Supporting Architectural Design Decisions through Risk Identification Architecture Pattern (RIAP) Model

THAMER AL-ROUSAN, SHAHIDA SULAIMAN, ROSALINA ABDUL SALAM

School of Computer Sciences,
Universiti Sains Malaysia,
11800 USM, Penang,
Malaysia
{tamer72, shahida, rosalina}@cs.usm.my

Abstract: -Web projects tend to have a high possibility of loss or failure compared to traditional projects. For this reason, risk management is becoming more emphasized and systemized in Web projects so as to improve the quality of difficult decisions that normally encompass a higher level of risk exposures. Software architecture process is seen as iterative process and the amount of risk-related software architecture artifacts in each iteration of the process differ from that of other iterations. Each iteration of a process needs a unique decision-making process to accommodate certain risk factors. Since each iteration of Web project design has different types of risks in the decision-making process, a decision support system should be tailored to satisfy the specific needs of a particular iteration. In this way, various risks that arise through the life cycle of a project can be constantly checked and monitored. This research aims to support architectural design decision-making process through a risk identification architecture pattern model called RIAP. The model is anticipated to clarify high-level design process and to support active design decisions. Consequently, the software architecture becomes easier to communicate, maintain and evolve. Furthermore, it supports the analysis, improvement, and reuse of architectural design decisions process in future Web projects.

Key Words: - Risk identification architecture pattern, software architecture, architectural design

1 Introduction

Nowadays Web applications are full-fledged and complex software systems that provide interactive, data intensive and customizable services accessible through various devices. They provide facilities for transactions and storing data in the underlying database [1]. As both dependency and requirement for superior and more complex Web applications increase, the requirement for larger and more complex Web applications increases as well [2]. The main challenge in a Web project development is to provide a high quality and reliable solution that is within the schedule and budget. However, the main obstacle to solve these problems is the complexity in developing software [3].

Recently, the software engineering community has come to realization that with the growing complexity of software systems and the problems that they are trying to solve, software architecture is an important instrument that determines the accomplishment of a software development project [4]. There is no common definition for the term "architecture". Over the past few years several definitions of software architecture have been proposed. Bass [5] proposes a popular definition:

"*The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them*".

In software engineering design flaws and risks cause, approximately 50% of problems [5]. Risk management conventionally has been used in different fields, for instance, cost, safety, and time management in projects creation. Risk management application fields have also been extended to include other fields such as decision-making, feasibility studies, business studies, performance assessments, and emergency management through the expression of different factors that cover all stages of the project lifecycle [6]. Risk management is an essential part of the project management and it plays a significant role in achieving a good business and project outcome

[7]. Hence it can be applied generally in various areas of decision-making on construction projects.

In our previous work [8], we proposed the risk identification architecture pattern (RIAP) to manage risks in Web projects. RIAP describes the recurring generic risk solution scheme that occurred in a specific context before the design process starts based on reuse of knowledge gained from other projects that they already managed for similar events. Thus RIAP is perceived as a model to guide risk management of Web projects. In this paper, we will focus on design decisions in software architecture process by using the model.

Decision-making is "a process by which a person, group, or organization identifies a choice or judgment to be made, gathers and evaluates information about alternatives, and selects from among the alternatives" [9]. This definition implies that decision-making involves risks in selecting one from several courses of action, which is usually compounded by time and information constraints [10].

Software architecture process is an iterative process and the amount of risk-related software architecture artifacts in each iteration of the process differ from that of other iterations. Each one of iteration of a process needs a unique decision-making process to accommodate unique risk factors [11]. Knowledge on properties of single architectural patterns or style is not enough to be useful in a process of software architecture design. To make a clear decision on which patterns to use for a certain software system, risks associated with architectural patterns must be identified early. Different architectural patterns address different architectural concerns. Thus, risks should present information on the relationships between patterns and concerns [18].

Normally the potential risks in software architecture identified after the initial design stage, this happened during the quality evaluation of the software architecture. Performing a risk management before the design stage is a significant element of a solid software architecture or artifact. Accordingly, it play significant role in design process and supporting design decisions [8].

The following Section 2 describes the two main aspects concerned in the proposed model. Section 3 reviews RIAP development process. Section 4 explains the risk pattern architecture. Then we discuss the bridge between RIAP and design decision in Section 5 and its evaluation in Section 6. Lastly we conclude and describe the future work in Section 7.

2 Risk Identification Architecture Pattern (RIAP)

RIAP model is designed based on important factors influencing Web projects among organizations and stakeholders. RIAP are based on two key aspects as in Section 2.1 and 2.2.

2.1 Organizations and Stakeholders

The problems confronted by web project management are always related to providing the relevant stakeholders with a satisfactory solution within a certain schedule and budgetary limits. Stakeholders are people or organizations that have direct or indirect influence on the requirements in a system development. Stakeholders may include customers, content authors, domain experts, usability experts, users, marketing professionals, developers, project managers, maintainers, government, shareholders and others who are influenced by or have interest in the development of a product based on their requirements. Moreover, Web projects are also influenced by the business strategy, vision and structure of the developing organization[12].

