers to the ability of the Web service to respond to expected invocations and to perform them at the level commensurate with the mutual expectations of both its provider and its customer" [Papazoglou 2008].

A first approach for an evaluation model follows the well known GQM (goal question metric) paradigm and leads to the granularity identification by the use of metrics. Therefore an assessment for the granularity behavior of a service offering should provide answers to the following questions [Schmietendorf 2010]:

- *How much should be the size of the service interface?*

  (Number of provided operations, Number of contained data types for each operation, Kinds of used data types)

- *How much business functionality is considered by the service?*

  (Information about the successful application, Number of encapsulated legacy systems, Difference between basic and composite services)

- *Is a "white box" or/and "black box" view supported?*

  (Information about the used implementation technology, Structure metrics for the service implementation, Overview of the chosen implementation architecture)

The next approach by Papazoglou is based on the QoS characteristics *performance and capacity, availability* and *security/privacy* and suggests the SLA (service level agreement) contents structures as following: *purpose, parties, validity period, scope, restrictions, service-level objectives, penalties, optional services, exclusion terms* and *administration* [Papazoglou 2008]. Furthermore, it was differed between *static SLAs* and *dynamic SLAs* in the development and application context (see also [Toma 2006]).

The ITIL (IT Infrastructure Library) is a meta concept as a *summarizing of best practices* for software management in generally and for SOA characteristics especially [Johnson 2007]. The quality assurance was kept *implicitly* as a orientation of *improvements* by appropriate kinds of management. These areas of management are

- *Demand management:* define and publish services, self-service user request catalog, represent SLAs and costs
- *Enterprise change management:* software and operational change, standardize and automate ITprocess workflow
- *Service level management:* define and track negotiated business service levels, prioritize service activities based on SLA impact
- *Incident/problem management:* integrated service desk, support automation (optimization, deflection and prevention), decision support for service desk optimization, centralized knowledge base
- *Service costing:* usage-based charge back (invoicing) or cost allocation, credits for SLA violations, role-based decision support

The well-known ITIL cycle is shown in the following figure adapted from [Johnson 2007].

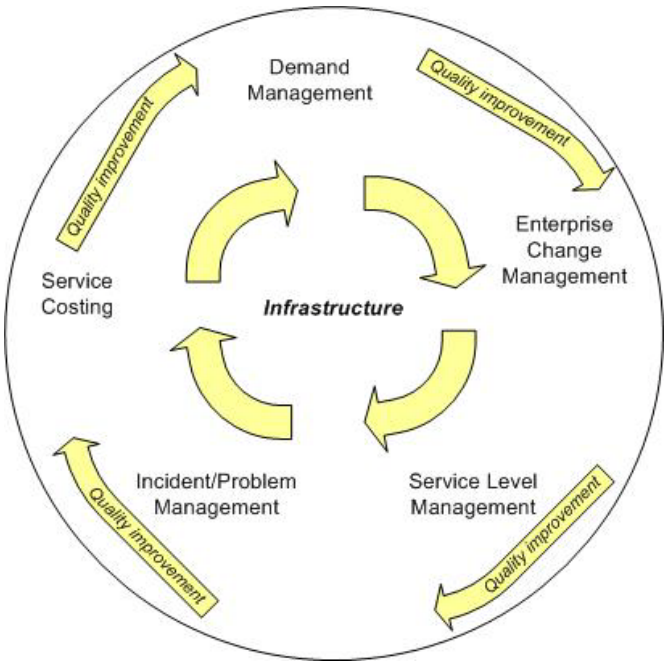


Figure 1. The ITIL management cycle

The general QoS approaches in the German BSOA (Bewertung serviceorientierter Architekturen) community can be characterized as ([Schmietendorf 2007], [Schmietendorf 2010]),

- Conception and defining of SOA quality reports,
- Benefits analysis of industrial service oriented architectures
- Technology-based analysis of SOA implementations (Web 2.0, Cloud computing etc.)
- Consideration of service wrapping for legacy systems
- Compliance and risk analysis in the IT service management
- Identity management improvements and measurements
- Performance analysis of SOA infrastructures and WS-BPEL processes
- Security protocols for service evaluation in SOA applications
- Availability analysis of long-time service uses and consumptions
- Dynamic quality models for Web service management
- Effort estimation for SOA solutions and industrial applications
- etc.

SOA processes quality measurements in the context of a SOA processes (as a scorecard) could refer to the following areas [Schmietendorf 2007]:

- *SOA business measurements*: Market penetration, time to market, customer satisfaction, turnover increase etc.
- *SOA process Measurements*: Process terms, process mistakes, number of process-referential events etc.
- *SOA financial measurements*: Return of Investment, cost savings, project costs etc.
- *SOA usage measurements*: Number of used service offerings, number of service customers etc.
- *SOA performance measurements*: Performance of basic and orchestrated Services and availability etc.
- *SOA IT efficiency measurements*: Productivity, development time, quality behavior etc.
- *SOA optimization measurements*: Number of services in conception, development and production etc.
- *SOA governance measurements*: Standard conformity, potential exceptions etc.

Background and Motivation

Enterprise legacy systems may individually run incomplete business process that missing some specific functions or data, synchronize real time events or consolidate its data, sending and exchange messages across the enterprise' systems, therefore and to draw a complete business picture, these systems need to work under an integrated enterprise architecture to cooperate internal with each other to overcome these limitations and to enhance the enterprise business performance.

Given the broad of integration access as a way to expose existing enterprise legacy functionality and data to business services, several integration access have introduced to provide the integration services. Specific proper approaches for integration access implementation are depending on some major factors and can be summarized as follows:

If the enterprise has an integration middleware including either a message broker or a widely used messaging infrastructure, existing middleware can be leveraged for implementing integration services.
If the enterprise using a packaged application to run its major business units and these applications expose its functions using a web services, the implementation of integration services should leverage these web services.

