Applying the ISRQCC Method in a Web Reengineering Process. The SwissInsurances (SWI) Web Engineering Audit

Antoine Trad[1], Damir Kalpić[2] and Catherine Trad[1]

[1]TradSoft GmbH, Hettlingen, Switzerland
[2]Faculty of Electrical Engineering and Computing, Zagreb, Croatia

The most important factor in maintaining Web Engineering Information Systems (WEIS) lifecycle is the reengineering process that in most cases is due to serious problems and hence high maintenance costs; this process must be closely monitored by audit procedures. Once the WEIS was put into production, the maintenance and reengineering considerations started, and the Information System Risk and Quality Check Coefficient (ISRQCC) method was periodically used to audit it. It also established the relationship between the (re)design and the reengineering process. This paper presents the application of the ISRQCC method in the SWI's Web Engineering (WE) project, where the audit operations resulted in a major reengineering process.

Keywords: ISRQCC, monitoring, audit, reengineering, (re)design, maintenance, risk.

1. Introduction

The ISRQCC audit method is intended to be of interest to non-technical auditors, audit managers, audit committee members, senior managers in charge of critical computing systems such as WE systems, executives, board members, and even seasoned IS auditors. This audit method results in a **heuristic model** which will help information system auditors to estimate the probability that an IS will succeed or fail [8]. Furthermore the ISRQCC offers the possibility to forecast IS problems, adjust the project management pathways and define the problem's source(s) as well as its possible solution(s). Although we have many tools and standards [11] for designing and implementing most of the IS components, until today we still do not have applicable interactive tools, methods or theories in the areas of estimating and auditing of risks, costs, feasibility, viability and hence quality of complex IS. The ISRQCC concentrates on the project **feasibility** and on finding the set of possible solutions for the current and future problems. This paper presents the use of the of *ISRQCC* audit method to audit the SWI WE project, and help the project manager and designer in establishing competitive maintenance and stabilisation procedures. The *ISRQCC* was used also to define future evolutionary steps; and to avoid the blind and risky method of *"let's re-develop"* the whole system. On the contrary, it reused the existing WEIS [25] as much as possible.

2. The web engineering project

2.1. General

The target of the WE project is to host the SWI 'DeMilitarized Zone' (DMZ) with a hosting provider. The WE project has five major objectives:

- The WE infrastructure to be hosted by an external Internet Service Provider (ISP),
- Implement a WE topology designed by the SWI security engineering,
- Have all the infrastructure closely monitored,

- Have backups performed by the ISP,
- Build an engineering and architecture team.

2.2. The topological view

The Hosted DMZ is based on the Web Frontal Server (WFS) topology design (as presented in Fig. 1). This WFS topology describes a secure topology for running applications in different security zones.

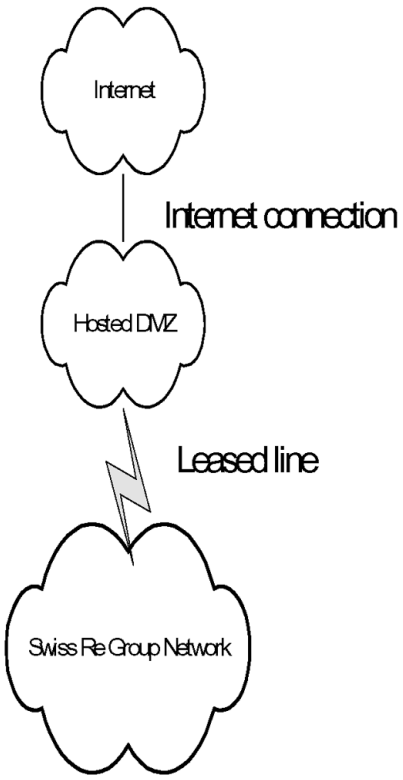

Fig. 1. The DMZ, hosted by the ISP, has a leased line connection to the SWI Group Network for management and data exchange access; and the Hosted DMZ is connected to the internet.

The WFS is SWI's platform to securely and efficiently deploy applications on the Internet. The scope of these applications reaches from purely informational HTML pages to fully integrated, transactional business-to-business processes. The WFS creates a secure environment protecting its applications from external and internal attacks and also from each other. It enforces authentication before allowing access to the applications. To this end, it provides a central login process using secured identification or plain username/password formats, depending on the application's needs. In addition, WFS also handles all encryption tasks to and from the Internet. The WFS allows communication over HTTP and CORBA, both of which can be encrypted using S|SL.