The proposed risk management is to minimize the risk of not achieving objectives of a project, an organization and stakeholders who have the interest in the project as well as to identify and exploit opportunities.

2.2 Characteristics of Web Projects

Since Web applications differ from conventional software applications, we consider the characteristics not in traditional application such as non-linear navigation and characteristics that are of particular importance in Web applications as proposed in the existing literature [13],[14],[15]. These characteristics constitute the reasons why many concepts, methods, techniques, and tools of traditional software projects are either insufficient to meet the needs of Web projects or have to be modified in order to do so.

3 Development Process of RIAP

The development process of RIAP model is divided into two parts. The first part constitutes the theoretical definition. The main goal of the first part is to identify risk factors that threaten Web project development then assess the relevancy of each risk factor to the characteristics of the Web project that were obtained particularly for the Web project [8]. The second part is about managing operational risks and it concentrates on the utilization of Bayesian networks (BN) as a tool to explore the causal relationships between risk factors and its parent risk factor (see Fig. 1).

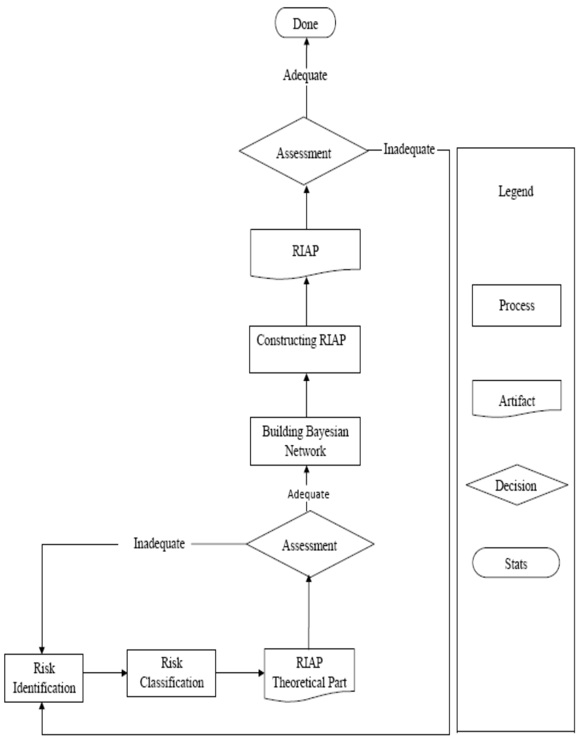

Fig. 1: The development process of RIAP

3.1 First Part: Theoretical Definition

The main goal of the first part is concerned with identifying what might go wrong when developing a Web project, and how these events will affect the project. The part has two phases: risk factors identification and risk factors classification.

3.1.1 Risk Factors Identification

The first step in the development of any model is to collect required information on threats to the project. This is derived by interviewing various experts from different Web application development companies, which develop Web applications for sale as well as in-house software development groups within organizations. It focuses on discovering possible risks of Web project developments and probable risk mitigation actions to prevent the risks. Then we identify the relationships among the possible risks and the success in Web project development processes.

Eight interviews were conducted with experts from diverse countries who develop Web applications for sale as well as in-house software development groups within organizations. These interviews resulted in twenty-one risk factors. A risk factor is identified as either a “not a success development risk” or a “not a success mode”. A “not a success development risk” is a risk factor that threatens the aspect of design in the developed process, which will not be developed in such a way to meet project requirements. For instance, of development risk factor is a build on emerging technology and methodology. A “not a success mode” is a failure that could happen once the system is designed or created example an employment failure.

The second step in the first phase involved conducted survey in which ninety-one participants from different Malaysian, American and Jordanian companies that develop Web applications for sale as well as in-house software development groups within organizations. The business that they serviced include government, financial institutions, Web portals, medical agencies, legal, banking, manufacturing, e-commerce retail, travel and tourism, software consultancy and consumer advisory services. The main objective of the study is to verify our results from first step and to understand the extent of Web project development practices currently in use by learning from the experiences of organizations who are already out there managing Web projects. In addition it investigates uncertainties that will affect the project and the level of threat they pose to the Web project’s success.

3.1.2 Risk Factors Classification

After reviewing all risk items captured during the interview process, the second stage was to assess the relevancy of each risk factor to the characteristics of the web project that were obtained particularly for the web project. These characteristics constitute the reasons why many concepts, methods, techniques, and tools of traditional software projects are either insufficient to meet the needs of Web projects or have to be modified in order to do so. These characteristics include Content, Navigation, Presentation, Social Context, Technical Context, Natural Context, The Development Team, Technical Infrastructure, Process, and Integration. The result of classifying risk to Web application characteristics is shown in Table 1.