If the enterprise business services are implemented using J2EE application servers, which provide prepackaged adapters (based on JCA/J2C), these adapters can be used as a basis for implementing access with integration services.

If the integrated applications are component-based with well-documents functionality and interfaces, and it is easy to generate web services wrappers via the implementation platform to use it as a basis for implementing integration services.

If the majority of integration is in the rational database layer, database-specific middleware (JDBC for example) should be used for implementing integration services.

The legacy system integrates in order to provide or consume a business data, or to cooperate with another system to finalize and complete specific business logic. Some challenges and problems might be occurred in the production phase after integration implementation with the real data:

1- The ***provider system*** provides incomplete data that make inconsistent data problem causes technical and/or business impact in the consumer system.

2- ***Legacy system*** send a document to complete its business logic to another legacy system while no status feedback coming, the problem might be occurred because some technical or business errors faced the target legacy system make it not to be able to carry out its role and send a feedback to the source system as well.

3- ***Dependency process*** problem that should to be considered to organize the dialog interactions ordering between the integrated systems.

Missing a mechanism to monitor and control the validity and verification of the integration process and its data flow may cause a failed integration process, the integration framework architecture and the implementation selection approach is responsible to overcome this kind of problems, the following example as shown in figure 1 display this point in a real case of integrated legacy systems: The purpose of the integration is that the source legacy system sends an approved invoice message to the target legacy system to carry out the payment missing step and to post it in specific financial period to execute all the relative financial transactions and journals, by this step the source legacy system can close the procurement cycle with a complete process action and confirmation.

The given example of integrated framework consists of three layers; first layer for the source legacy system which called CMMS (Computerized Maintenance Management System) that manages plant maintenance and logistics activities through its modules as follows: Maintenance and warehouse modules that have its own business processes like planning and schedule work tasks, creating a work request to fix plant equipments that required to order a spare part, monitor and control the inventory

transactions of receiving and issuing materials, distributing maintenance cost, etc. Procurement and Invoice Modules, which manage the business processes of the purchasing functions from opening the purchase request up to delivering the ordered items to the requester, and finalize the process by issuing the invoice to the vendor which control the 3 way matches method that checks the value and the quantity mentioned between the purchase order, material receipt, and vendor invoice.

The second layer of target legacy system called ERP solution that supports the accounting and the financial functions through its modules like LG (General Ledgers) that create, update, and post the journal entries against any organization transactions. AP (Account payable) which manage the invoice payment process and close the procurement cycle opened in the source legacy system CMMS, and many other modules dealing with Account Receivable AR, Fixed Assets FA, Cost Management CM, etc.

Most of the maintenance business functions are addressed by the CMMS system while another type of functions are not covered such as accounting and financial functions that manage the communications between the enterprise money flow from one side and the customers and the suppliers from the other side, this limitations are founded because the CMMS system is not a financial solution, so, there is a need to integrate it with another system that support this kind of missing applications in order to provide a complete business process. And this done through sending the invoice document message via direct accessing the ERP interface table to insert the invoice data, and then the ERP system post the invoice and change its status from approved to paid status.

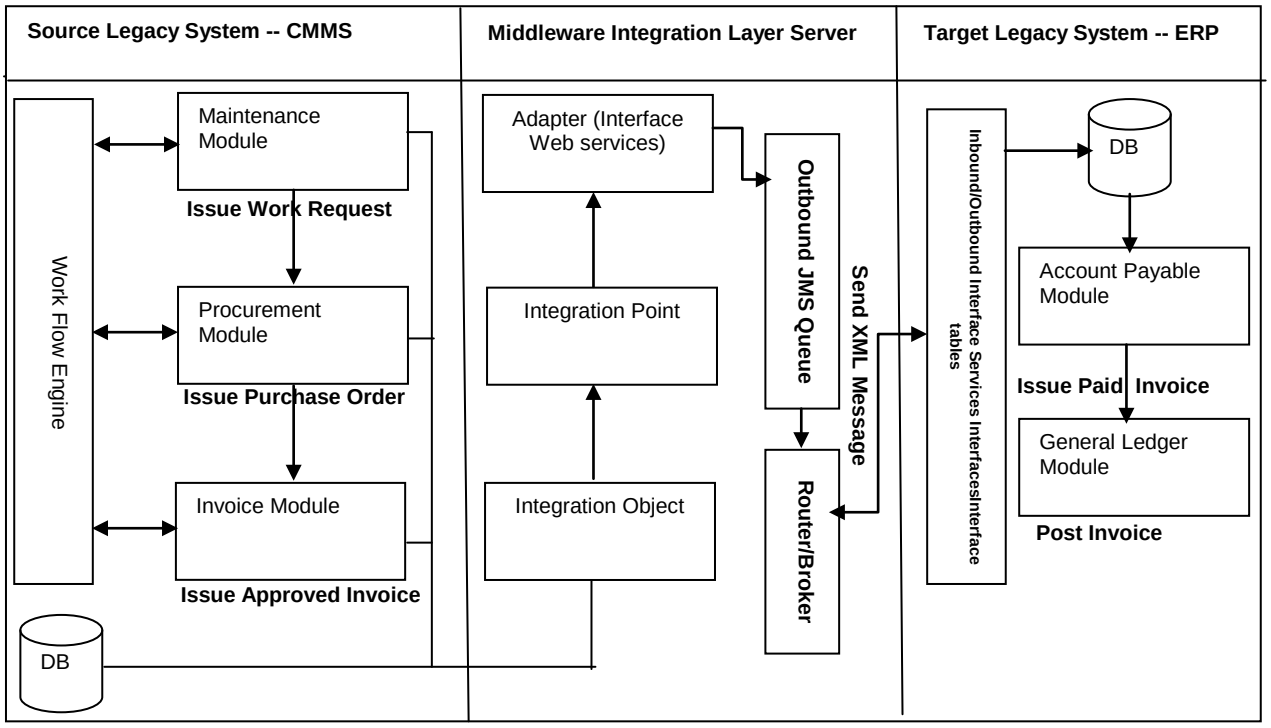


Figure 2. Integrated Legacy Systems framework using messaging infrastructure

The components of the middleware integration layer are working together to exchange messages between different systems, the given example involved several components functions as follows:

Integration Object is an integration service that mapping a functional unit of the business application server -in our example is a procurement business unit- and consists of a set of database fields and business rules used to update the database during the integration processing.