The *WEIS* audit started with the *Decision Making Process (DMP).* When the decision was made to reengineer the system, then the *Choosing of Evolutionary Technology (CET)* yielded the resultant technological solutions for the WEIS problems. *CET* defined the corrections to be made to the system's architecture and the technical guidelines. After the *CET* process, the *Consolidation Process (CP)* was executed. During this process, the system was stabilised and documented. The next step was the *Domain Analysis Process (DAP)* where the *ISRQCC* team analysed the business requirements. It was highly recommended to include experienced end-users in this process. The *DMP, CET, CP* and *DAP* steps are considered as preparations for the *System Reengineering Process (SRP)* which is the real transformation step. During this step the system was reengineered according to the *CET* guidelines. After the *SRP's* execution, the *ISRQCC* recommends the *Evaluation of the Reengineering Process (ERP).* During this step the *ERP* applied the *ISRQCC's* audit module *Audit View (AUDV).* Depending on the success of the *ERP's* outcome, the team had to check whether the final goal had been reached. Otherwise, we recommended the use of the *ISRQCC's* audit component *AUDV* to correct the malfunction [25]. If the goal was reached, then the whole task was declared as successfully completed.

3. ISRQCC's application

3.1. Decision making process *(DMP)*

During this process the audit personnel analysed the WEIS using the ISRQCC's DMP and decided to reengineer it. The reasons for reengineering were:

- Very poor system stability, hence frequent downtimes (outages) [13],
- Very high maintenance costs [9],
- No architectural blueprints hence system evolution concept was inexistant. Added to that, nobody had the global system overview [17],