Table 1: Risk factors classification

Characteristics	The Risk Scenarios
Content	1. No explicit definition about the standard of project quality. 2. Lack of understanding the structuring of content.
Navigation	3. Difficult to navigation and find problems. 4. Large volumes of information.
Presentation	5. Difficult of operation and simplicity. 6. Lack of aesthetics in content.
Social Context	7. Difficult in defining content and functional requirement. 8. Hard to term possible threats from competitors.
Technical Context	9. Complexity of designing models increases by using mobile devices. 10. Lack of understanding delivery medium concept.
Natural Context	11. Time and location from where the applications are accessed cannot be predicted. 12. Meet user's expectation to have accessibility around the clock, every day.
Development Team	13. Web developers have variety of background, experience and age. 14. Build on emerging technology and methodology. 15. Lack of development consideration on safety, security, reliability.
Technical Infrastructure	16. Immaturity of new techniques. 17. Hard to predict operational environment.
Process	18. Continually changing project/scope/objectives. 19. Lack of defined user categories.
Integration	20. Legacy systems are poorly documented. 21. Many external supplier involved in the development project.

3.2 Second Part: Transformation of RIAP Into an Operational Model

Identification and classification of the risk factors is only the first step of the RIAP development process. The second step is to use specific approach that can bring more structure and formalism to identify and predict risks associated with a Web project, and to which we can associate measures and criteria. We decide to use a probabilistic approach using Bayesian networks to transfer the theoretical part of RIAP into an operational model.

3.2.1 Bayesian Network

Bayesian network (BN) is a probabilistic graphical model that indicates an assumption between a group of variables and their probabilistic independence [16]. This network is a directed acyclic graph containing nodes and arcs. Nodes denote the random variables while directed arcs denote the probabilistic relationship between variables [16]. In this research, we use BN for three major reasons. The first is because of its completeness since the knowledge structure must enable the representation of all relationships between risk factors and web project characteristics that can simultaneously affect other factors. The second is it links to the general applicability of our proposed model to any Web projects. The final reason is associated to its inherent flexibility as it can be applied to different situations according to the structures of different Web project domains and different types of data.

To create a BN for RIAP a two-phase process has been identified. Firstly, build the BN graph structure that represent domain variables, and the qualitative relations among them. Secondly, quantifying the obtained structure, define the node probability tables for each node of the graph over and above measures for the quantifiable variables linked to the each node of the graph.

3.2.1.1 Dimensions for Web Project Characteristics

The first phase is the construction of graphical structure of BN for RIAP model by arranging the characteristics into three dimensions: "product", "usage", and "development". These dimensions represent the BN network segments. The utilization of these three dimensions is based on the ISO/IEC 9126-1 standard for the evaluation of software quality characteristics [17]. Fig. 2 shows the dimensions for the categorization of Web project characteristics.

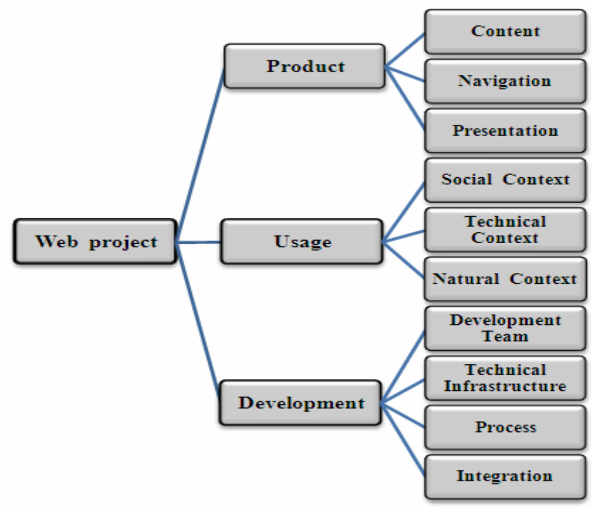

Fig. 2: Dimensions for categorization of Web project characteristics

This phase also obtains the graphical structure for BN that represents the relationship of direct influence among characteristics of web application in each dimension and its risk factors. Our aim was to establish which risk factors in one sub-network had a direct influence on other risk factors in the same sub-network, and eventually on risk factors in other sub-networks. Each relationship is supported by principles that represent the direct influence or dependence between variables and its parent's risk factors. Fig. 3 depicts the graphical structure of the three-level BN.

3.2.1.2 Definition of Node Probability

Quantifying BN means assessing the probability distributions that is node probability tables (NPTs) for each of the nodes represented in BN. A parent node's NPT represents the relation probability of each state (value); a child node's NPT represents the relation probability of each value conditional on every combination of values of its parents. The knowledge for assessing the probability distributions could be obtained either from empirical data, if it exists or elicitation of probabilities from experts domain. Both kinds of knowledge could be used independently or in arrangement with each other [16].

We used the questionnaire responses [8] as objective data for assessing the probability distributions for the nodes in Bayesian networks. As soon as the nodes in BN specified, we used the Bayes' theorem [16] to calculate relationship between nodes. The Bayes' theorem for two nodes is:

$$P(E \mid C) = (P(C \mid E)\, P(E)) / (P(C))$$

where:

- $P(E \mid C)$ is called the posterior distribution and represents the probability of E given evidence C;
- $P(E)$ is called the prior distribution and represents the probability of E prior to evidence C is given;
- $P(C \mid E)$ is called the likelihood function and represents the probability of C assuming E is true.