Integration point provides access to an integration object in a given direction (inbound or outbound) of the data transfer to and from the integrated application. The integration point direction properties used to perform a data synchronization, processes and responses queries.

Adapter list of components (interfaces, programs, mapping, and controls) which expose the integration object to the integrated systems, provide the ability to the external integrated system to access the integration point in the inbound direction, and send data to the external system in the outbound direction.

Interfaces used to transfer the data between the integrated systems format. Like the integration point the interface used the direction properties to perform a data synchronization, processes and responses queries.

Messages Queue inbound and out bound queue is a message service queue JMS used as a staging area during the exchange of message between the integrated systems like JMS BEA WebLogic and JMS IBM WebSphere application servers.

XML Message & Interface Tables integrated systems can exchange data transactions via XML messages, interface tables, and flat files. XML message and interface table (rational DB tables) can used interchangeably because both of them represents the same data files with the same format.

Router, responsible for routing outbound messages from outbound JMS queue to an end point associated with an external system. The router use one of the configured and implemented handler to convert the outbound data from the queue into one end point data location and form like EJB enterprise java bean. FLATFILE flat file, HTTP as an XML document to a URL using HTTP, IFACETABLE write data to an interface table to an interface tables in local or remote database, JMS handler is a Java component that delivers XML data into a messaging system that has been enabled through Java Messaging Service (JMS). Depending upon the messaging model had implemented, messages are placed in a virtual channel called a queue or topic: In the point-to-point messaging model, messages are generated by a sender and placed in a queue then only one receiver can obtain the message from the queue, in the publish-subscribe messaging model messages are generated by a publisher and placed in a topic then multiple subscribers can retrieve the message from the topic, WEBSERVICE a java component that invokes a specified web service with a SOAP request parameter, XMLFILE handler is a Java component that converts data in the outbound queue into an XML file format, then delivers it to the xmlfiles directory within the global directory.

So, we noticed from the previous framework there is no function component supported to make sure that the verification and validation properties has been applied. Therefore, a new framework provided to cover these properties should be modelled and implemented in an industrial anvironment.

References

[Alonso 2004] Alonso; Casati; Kuno; Machiraju: *Web Services – Concepts, Architectures and Applications.* Springer Publ. 2004

[Ardagna 2011] Ardagna D.; Baresi, L.; Comai, S.; Pernici, B.; Comuzzi, M.: *A Service-Based Framework for Flexible Business Processes.* IEEE Software, March/April 2011, pp. 61-67

[Gangadharan 2011] Gangadharan, G. R.; D'Andrea, V.: *Managing Copyrights and Moral Rights of Service-Based Software.* IEEE Software, March/April 2011, pp. 48-55

[Hilari 2009] Hilari, M. O.: *Quality of Service (QoS) in SOA Systems.* Master Thesis, Uni Catalonya, Spain, 2009

[Johnson 2007] Johnson, B.; Higgins, J.: *ITIL and the Software Lifecycle: Pactical Strategy and Design Principles.* Van Haren Publ., Amerfoort, Netherland, 2007

[Kalepu 2004] Kalepu, S.; Krishnaswamy, S.; Loke, S. W.: *Reputation = f(User Ranking, Compliance, Verity).* Proc. of the IEEE Int. Conf. on Web Services, July San Diego, 2004, pp. 200-207

[Marks 2006] Marks, E. A.; Bell, M.: *Service Oriented Architecture- A Planning and Implementation Guide for Business Technology. John Wiley & Sons, 2006*

[Masak 2009] Masak, D. *Digitale Ökosysteme*. Springer Publ. 2009

[Papazoglou 2008] Papazoglou, M. P.: *Web Services: Principles and Technology*. Pearson Education Publ., Harlow, 2008

[Schmietendorf 2007] Schmietendorf, A.: *Eine strategische Vorgehensweise zur erfolgreichen Implementierung serviceorientierter Architekturen in großen IT-Organisationen.* Shaker Publ., 2007

[Schmietendorf 2010] Schmietendorf, A. et al.: *BSOA 2010 – Bewertungsaspekte serviceorientierter Architekturen.* Shaker Publ., Aachen, Germany 2010

[Skyttner 2005] Skyttner, L.: *General Systems Theory – Problems, Perspectives, Practice*. World Scientific Publ., New Jersey, 2005

[Toma 2006] Toma, I.; Foxvog, D.; Jaeger, M. C.: *Modeling QoS characteristics in WSMO. MW4SOC* '06, November 27-December 1, 2006 Melbourne, Australia, pp. 102-107

[Welke 2011] Welke, R.; Hirschheim, R.; Schwar, A.: *Service-Oriented Architecture Maturity.* IEEE Computer, February 2011, pp. 61-67

SOA Measurement for IT Process Improvement

Anja Fiegler

T-Systems Magdeburg,Germany

anja.fiegler@t-systems.com

Abstract. *The IT's industrialization, standardization and commodity approach is based on Service-oriented architectures (SOA) and can be considered as an enabling methodology for Cloud Computing. The effort of effective SOA introduction requires new kinds of SOA measurement in order to keep the higher complexity and integration of new complex technologies.*
Our new approach is based on growth models related to SOA's reference architecture as well as diversity and entropy

1 Introduction

The consideration of Information Technology (IT) as an important enabling entity has changed critically to a cost factor, uncontrollable and in many cases an obstructive business instance. SOA was intended to solve some of the major struggles of IT.

