1-1-2014

Enhancing Requirements-Level Defect Detection and Prevention with Visual Analytics

Shirin Rad

Enhancing requirements-level defect detection and prevention with visual analytics

By

Shirin Rad

A Master Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Science
in the Department of Computer Science and Engineering)

Mississippi State, Mississippi

May 2014

Enhancing requirements-level defect detection and prevention with visual analytics

By

Shirin Rad

Approved:

____

Nan Niu
(Major Professor)

____

David Dampier
(Committee Member)

____

Edward B. Allen
(Committee Member and Graduate Coordinator)

____

Jason M.Keith
Interim Dean
Bagley College of Engineering

Name: Shirin Rad

Date of Degree: May 17, 2014

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Nan Niu

Title of Study: Enhancing requirements-level defect detection and prevention with visual analytics

Pages in Study: 49

Candidate for Degree of Master of Science

Keeping track of requirements from eliciting data to making decision needs an effective path from data to decision [43]. Visualization science helps to create this path by extracting insights from flood of data. Model helps to shape the transformation of data to visualization. Defect Detection and Prevention model was created to assess quality assurance activities. We selected DDP and started enhancing user interactivity with requirements visualization over basic DDP with implementing a visual requirements analytics framework. By applying GQM table to our framework, we added six visualization features to the existing visual requirements visualization approaches. We applied this framework to technical and non-technical stakeholder scenarios to gain the operational insights of requirements-driven risk mitigation in practice. The combination of the first and second scenarios' result presented the multiple stakeholders scenario result which was a small number of strategies from kept tradespase with common mitigations that must deploy to the system.

ACKNOWLEDGEMENTS

This thesis is dedicated to my parents and my brother. This thesis is dedicated to all those who believe in the richness of gaining knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In a software project, risks represent the situations that threaten the satisfaction of stakeholder goals. For iterative and incremental development models like Spiral [4], analyzing risks is the fundamental mechanism for evaluating design alternatives and driving project advancements. This makes understanding risks at the requirements level especially important because the cost of correcting a requirements-level error during implementation or system integration can be 10-200 times more than that during requirements engineering (RE) [36, 40].

Many approaches have been proposed to address risks in RE, such as quality function deployment [1], fault mapping [45], and goal modeling [12]. One model, called defect detection and prevention (DDP), was originated from the Jet Propulsion Laboratory and used in safety- and mission-critical contexts [5]. Although this model has been adapted in different applications (e.g., expertise matching [14] and technology scouting [15]), its primary focus is on quantitative risk-based requirements reasoning [12]. We review the basic concepts of DDP in Chapter 2.

A recently emerging feature of DDP is the use of visualization to facilitate the risk assessment process [16]. In particular, a cost-benefit plot positions all possible risk-mitigation strategies over a two-dimensional space, allowing an optimal solution or a set of solutions to be readily identified. While such a plot is valuable for gaining a static overview of the mitigation solution space, further enhancements are possible.

In this thesis, we propose to integrate visual analytics (VA) into the DDP model. VA is defined as "the science of analytical reasoning facilitated by interactive visual interfaces" [53]. The goal of our work is to create the analytical capabilities for requirements-level risk assessment and blend these capabilities into the underlying model and existing visualization of DDP. The resulting enhancement thus goes beyond a static visual depiction of the data by moving towards a truly interactive visual interface that allows the user of our approach to directly manipulate the data so as to gain insights in dynamic ways.

The contributions of our work are twofold: (1) development of analytical visual supports that take stakeholder preferences and multi-stakeholder tradeoffs into account in the DDP process; and (2) evaluation of our approach in the context of an industrial case study. Overall the findings from our initial evaluation suggest that the increased interactivity of requirements visualization leads to more accurate, informed, and defensible decisions.

The rest of the proposal is structured as follows. Chapter 2 lays the background of our research by introducing the DDP model and the requirements-driven risk assessment process. Chapter 3 presents our VA enhancements and prototype tool. Chapter 4 describes our empirical evaluation in the context of an industrial case study. Chapter 5 discusses related work, and finally, Chapter 6 concludes the proposal.