Fig. 4 shows the specification of the node probability tables for the product dimension.

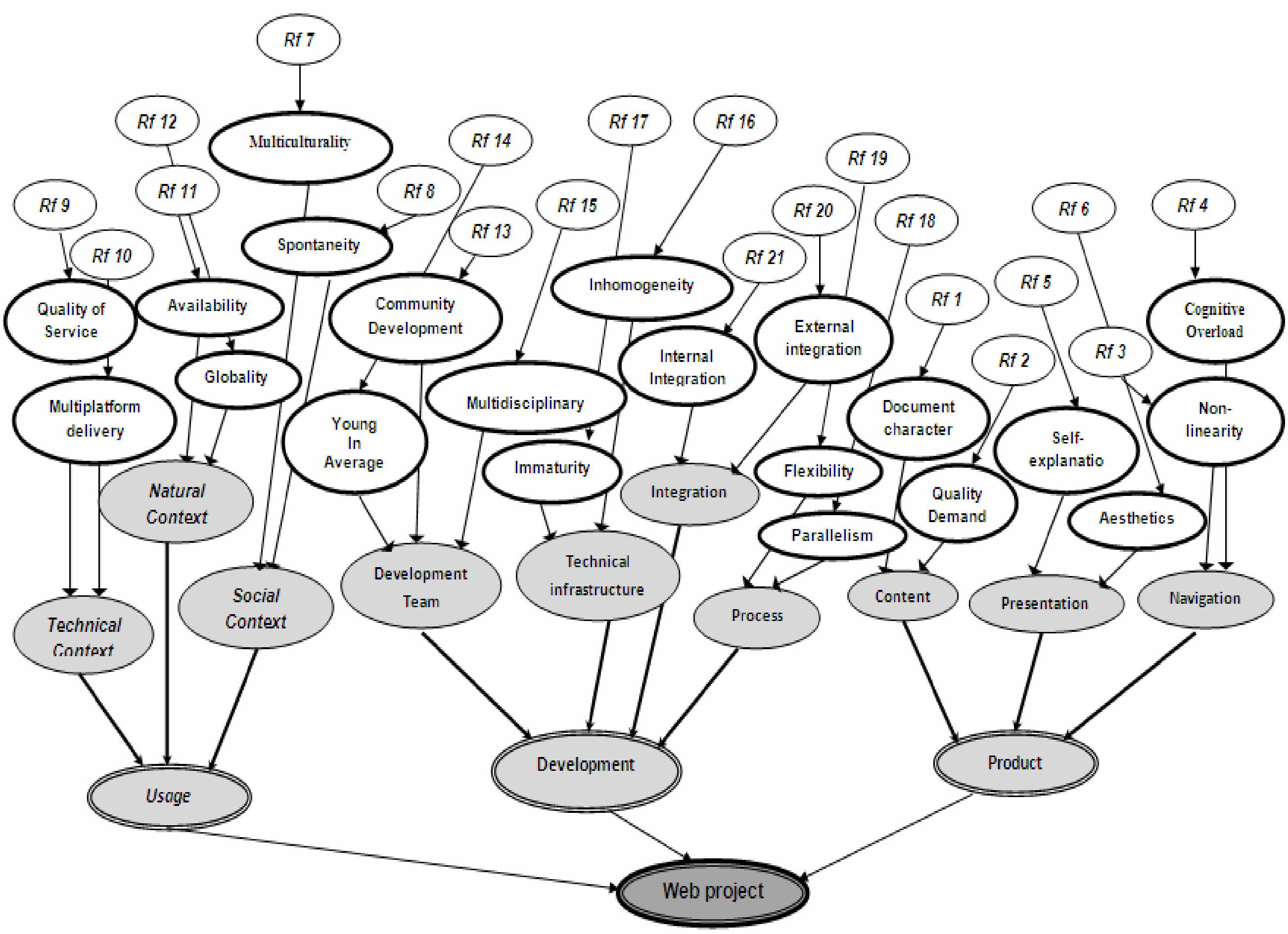

Fig. 3: The Bayesian Network graphical structure that represents RIAP

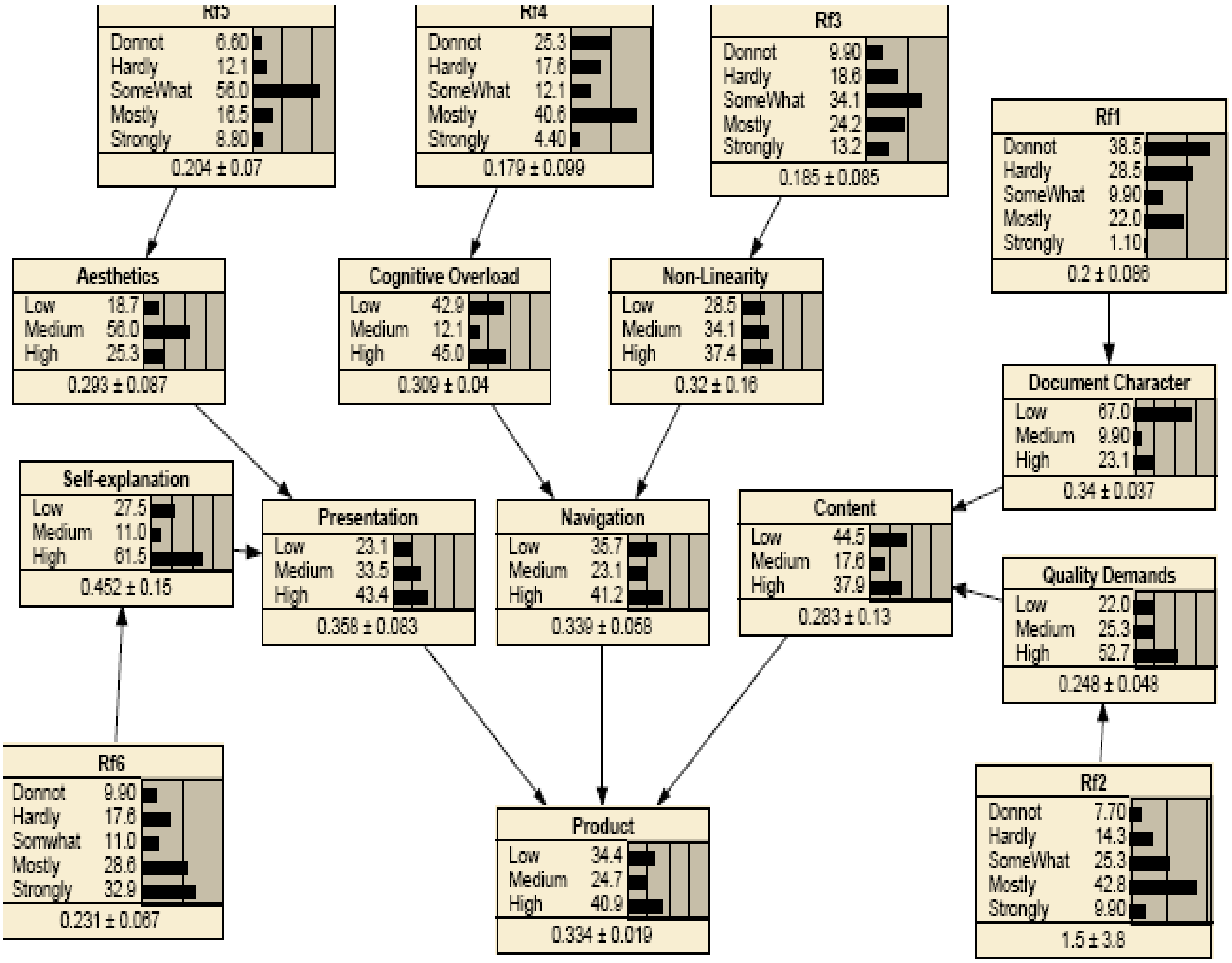

Fig. 4: Node probability tables for the product dimension

4 Risk Pattern Architecture

Preferably, the identification of risks associated with a web project and the process of constructing BN should give us a risk statement that we can apply in the next stage of constructing RIAP. A risk statement is a real description of the not needed event or set of situation and is generated based on the relationship of direct influences between the sub-networks and the identified risk factors.

Upon an evaluation of all risk statements, we can build RIAP by assigning identified risk factors to architectural sub-networks based on their relationship, responsibilities and interplay. Each relation between every risk factor and the causal sub-networks chain represent one risk pattern. In general, risk patterns are derived from the experience of identifying same or approximate risks in a continuously exponential manner. Such lessons learned from each iteration of software architecture process, the risk patterns should be refined and improved or freshly redefined to ensure that the most appropriate course of action can be instituted to confront future risk-endowed challenges.