The current IT driver "Cloud Computing" is the next approach for increasing standardization, commodity and industrialization in IT, where SOA can be considered as an enabling methodology for example; to distribute business processes and performances through the cloud and different service providers. Therefore SOA still has an important role in IT whilst diversity and complexity continues as a key driver on IT development projects.

In the context of SOA quality measurement, some hypotheses indicating how favorable characteristics could be measured are discussed.

2 Quality of Service-Oriented Architectures

SOA projects where challenged with the same classical problems like insufficient and/or inconsistent requirements as well as discrepancies between IT and specialist departments. Software quality measurement in a classical way is based on metrics such as Lines-of-Code (LOC), Functions-Points, McCabe, Healstead, classical Test-Bug statistics or later in operation with KPI's.

However, these approaches are not fully satisfactory and appropriate to give answers to the quality of SOA systems and processes over the complete lifecycle; beyond development and testing. As a baseline of new measurement approaches we have to organize the SOA reference architecture (Figure 1) as it is introduced from The Open Group [1] in the SOA Source Book [2].

Some other but similar references have been set and the alignment of interfaces, business process, service (components) and data or operational layers all share common abstracts. A further model is introduced in [3] were the business domains where deposited crosswise behind the layers.

Typical hypotheses about quality assurance in SOA could be identified in the following manner.

- Interface and business process layers may grow fast while lower layers like component or operational layers grow slower
- One possible quality aspect might be that the top layers growth does not adversely affect the growth of lower layers

- The system's complexity could be measurable and may reveal significant changes in each architectural layer due to clear business and / or technological reasons with each layer displaying different levels of dynamics
- The system's diversity overall is ever expanding but with flat rather than exponential gradients
- The system diversity could peak in one or different layers as a result of poor design; this would result in a negative impact in overall systems performance

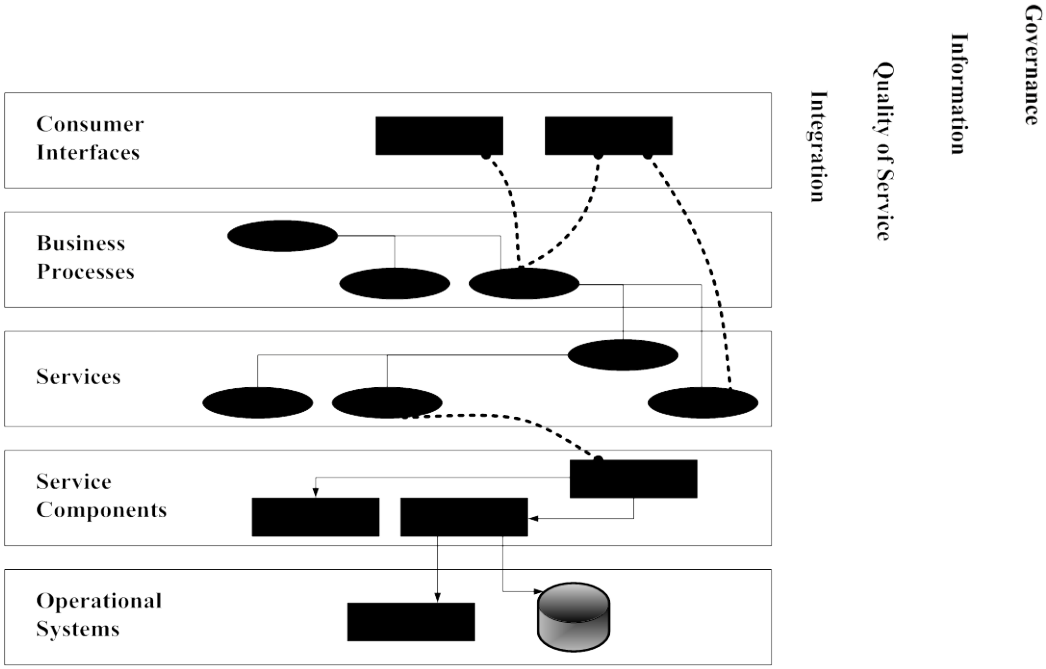


Figure 1. TOGAF - SOA reference architecture

These hypotheses apply to all phases of the systems lifecycle – particularly for operation and maintenance.

The following approaches on SOA measurement where introduced by Dieter Masak in [4] advancing the idea to adapt some measuring and analyzing strategies from other branches of science to the field of IT considering complex IT systems as digital ecosystems [5].

The growth model view: Ecosystems can be characterized as areas with constant changes and growth. These common principles and characteristics to IT are defined in Lehman's laws of software evolution [6]. An answer to the driving question of software system's growth or gradual progress can be sought in modeling this aspect to obtain a future forecast. The modeling of this into commercial software systems can be found within the Turski-Lehman growth model [7], equation as below.

$$\frac{\partial \psi}{\partial t} = c_1 \psi + \frac{c_2}{\psi^2} \qquad (1)$$

Here ψ constitutes a function of time *t* and c_1 and c_2 are constants, which varies depending on the considered software.

The entropy-based view: No clear and standardized definition is set to the term complexity; furthermore the understanding is often used differently in the respective science branches tough some approaches for a unified definition or abstraction were started [8].

One approach of measuring a complex system with a strong varying structure to runtime is making a conclusion by analogy to elements of gases or fluids in thermodynamics. They share similar attributes; whereby the exact state to an exact timestamp within either SOA systems or thermo dynamics systems are not well known.