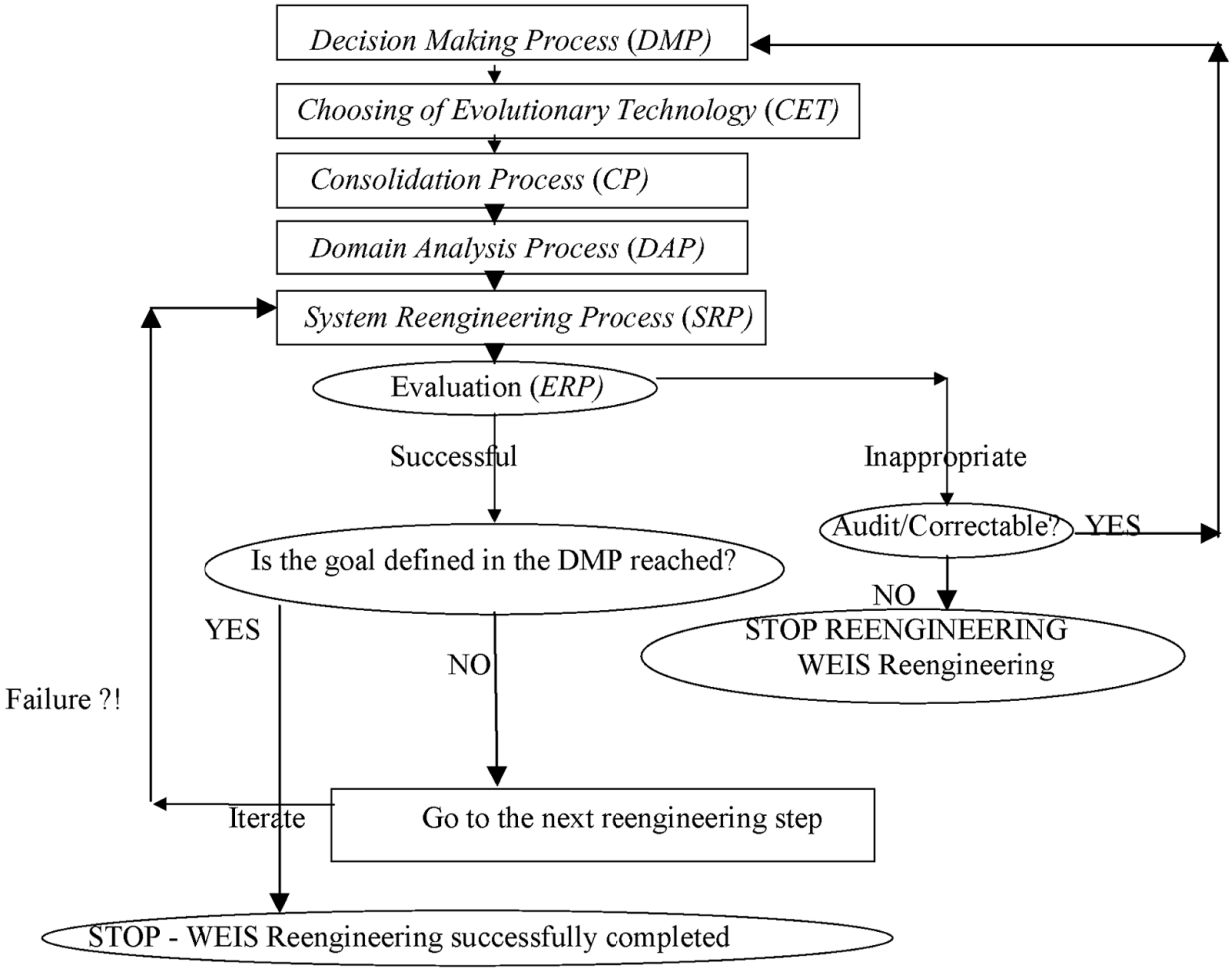

Fig. 2. The ISRQCC's SWI WEIS audit.

- Faulty engineering processes. As the system architecture was inexistent most of the engineering processes were not defined.

The *"Team Design and Quality Check Coefficient"* was used to estimate whether the team was appropriate for the job, and recommended the creation of an architecture and engineering team [18]. All the decisions and estimations were based on the ISRQCC's DMP and Factors Estimation System (FES) [26].

3.2. Choosing an evolutionary technology *(CET)*

The *ISRQCC* audit method is technology-independent, and in this case was applied in a highly distributed WE environment. The use of object-oriented paradigm allowed the production of highly reusable software components at a higher quality standard and a better coordination between different project teams [7]. The *ISRQCC* audit method recommended the Object-Oriented *(OO)* decomposition in the *DAP* and the *SRP*. During *CET*, the *Requirements Engineering Process* was established and a direct map of the resulting model was reused in the *OO* analysis and design phases [10]. As the system was to be reengineered, an iterative migration from the actual 'language only' OO model towards a 'designed' OO was recommended [25].

The migration steps were:

- Early steps, included preparations, such as reviewing the Requirements Engineering, OO Analysis (OOA), OO Design (OOD) and OO Programming (OOP) documentation [7],
- *Graphical User Interface (GUI)* transformation steps, using the Model View Control (MVC) model [29],

- *Code* transformation steps, moving code to OOA and OOD models [21],
- *Data* transformation steps, moving to an OO Relational Mapping Model [1],
- *Communication* transformation steps, using the MVC model,
- *Monitoring* transformation steps, using the FES module [22].

The chosen and reviewed technologies:

- The use of *Unified Model Language* was recommended by the auditor [5],
- The *Sun Workshop C++* [20] was introduced by the audit team to support the development of the standard panic procedures,
- The *Java Development Kit* was used to develop client web interfaces [12][4],
- The *HTML* was used to create client web interfaces [16],
- The *Oracle* database server was used for data persistency [3],
- The *Perl* programming language was used to develop scripts [30],
- The *IBM Websphere* was used for web application transaction management [4],
- The *IONA Corba* was used to access mainframes [28],
- The *Sun Unix Servers* were used as 2n̂d tier processing servers [31],
- The *Sun Cluster 3* was used for ensuring fail-over scenarios [19],
- The *Veritas Volume Manager* was used for insuring disk availability [27],
- The *Altheon Switches* were used for ensuring load balancing on the servers [2],
- The *Nokia Firewalls* were used for security activities [15],
- The *Cisco Routers* were used for routing activities [6],
- The *Apache Web Server* was used as a web server [14].

3.3. Consolidation process *(CP)*

Once the decision had been made to reengineer the system using the results of *CET (that is mainly based on the FES's information estimations),* then the consolidation process started. The CP's main aims were:

- To stop (freeze) any development and to prevent from adding any component to the WEIS,
- To avoid any platform modification (operating systems versions, hardware installations...) in this project, which was very important because of the platform complexity,
- To verify the system's documentation (the system was actually over-documented and the documentation had to be reduced to a manageable set).
- To convince the management [23] of the usefulness of such operations.

3.4. Domain analysis process *(DAP)*

The DAP's process's aim was to audit the business requirements and the corresponding system implementation. The end-users assisted the verification operations. The *DAP* team concentrated on the IS functionalities and stopped the introduction of any new requirement.