Each pattern for each type of Web application domain contains information about risk factors and it causal chain. As well as, the best risk mitigation actions, the nodes probabilities tables for the risk factors and the related sub-network nodes, it relations with specific architectural patterns, it is impact on the qualities attributes, and last report that contains information about the pattern improvement or redefined and the discuss of it is benefits and limitations. In addition each pattern have a unique identify name.

Risk identification architecture pattern is presented by using set of risk scenarios, where a risk scenario is a causal chain of risk factors that ends with the occurrence of a risk loss. The causal chain includes identifying the important resource that contain effected risk factors, the Web project characteristic that have that resource and the dimension including that characteristic.

Risk scenario for RIAP can be built using if…then… constructs and logic operators and, or, not. For example:

- *If* Project has too large volume of information contains <**Risk factor**> *then* it will be, cognitive overload <**important resource**> *and* it will be loss of user navigation capability <**characteristic**> *and* it will affect product success <**dimension**>.

It should be noted that each scenario could be redefined and used as a pattern to extract scenarios that are more specific. In addition, in our model, each risk pattern is clarified with text in a natural language. For example:

- {the magnitude of information that contains a high concentration should take into account the many paths or tasks simultaneously. Meaningful linking and smart link label moderate cognitive overload. Furthermore, design patterns in modeling the hypertext aspect could also help address this problem}.