The states of such systems in physics and in thermodynamics were analyzed using Entropy according to Bolzmann [9] for example; considering the amount of elements (size of fabric capacity) and their possibilities of interaction (gaseous, fluid, and solid). The transition from classical thermodynamics to information theory was covered by Shannon [10].

$$S = -\sum_{j=1}^{N} p_j \log_2 p_j \tag{2}$$

The Entropy S represents the measurement of a system's disorder. p_j Enunciates the probability of an event or state j and were described by a numeric value in detail. The entropy can be adapted to a variety of distributions.

The diversity model view: One more possibility to describe complexity is the consideration of the elements quantity related to a system. Another approach for this is the Diversity which can be derived from the Renyi-Entropy as in equation (3).

$$S_\alpha(X) = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n} p_i^\alpha\right) \tag{3}$$

The consideration of these states to logarithm base two is founded in Shannon's transition of Entropy to information theory. Furthermore this has some additional advantages like the possibility to add diversity of disjoint systems directly [2] as well as the relativity of effects in the case of adding states or services to a large scale system.

3 A Field Test of the New Approach

The background of the field test system is in the branch of IT service provisioning in the context of messaging and collaboration services within multi-tenant infrastructures. This system maintains approximately 9 million objects and about 30 thousand operations per hour across a globally dispersed platform. The following measurements results were anonymized and the systems working title is "S1"; the underlying architecture is based on the SOA reference model where the 10 different business domains are aligned crosswise. The current key figures show the state of the system after this fourth release and are listed in the table below.

Table 1: Key facts of the measured system S1

SOA Layer	Component Measurement		
	Components	***Services***	***LOC***
Interaction Layer	4 Major Components	65	393.655
Process Layer	5 Major Components	139	238.168
Functional Layer	7 Major Components	82	556.895
Operational Layer	2 Major Components	67	284.621

The measurement was run from the beginning of the system's development till a fourth major release. Furthermore, the mainly horizontal measurement of diversity and entropy in the business layers was performed continuously during operation which started on release 2.0.

The following figure shows that the process layer owns the largest growth and dynamics while the other layers, specially the lower ones, distinctly grow slower and especially tend to be stable in release 4.0 while the process layer continuous to growth.

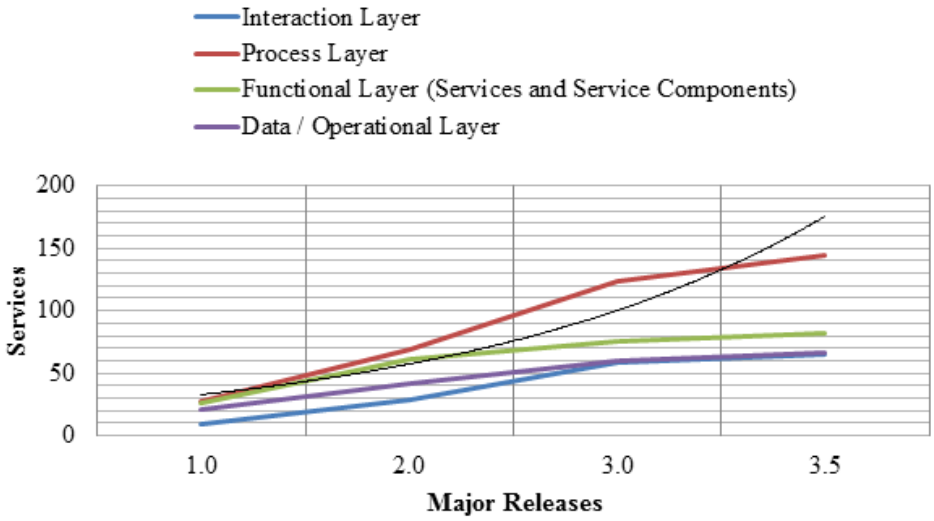


Figure 2. Growth Model of “S1” till fourth Major Release

Considering to the hypotheses above, we can derive that this measurement indicates the successfully implementation of favorable SOA characteristics like business process flexibility and service reusability. Due to the fact that this layer might be an interesting field of research the three remaining measurement results selected in this paper are dedicated to this layer and not to the overall systems.

The measurement of diversity based in the following diagram is dedicated to the process layer of the SOA system. The next figure shows the diversity of the workflow domains and their methods over the four major releases.

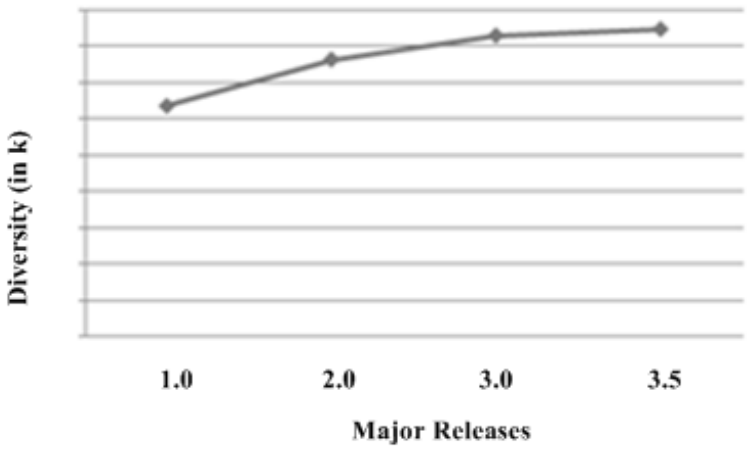


Figure 3. Diversity of “S1” process layer

The diversity which needs to be steadily growing according to the laws of software evolution is stable and does not have an exponential effect to the growth model. This can be taken to mean that the design of the process layer is well designed and that the system’s complexity is controllable.

The measurement of entropy based measures in the following diagram are dedicated to the process layer of the SOA system or rather the workflow domain and their methods and in relation to the entropy of the system overall. To outline the relation of another representative metric the range of functions (services) is shown to each bar graph. In this context disorder needs to be understood as a leak of structure or consistent path (orchestrations) of the service calls.