3.5. System reengineering process *(SRP)*

This section presents the SRP that was started after *DAP* and its main objective were:

- Audit the system (including code); and bring it to high quality in production [7],
- Define basic categories and components,
- Define the project conventions and interfaces to the various components to *FES.*

3.6. Evaluation of the reengineering process *(ERP)*

The *ERP* was executed after the SRP. Once the phases *DMP, CP* and *DAP* were completed, the audit and reengineering process could not modify any of the IS components. The *ERP* evaluated the reengineering audit results, and if they were successful, then we either tried to further enhance our process or stop and declare final success. Otherwise if major redesign had been necessary, we would have recommended the following steps:

- To rebuild the system and reuse the *healthy* components (if any were available),
- To apply a full IS audit in order to understand what caused the IS failure [23]

4. ISRQCC factors and views

The *ISRQCC* audit method is composed of categories of factors, where each category can contain one or more of these factors [23]. This pool of factors plays a major role in determining the IS quality status [24], where each factor identifies the type of problem, which in turn results in a corresponding action(s). The ISRQCC factors are independent of all the ISRQCC **Components** or **Views,** but one view is defined as a set of one or more factors. For the WEIS project the significant factors were determined using the ISRQCC's FES. Analysing the FES the auditors filtered the serious problems and hence defined the factors that influenced their appearance. This proves that the ISRQCC is not just another theoretical audit method, which localises problems using subjective estimations, where the FES is the nucleus tool that monitors the system in production [26].

5. Conclusion

In line with the expansion of the Internet information technology trends, the WEIS evolved towards a very complex architecture. Consequently, the developed system needed to be constantly upgraded and aligned with the state of the art technology. As in most of other projects, the WEIS was built *just to work* and the management considered that no extra funds are to be allocated to improve its status. This caused serious problems in production that in turn implied long downtimes, that resulted in significant budget losses. The *ISRQCC* audit method evaluated the current WEIS and proposed a set of actions to be taken in order to *improve* the current system. Its main goal was to avoid complete system rebuild; and recommended an iterative and cautious method of conversion to a *liveable* system; to support such a complex process, a monitoring system known as FES was built [26]. The *ISRQCC* auditor with the project's designer help defined the future WEIS technical and domain requirements in order to reengineer it in careful paces and reach the requested stability; therefore this audit work was considered as WEIS redesign.

References

[1] S. AMBLER, editor. *Mapping objects to relational databases*. USA: Ronin International-IBM developerWorks 2000.

[2] Alteon WebOS Software, Command Reference, CA-USA: Alteon Systems; 2000.

[3] S. BOBROWSKI, editor. *ORACLE 8 Architecture*. CA, USA: Osborne, McGraw-Hill; 1998.

[4] K. BROWN, *Enterprise Java (tm) Programming with IBM Websphere*. USA: Addison Wesley Professional; 2001.

[5] R. BURKHARDT, editor. *UML Unified Modelling Language*. USA: Addison Wesley; 1997.

[6] Cisco Systems. Cisco Network Module Guide – for Cisco 2600 and 3600 Series Cisco Systems Inc. USA San Jose: 2000.

[7] DAVID M. PAPURT, editor. *Inside the Object Model – The Sensible Use of C++*. USA: Sigs Books; 1995.

[8] D. DAYTON, editor. *Information Technology Audit Handbook*. USA: Daytonassociates 1999.

[9] Y. DERRIEN, editor. *Les techniques de l'audit informatique*. France: DUNOD; 1992.

[10] H E. ERIKSSON AND P. MAGNUS, editors. *UML Toolkit*. USA: John Wiley & Sons, Inc; 1998.

[11] ISACA-S, editor. *Standards For Information Systems Auditing*, USA: ISACA; 1998.

[12] J. JAWORSKI, editor. *Java 2 Platform Unleashed*, Indiana, USA: SAMS Publishing; 1999.

[13] JONES, editor. *Applied Software Measurement: Assuring Productivity and Quality,* USA: McGraw-Hill; 1996.

[14] B. LAURIE, *Apache: the Definitive Guide*, CA, USA: O'Reilly & Associates; 1999.

[15] A. MCNAB, *Essential Checkpoint Firewall-1: An Installation, Configuration, and Troubleshooting Guide by Firewall*. USA: Pocket Books; 2001.

[16] D. OLIVER, M. HOLZSCHLAG, editors. *Teach yourself HTML 4*. Indiana USA: SAMS net; 1997.

[17] R. PRESSMANN, editor. *Software Engineering – A practitioner's approach/Fourth Edition*. USA: McGraw Hill International Editions; 1997.

[18] J. RYMER, *Optimising Software Teamwork – The Process and Technology of Healthy Application Development Teams*. USA: Rational Software Corp; 1999.

[19] Sun. Sun Cluster Architecture – Technical Overview. CA, USA: SunMicrosystems; 2000.

[20] SunMicrosystems. The Workshop development C++. Palo Alto, California USA: Sun Microsystems Inc.; 2001.

[21] A. TRAD, *Risk Estimation In Building Object Oriented Systems*, MSc Thesis. Zagreb, Croatia: Faculty of Electrical Engineering and Computing; 1995.