Risk identification architecture pattern focuses on primary project components for instance, the diverse Web application business domain, the patterns and architectural styles used in the system, OTS components and additional communications selections, besides other features depict in traditional architecture design. Accordingly, risk identification architecture pattern can have several diverse levels of abstraction. Moreover, risk identification architecture patterns engage a broad range of concerns, from absolute technical ones to business, political, social and organizational ones.

5 The Bridge between RIAP and Design Decision

Software architecture can be seen as a decision making process; to take the right decisions right time. Usually, these design decisions are not clearly described in the artifacts representing the design [18]. Project management is frequently proposed as a solution, but these solution lacks a close relationship with software architecture artifacts [21]. Risk identification architecture pattern in the software architecture bridges this gap, as it allows for a close integration of management with software architecture Consequently, the software architecture becomes easier to communicate, maintain and evolve. Furthermore, it allows for analysis, improvement, and reuse of design decisions in the design process.

5.1 Software Architecture Process

Software architecture is built based on the requirements of the system. Requirements narrate what the system is supposed to do, while the software architecture defines how to accomplish this. There are many available software architecture design methods, and all of them use diverse methodologies to design the software architectures [19]. Nevertheless, they can all be review with the same abstract software architecture design processes. Fig. 5 presents a view of this abstract software design process and its related artifacts.

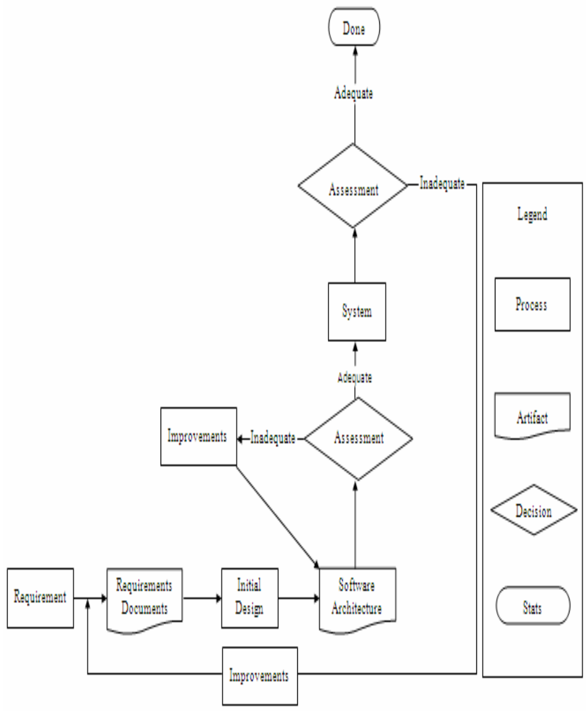

Fig. 5: Software architecture design process

The requirements document is the main input in of the software architectures design process. During the initial design stage, the software architecture is produce, which fulfills fractions of the requirements declared in the requirement document. After this initial design stage, the quality of the software architecture is assesses in order to analyze the architecture, identify the potential risks and verify that the quality requirements were addresses in the design. The architecture is modifying if its quality is not satisfying. In order to modify the architecture, the design can be improved by employing several tactics [19] or adopt one or more architectural styles or patterns[18] and this process in repeated until an acceptable quality is achieved.

5.2 Risk Identification Architecture Pattern (RIAP) Process

A general process for creating (RIAP) is visualized in Fig. 6. First, the problems are identified (risk identification) and produced in a risk identification architecture pattern. Then, the risk identification architecture pattern are evaluated (pattern remaining) one by one, and patterns are generated (generate patterns) for a risks. These patterns are evaluated and weighed for their appropriateness to resolve the problem at hand (risk assessment). The finest solution (for that state) selected, and the selection solution is documented together with its risk identification architecture pattern. If new risks surface from the risk mitigation decision made, they must be recorded and resolved within the same process.

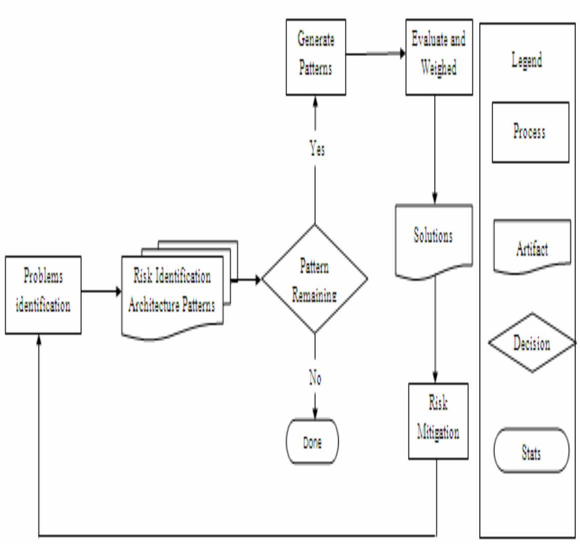

Fig. 6: RIAP process

5.3 Enriching Software Architecture with RIAP

The processes explained in Sections 5.1 and 5.2 have some obvious similarities. Problems (requirements) are managed by solutions (modifications and software architectures), and the evaluation decides if all the problems are resolved satisfactorily.