The following graph represents the varying values on entropy and shows the overall low system entropy in relation to the high dynamic process layer. Some dedicated workflow domains show differing values of entropy which are either higher or lower than those shown below.

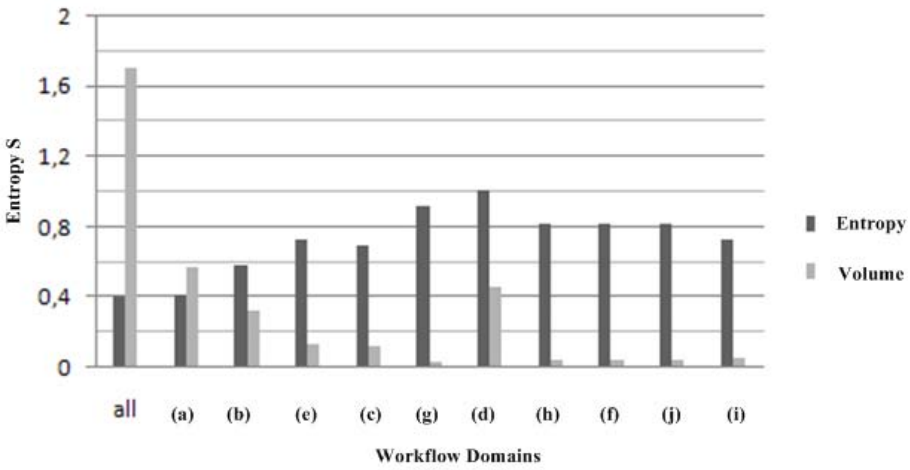


Figure 4. "S1" entropy overall and workflow domains

The graph can thus be used to analyze design characteristics of domains with low entropy values with possible adaptation to domains with a higher value. Although higher values could be caused by larger growth spurts due to the law of supply and demand in the business processes.

4 Summary

A new approach of SOA quality measurement based on [4] and the idea to consider today's growing IT landscapes and their continuous cross-linking as digital ecosystems [5] were discussed. SOA in context of enabling methodology for "Cloud Computing" still has an important role in IT. Some major goals of both like reduced complexity and advanced business flexibility should be objectives to measure quality and favorable characteristics. The described measurement field test was only on an experimental base but could showed the existence of favorable SOA characteristics of business flexibility and a form of controllability thus continuous growing. Furthermore it revealed some significant findings in probable not optimal designs of certain business process domains.

Some initiatives particularly to "Cloud Measurement" and standardization can be found in the Service Measurement Index (SMI) of the Cloud Service Measurement Initiative Consortium (CSMIC) [11].

References

[1] The Open Group: “TOGAF (The Open Group Architecture Framework) and the available SOA Source Book”, http://www.opengroup.org/soa/source-book/ra/perspec.htm#_A_High-Level_Perspective, June 2011.

[2] The Open Group : SOA Source Book; Van Haren Publishing, ISBN: 978-0-060-92587-1, pp. 34-37, 2009.

[3] R.Reussner, W. Hasselbring, “Handbuch der Software-Architektur“, dpunkt.verlag; ISBN: 978-3-898-64559-1, pp. 133-134, 2009.

[4] D. Masak, “SOA? - Serviceorientierung in Business und Software“, Springer-Verlag, ISBN: 978-3-540-71871-0, pp. 360-364, 2007.

[5] D. Masak, “Digitale Ökosysteme”, Springer-Verlag, ISBN: 978-3-540-79129-4, pp. 209-241, 2009.

[6] M. M. Lehman, “Laws of Software Evolution Revisited”, Department of Computing, Imperial College, London, pp. 1-4, January 1997.

[7] M. M. Lehman, J. F. Ramil, P. D. Wernick, D. E. Perry and W. M. Turski. Metrics and laws of software evolution— the nineties view, In Proc. of the Fourth, Intl. Software Metrics Symposium (Metrics’97), Albuquerque, NM, 1997.

[8] M. Flückiger, M. Rauterberg, “Komplexität und Messung von Komplexität“, ETH Zürich - Institut für Arbeitspsychologie, Technical Report IfAP/ETH/CC-01/95, pp. 4-8, 1995

[9] W. Ebeling, J. Freund, F. Schweitzer, “Komplexe Strukturen: Entropie und Information”, Teubner Verlag, ISBN: 978-3-8154-3032-3, pp. 29-39, 1998.

[10] C. E. Shannon, “A Mathematical Theory of Communication”, The Bell System Technical Journal,Vol. 27, pp. 379–423, 623–656, July, October, 1948.

[11] Cloud Commons: “The Details behind the Service Measurement Index“;http://www.cloudcommons.com/servicemeasurementindex/-/asset_publisher/M69c/content/the-details-behind-the-service-measurement-index], November 2010.

Büren, G.; Dumke, R.R.; Münch, J.:

MetriKon 2011
Praxis der Softwaremessung

Tagungsband des DASMA Software Metrik Kongresses
17.-18. November 2011, Kaiserslautern

Shaker Verlag, Aachen, 2011 (210 Seiten)
ISBN 978-3-8440-0557-8
ISSN 1618-7946

The book includes the proceedings of the MetriKon 2011 held in Kaiserslautern in November 2011, which constitute a collection of theoretical studies in the field of software measurement and case reports on the application of software metrics in companies and universities.