[22] A. TRAD, *Information System Risk and Quality Check (ISRQCC)*, Doctoral Thesis. Zagreb, Croatia: Faculty of Electrical Engineering and Computing; 2002.

[23] A. TRAD, *The Absolute Monitoring Method – AMM*. Kloten, Switzerland: SAirGroup; 1997.

[24] A. TRAD, Swissair. *The Unified Stability Model*. Kloten, Switzerland: SAirGroup; 1995.

[25] A. TRAD, Reengineering Quality and Risk Check (RQRC) – Theoretical Basis. In Kalpić, D, editors. *Proceedings of the 23rd International Conference on Information Technology Interfaces*; 1999 Jun 15–18; Pula, Croatia. Zagreb: SRCE University Computing Center, University of Zagreb; 1999. pp. 497–502.

[26] A. TRAD, Proactive monitoring of the information system risk and quality. In Kalpić, D, Fertalj, K., editors. *Proceedings of the 23rd International Conference on Information Technology Interfaces*; 2002 Jun 24–27; Dubrovnik, Croatia. Zagreb: SRCE University Computing Center, University of Zagreb; 2002. pp. 497–502.

[27] Veritas. Volume Manager – Storage Administrator 3.2 – Administrator's Guide. California, USA: Veritas Software Corporation; 2001.

[28] A. VOGEL, B. VASUDEVAN, M. BENJAMIN, T. VILLALBA, editors. *C++ Programming with CORBA*. Canada: John Wiley & Sons Inc.; 1999.

[29] P. VON ZIMMERMANN, editor. *Das MVC-Modell*, Germany: SAP AG; 1992.

[30] L. WALL, T. CHRISTIANSEN, J. ORWANT, editors. *Programming Perl (3rd Edition)*. CA, USA: O'Reilly & Associates; 2000.

[31] P. WATTERS, VEERARAGHAWA, editors. *Solaris – The Complete Reference*. USA: McGraw-Hill; 2000.

Received: June, 2002
Accepted: September, 2002

Contact address:

Antoine Trad
TradSoft GmbH (Switzerland)
Schulstrasse 7c, 8442 Hettlingen, Switzerland
e-mail: antoine.tradsoft@bluewin.c|h

Damir Kalpić
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
e-mail: damir.kalpic@fer.hr

Catherine Trad
TradSoft GmbH (Switzerland)
Schulstrasse 7c, 8442 Hettlingen, Switzerland
e-mail: catherine.tradsoft@bluewin.c|h

ANTOINE TRAD received the Dipl.Ing. and M.Sc degrees in information and computer sciences from the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 1985 and 1995, respectively.

Since 1986 he has been working as an IT consultant and auditor for various companies in France and Switzerland. He is now managing TradSoft GmbH (Switzerland), that is specialized in information system building, reengineering and audit processes. His professional and current research interests and activities have been concentrated on problem solving and the audit of information systems in different business domains. Essentially, it is the optimisation of the *'Information System's'* (IS) building process and its introduction into production, assisted by an iterative audit and control methodology.

He has developed a corresponding methodology called the *'Information System Risk and Quality Coefficient'* (ISRQC). Presently, this method is being used to offer support and control through the assistance to companies facing difficulties, such as Swissair, Philips, Swiss Re, Winterthur Insurances, Ascom, Fotolabo and the WHO. He has published six papers about IS audit and risk management and applications programming systems.

CATHERINE TRAD received the BA degree from the KVB of St Gallen, Switzerland in 1988 . Since 1988 she has been working as a business consultant for various companies in Switzerland. She is now administrating TradSoft GmbH (Switzerland), that is specialized in information system building, reengineering and audit processes. Her professional and current research interests and activities have been concentrated on assisting the development of the *'Information System Risk and Quality Coefficient'* (ISRQC). Presently, this method is being used to offer support and control through the assistance to companies facing difficulties, such as Swissair, Philips, Swiss Re, Winterthur Insurances, Ascom, Fotolabo and the WHO.

DAMIR KALPIĆ is professor in the field of computing at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received his degrees of Dipl.Ing., M.Sc. and Ph.D. from the same institution. His professional interest and activity has been the application of computers in different fields. Primarily, it is the development of information systems, supported by database systems and extended with mathematical models stemming from operational research. From this field of activity, a scientific, professional and educational group, in which he is the senior member, offers consulting, education and software development to business, industry, administration and other institutions. He teaches Algorithms and Data Structures, Operational Research and Information Systems. He has mentored more than 150 graduation theses, 25 master's theses and 6 doctorates. He authored or co-authored about 30 internationally published articles. He has been member of numerous international conference programme committees and has led over 15 projects.