The objects produced in both processes attend to depict the same thing. Nonetheless, the software architecture design process focuses on the results of the decision process, while the risk identification architecture pattern mainly concentrates on the path to the decision.

Some facts, which are explained in RIAP process are missing in the software architecture design process. There are two artifacts, which contains facts that do not exist in the software architecture artifact: not selected solutions and the solution impact. On the other hand, the outcomes of the design process (software architectures and modifications) overlook the risk identification architecture patterns process.

The concept of our research represented is to compile the two processes together. A software architecture design process is no longer results in a inactive design description of a system, but in a set of risk solutions that leads to a clear design process of the system. The compilation of the software architectures process with RIAP process is shown in Fig. 7.

During software architecture assessment, the architecture is modifying if its quality is not satisfying, and the risk identification architecture patterns refined and improved or freshly redefined to ensure that the most appropriate course of action can be institute to confront future risk-endowed challenges. Risk identification architecture patterns must consequently be an evolving and learning process, adjusting to latest and changing experience as the project proceeds.

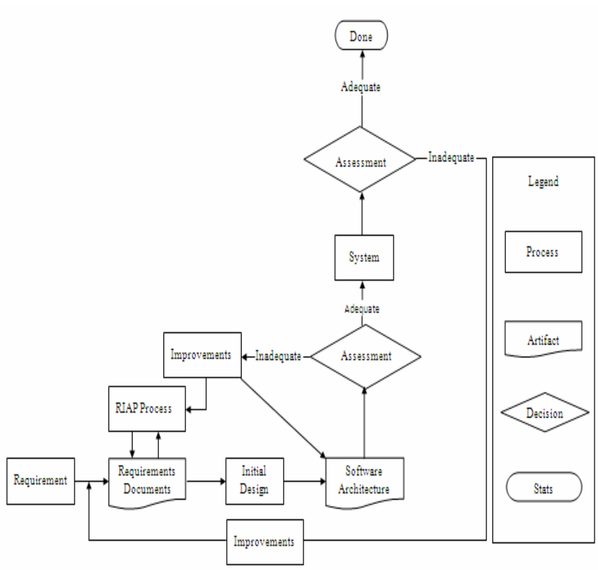

Fig. 7: Enriching software architecture with RIAP

5.4 Designing with RIAP

In order to present design approaches; Bosch [18] illustrates ways in which substitutes are drawn out and trade-offs are completed. An architect designing with risk identification architecture pattern still uses these design methods. The major differentiation lies in the consciousness of the architect to clearly describing the design decision made by identifying how the generic solution scheme addresses a recurring risk that occurs in a specific context.

Designing with risk identification architecture pattern helps in handling the problems on the design decision in the following ways:

- The use of risk architectural pattern resides in its capability as a good general-purpose model with which risks can be detect prior to the design of a web application. In other words, by taking into account the possible impact of each risk item, we can concentrate on controlling the most critical risks first. Moreover, it assists the architect in avoiding obvious decisions faults.
- When designing with risk identification architecture pattern the architect or software engineers they can detect the risks , reasoning, relationships between them, probability of their occurs, and appropriate risk management action can be preformed. Consequently, this can help architects to increase conscious on design decisions to process, and the relations among them. In the short-range, if the identified decisions are not wanted, the concerned design decisions can be reassess mating and substitute solutions can be taking into account before the design is supplementary developed. In the long-standing, the architect can (re)learn which most appropriate courses of design decisions are closely connect with each other, and what kind of risks are related with this.
- The use of Web applications characteristics as main categorize to list the selected typical risk factors, it help the architect or software engineers to explicitly defines the risks and there mitigation a in each characteristic based on it is concerned to project. The concept of separation of concerns would be of greater value in order to gain a better insight into complex systems. This is attaining through narrowing down the scope of system interpretation based on the existence of different entities within the overall framework. Furthermore, the separation of concerns assists in the detailed and comprehensive specification of each concern including its respective dependencies.

- The RIAP represents the reasons behind an architectural design decision.
- Risk identification architecture patterns are resulting from the knowledge of identifying the same risks or similar risks repeatedly. During software architecture evaluations, patterns refined and improved or freshly redefined to ensure that the most appropriate course of action can be institute to confront future risk-endowed challenges. This can enhance the traceability of the design-making decisions and their relationship to features design aspects, interests, and between themselves.

Hence, we can see that the risk identification architecture pattern before initial design is, a good general-purpose model to clarify design process and to take active design decisions. Consequently, the software architecture becomes easier to communicate, maintain and evolve. Furthermore, it allows for analysis, improvement, and reuse of design decisions in the design process.

.

6 Evaluation of RIAP

We conducted a case study that attempted to identify risks in an actual web project using RIAP model. Seventeen final year students from the School of Computer Sciences, Universiti Sains Malaysia, were the subjects of the evaluation. The concerned project was an e-learning portal for the school.