Schmietendorf, A.; Simon, F.:

BSOA 2011

6. Workshop Bewertungsaspekte serviceorientierter Architekturen

15. November 2011, Köln

Shaker Verlag, Aachen, 2011 (108 Seiten)
ISBN 978-3-8440-0503-5
ISSN 1867-7088

Seit nunmehr 6 Jahren beschäftigt sich die BSOA-Initiative mit der Bewertung von serviceorientierten Architekturansätzen. Zunächst beschäftigten sich die Teilnehmer im Rahmen der ersten Workshops mit der messtechnischen Erfassung der mit einer SOA einhergehenden Ausprägungen und Merkmale bzw. den involvierten Stakeholdern. Sehr schnell wurde deutlich, dass sich eine SOA weniger auf technologische Sachverhalte bezieht als vielmehr auf die veränderte Sichtweise zur Gestaltung unternehmensweit genutzter IT-Systeme. Erwartete Vorteile einer SOA bezogen sich insbesondere auf die Zielstellungen des Informationsmanagements. In diesem Zusammenhang wurden Mehrwertpotentiale durch eine verbesserte Geschäftsprozessorientierung der IT, reduzierte Daten- und Funktionsredundanzen, verringerte Komplexitäten bei Anwendungen und Schnittstellen, verringerte Kundenbindungen oder auch die Flexibilität mit der eine benötigte IT-Lösung bereitgestellt werden kann, ausgemacht.
Aus der Vielzahl an eingereichten Beiträgen konnte durch das Programmkomitee eine anspruchsvolle Agenda zusammengestellt werden.

IWSM-MENSURA 2011

The Joint Conference of the 21st International Workshop
on Software Measurement (IWSM) and the 6th International Conference
on Software Process and Product Measurement (MENSURA)

November 3-4, 2011, Nara, Japan

IEEE Computer Society Los Alamitos, California, Washington, Tokyo, 2011 (324 Seiten)
ISBN 978-0-7695-4497-7

This proceedings includes the full papers and the short papers of the Joint Conference of the 21st International Workshop on Software Measurement (IWSM) and the 6th International Conference on Software Process and Product Measurement (MENSURA).

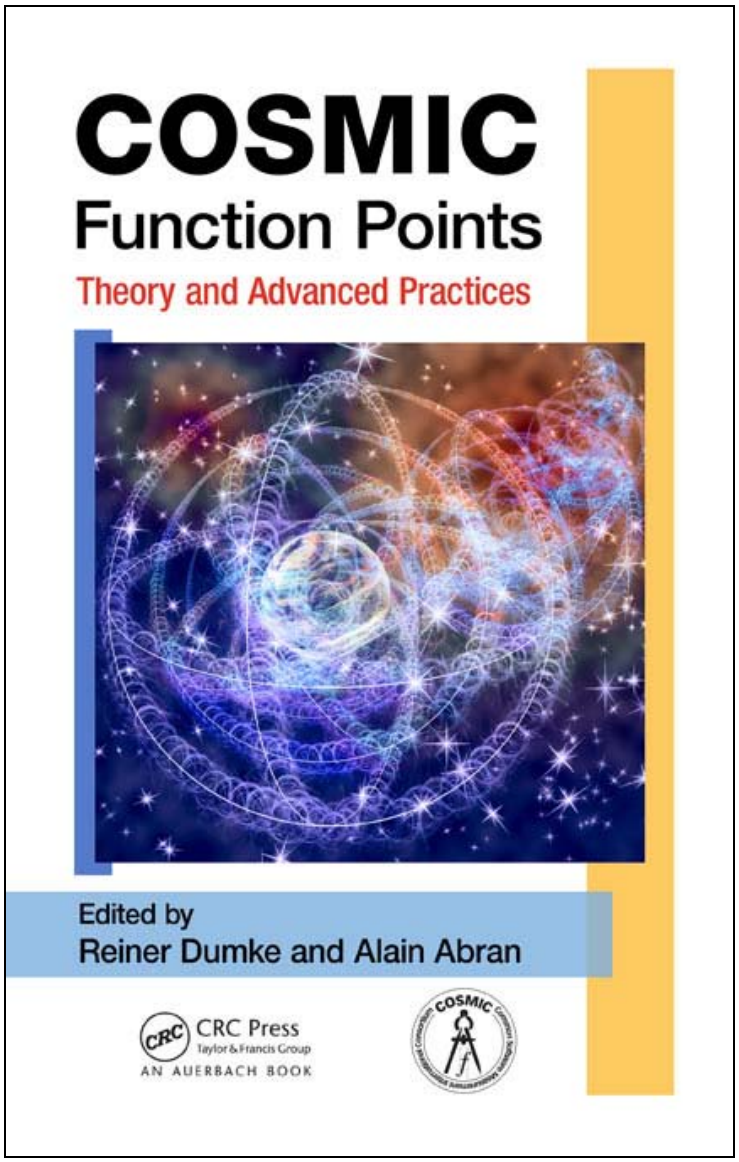
Dumke, R.; Abran, A.:

Cosmic Function Points

Theory and Advanced Practices

CRC Press Taylor & Francis Group, 2011 (334 Seiten)
ISBN: 978-1-4398-4486-1

This book has the following characteristics: the theme is about a new software size estimation method including their scientific and practical background; the chapters are based on papers, that would be published in our conference proceedings during the last six years; the authors are wellknown participants of the international software measurement community (see e. g. COSMIC, IFPUG etc.) and the book content is structured in the main problems of building new measurement or estimation methods in general and should be interesting for the software quality.



SWQD 2012:

Software Quality Days
January 17-19, 2012, Vienna, Austria
see: **http://www.software-quality-days.at/**

SEPG 2012:

24th Software Engineering Process Group Conference
March 12-15, 2012, Albuquerque, New Mexico, USA
see: **http://www.sei.cmu.edu/sepg/na/2012/**

EASE 2012:

International Conference on Empirical Assessment in Software Engineering
March 14-15, 2012, Ciudad Real, Spain
see: **http://alarcos.esi.uclm.es/ease2012/**

REFSQ 2012:

17th International Working Conference on Requirements Engineering Foundation for Software Quality
March 19-22, 2012, Essen, Germany
see: **http://www.refsq.org/2012/**

CSMR 2012:

15th European Conference on Software Maintenance and Reengineering
March 27-30, 2012, Szeged, Hungary
see: **http://csmr2012.sed.hu/**

STAREAST 2012:

Software Testing Analysis & Review Conference
April 5-20, 2012, Orlando, FL, USA
see: **http://www.sqe.com/stareast/**

SERA 2012:

10th ACIS Conference on Software Engineering
April 16-18, 2012, Kunming, China
see: **http://conference.researchbib.com/?eventid=13270**

iqnite 2012:

Software Quality Conference
April 24-26, 2012, Düsseldorf, Germany
see: **http://www.iqnite-conferences.com/de/index.aspx**

AGILE 2012:

12th International Conference on Agile Software Development
April 24-27, 2012, Avignon, France
see: **http://irtg.ifgi.de/agile-2012-call-for-papers/**

QEST 2012:

International Conference on Quality Engineered Software and Testing
April 30 - May 4, 2012, Chicago, USA
see: **http://www.qaiquest.org/2012/**

PSQT 2012:

International Conference on Practical Software Quality & Testing
West: May 7-11, 2012, Las Vegas, USA
see: **http://www.psqtconference.com/2012west/index.php**

SPICE 2012:

SPICE Conference
May 29-31, 2012, Palma de Mallorca, Spain
see: **http://www.spiceconference.com/**

ICSE 2012:

International Conference on Software Engineering
June 2-9, 2012, Zurich, Switzerland
see: **http://www.ifi.uzh.ch/icse2012/**

SEPG Europe 2012:

Software Engineering Process Group Conference
June 5-7, 2012, Madrid, Spain
see: **http://www.sei.cmu.edu/sepg/europe/2012/**

ICPC 2012:

20th International Conference on Program Comprehension
June 11-13, 2012, Passau, Germany
see: **http://icpc12.sosy-lab.org//**

PROFES 2012:

12th International Conference on Product Focused Software Process Improvement
June 13-15, 2011, Madrid, Spain
see: **http://www.grise.upm.es/profes2012/PROFES_2012/Welcome.html**

IASTED SE 2012:

IASTED International Conference on Software Engineering 2010
June 18-20, 2012, Crete, Greece
see: **http://www.iasted.org/conferences/home-780.html**

ICWE 2012:

International Conference on Web Engineering
June 23-27, 2012, Berlin, Germany
see: **http://icwe2012.webengineering.org/**

SMEF 2012:

Software Measurement European Forum
June 25-26, 2012, Rome, Italy
see: **http://www.iir-italy.it/smef2012eng**

ENASE 2012:

6th International Conference on Evaluation of Novel Approaches to Software Engineering
June 28- July 1, 2012, Wroclaw, Poland
see: **http://www.enase.org/**

UKPEW 2012:

24th Annual United Kingdom Workshop on Performance Engineering
July 2-3, 2012, Edinburgh, UK
see: **http://aesop.doc.ic.ac.uk/conferences/ukpew/**

ISSTA 2012:

International Symposium on Software Testing and Analysis
July 16-20, 2012, Minneapolis, USA
see: **http://crisys.cs.umn.edu/issta2012/**

QSIC 2012:

12th International Conference on Software Quality
August 27-28, 2012, Xian, China
see: **http://www.di.univaq.it/qsic2012/**

ICGSE 2012:

7th International Conference on Global Software Engineering
August 27-30, 2012, Porto Alegre, Brazil
see: **http://conference.researchbib.com/?eventid=13270**

ASQT 2012:

Arbeitskonferenz Softwarequalität und Test
September 6.-7., 2012, Klagenfurt, Austria
see: **http://www.asqt.org/**

CONQUEST 2012:

13. International Conference on Software Quality
September , 2012, Nuremberg, Germany
see: **https://www.isqi.org/de/isqi-news/items/conquest-in-nuernberg.html**

ESEM 2012:

International Symposium on Empirical Software Engineering & Measurement
September 20-21, 2012, Lund, Sweden
see: **http://www.esem-conferences.org/**

UKSMA 2012:

Annual UKSMA Conference - Managing your Software (through Measurement)
October , 2012, London, UK
see: **http://www.uksma.co.uk/**

IWSM/Mensura 2012:

Common International Conference on Software Measurement
October 17-19, 2012, Assisi, Italy
see: **http://iwsm2012.wordpress.com/**

BSOA 2012:

7. Workshop Bewertungsaspekte service-orientierte Architekturen
November 15, 2012, Desden, Germany
see: **http://www-ivs.cs.uni-magdeburg.de/~gi-bsoa/**

MetriKon 2012:

International Conference on Software Measurement
November 7 - 9, 2012, Stuttgart, Germany
see: **http://www.metrikon.de/**

ICPE 2013:

International Conference on Performance Engineering
Spring 2013
see: **http://icpe.ipd.kit.edu:80/call_for_proposals/**

see also: OOIS, ECOOP and ESEC European Conferences

SOFTWARE MEASUREMENT NEWS

VOLUME 17 2012 NUMBER 1

CONTENTS

Announcements **3**

Workshop Report **7**

Position Papers **17**

Hussein, A. A.:
An Enhanced Security Approach for Securing and Validating Enterprise Business Processes *17*

Janus, A.:
Der 3C-Ansatz für Agile Qualitätssicherung *30*

Günther, D.:
Quantitative Approach of IT Security Management Processes *35*

Massoud, A.:
Efficiency in Integrated Legacy Systems based-SOA *42*

Fiegler, A.:
SOA Measurement for IT Process Improvement *51*

New Books on Software Metrics **57**

Conferences Addressing Metrics Issues **59**

ISSN 1867-9196