Each student had to identify risk factors associated with the project development. An analysis of the data revealed that the students identified a diverse set of risks such as vague project requirements or the lack of a detailed plan for the project. The latter type of risks included team members lacking project specific skills, the lack of a standard and explicit definition of project quality as well as the absence of top management support for the project. Then we requested project managers to evaluate the students responses based on their perceptions and personal experiences.

The case study involved the steps below:

i. Analyze all identified risk factors and categorize them under web project dimensions: “product”, “usage”, and “development”.
ii. Rank factors in (i) based on the characteristics of the web projects.
iii. Feed (ii) into BN to study sources of risk factors and represent all existing relationships between the risk factors and the web project’s characteristics.

iv. Represent risk factors using RIAP to verify whether the risk identification architecture patterns could rephrase the identified risk factors.
v. Organize RIAP to locate unidentified risks not detected by the subjects. This step could determine whether RIAP could add meaningful risks to those risk factors already identified by the subjects.

From the case study, we were able to interpret all the risk factors identified in step (ii) according to the risk patterns. Then we were able to identify a number of risks that were not identified by the subjects based on project managers' or experts' evaluation. Lastly, the identification of risk factors using RIAP can widen the scope of risk identification using a comprehensive and extremely detailed risk identification architectural pattern. Hence it can support the architectural design decision of the concerned project.

7 Conclusion

Architectural design decisions play an important role in the design, development, integration, evolution, and reuse of software architectures. However, the notion of architectural design decisions is not part of the current perspective on software architectures.

We propose a new perspective on software architecture, where software architectures are described as set of design solutions. The presented RIAP model is centered on this idea. RIAP model described the relationship between risk management and design decisions in detail. It uses the risk causal chain, risk impact, risk probabilities, risk solutions, and separated concerns to clarify design process and to take active design decisions.

References:

[1] G. Kappel, B. Proll, B. Retschitzegger, W. Schwinger, Th. Hofer, *Towards a Generic Customization Model for Ubiquitous Web Applications*, Proc. of the 2nd International Workshop on Web Oriented Software Technology (IWWOST), Spain, July, 2002.

[2] S. Murugesan, Y. Deshpande, *Meeting the challenges of Web application development: The Web engineering approach*. Proceedings of the 24th International Conference on Software Engineering, 2002, pp. 687-688.

[3] PMBOK Guide, 2000 Edition, *Project Management Institute*, 2000.

[4] G. Starke, *Effective Software Architecture – Ein praktischer Leitfaden, Carl Hanser*, Verlag, 2002, in German.

[5] L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice*, Addison-Wesley, 1998.

[6] A. del Caño, M. P. de la Cruz, Integrated methodology for project risk management, *Journal of Construction Engineering and Management*, ASCE Vol. 128, No. 6, 2002, pp. 473–485.

[7] H. Mayr, *Project Engineering – Ingenieurm¨aßige Softwareentwicklung in Projektgruppen*, 2nd revised Edition, Fachbuchverlag Leipzig/Carl Hanser Verlag, 2005, in German.

[8] T. Rousan, S. Sulaiman, R. Abdul Salam, A Risk Identification Architecture Pattern Based on Bayesian Network, International Symposium on Information Technology, IEEE, Vol. 4, Aug. 2008, pp. 2224-2234.

[9] J. S. Carroll, E. J. Johnson, *Decision Research: A field guide*, Sage Publications, 1990.

[10] K. R. MacCrimmon, D. Wehrung, *Taking Risks: The Management of Uncertainty*, the Free Press, New York, 1986.

[11] F. Buschmann, J. Meunier, H. Rohnert, P. Sommerlad, M. Stal, *Pattern Oriented Software Architecture*, John Wiley & Sons, 1996.

[12] R. S. Pressman, Applying Web Engineering. In: *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill, 2005.

[13] R. Balasubramaniam, J. Pries-Heje, R. Baskerville, *Internet Software Engineering: A Different Class of Processes*, Annals of Software Engineering, Vol. 14, No.1–4, December 2002, pp. 169–195.

[14] Lowe, D., *Characterisation of Web Projects*, Proc. of the 8th Australian World Wide Web Conference (Aus Web 2002), Queensland, July, 2002.

[15] E. J. Whitehead, A Proposed Curriculum for a Masters in Web Engineering, *Journal of Web Engineering*, Vol. 1, No. 1, Rinton Press, October, 2002, pp. 18–22.

[16] M. Neil, N. E. Fenton, L. Nielsen, *Building large-scale Bayesian Networks, The Knowledge Engineering Review*, Vol. 15, No. 3, (2000), pp. 257-284.

[17] ISO/IEC 9126-1.Standard for the evaluation of software quality characteristics.2001. (http://www.iso.org/).

[18] J. Bosch., *Software architecture: The next step.* In Software Architecture, First European Workshop (EWSA), Vol. 3047 of LNCS, Springer, May 2004, pp. 194–199.

[19] L. Bass, P. Clements, R. Kazman, *Software architecture in practice*, 2nd edition, Addison and Wesley, 2003.