CHAPTER II

BACKGROUND AND RELATED WORK

After several years of research, Feather et al. developed "Defect Detection and Prevention" (DDP) model with the focus on risk-based quantitative reasoning to assess the viability of novel systems development [6]. They believed "DDP model tries to fill the niche between qualitative approaches and detailed design centric analysis approaches" [12].

DDP model was planned to assess quality assurance activities by Cornford at JPL; therefore many approaches have been proposed to address how to select assurance activities such as defensive measurement, analysis, inspection, and test in order to better manage time, budget, and trained technical individuals as resources.

These are the three levels of classification as the key concepts of DDP model [12, 17, 46] as shown in Figure 2.1:

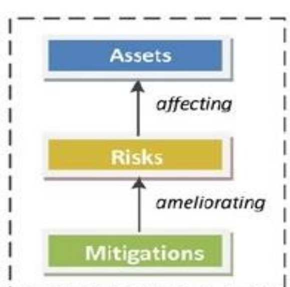

Figure 2.1 Overview of requirements-driven model [46]

- Assets represent what requirements system or technology needs to have. This level represents all requirements-level information which stakeholders need to know. Feather et al. used VA, ‘‘the science of analytical reasoning facilitated by interactive visual interfaces’’ [53], to create a tree chart from assets as a group of artifacts. These assets were checked in two dimensions: how much their implementation would cost, and how much they are valuable for the system.
- Risks reflect what could affect the attainment of objectives or assets. The typical risk management asks users to anticipate risk after implementing all risk assessment activities on the system. DDP model calculates the risk severity as the summation of all the risk "impacts" which are moderated by mitigations. In other words, the degree of risk management can show how much it would satisfy a requirement.
- Mitigations propose those actions that should be taken to diminish the probability of risk occurrence. Assurance activities, testing, process control, and analysis, as mitigations can control the amount of failure and risk occurrence through requirements elicitation. Selecting right mitigations with high benefit and low cost can reduce the risk.

As a concrete example for DDP model, we show three tables of requirements, risks, and mitigations. Table 2.2 contains four requirements with different value to the system. Deploying each of these requirements may be affected by a failure or risk. Table 2.2 highlights some examples of the risks that may hinder these requirements, and Figure 2.2 shows the relation of requirements and risks through the "attack" relationship. Also, each number on the line shows the weight. Based on our description about requirements in DDP model, each of the risk-to-requirement links has their weights, and the higher weight means the higher percentage of risk occurrence in deploying that requirement in software system. For instance, the impact of selecting R1 for the software system would be the summation of all three connected risks to R1, 0.8 + 0.9 + 0.4 = 2.1 as the "impact" measure of R1.

Table 2.2 Requirements Table

Requirement Number	Description	Benefit
R1	Improve healthcare system	81
R2	Develop children from different perspective (cognitive, social. emotional, creative)	75
R3	Be aware of prematurity and genetic conditions	90
R4	Check the children concentration and mental health	70

Table 2.3 Risk Table

Risk Number	Description
r1	grow in a environment without attention to the early childhood development needs
r2	Increase the number of children with serious disease
r3	Decreased the level of self-confidence, communication and level of motivation to learn more

Table 2.4 Mitigation Table

Mitigation Number	Description	Cost
M1	Set some medical tests for children specially for under school age	87
M2	Classify teachers in some level of proficiency for different educational level	75
M3	Teach children how to have social engagement and encourage them for further learning	58

Table 2.3 states three mitigations that can protect system against risk happening. Each of the mitigations can affect some risks with different effect value. In accordance with Figure 2.2, r1 is affected by M1 and M3, and its effect value can be measured (1-(1-0.8)*(1-0.8)) = 0.96 [12]. In other words, injecting M1 and M3 to the system can reduce the risk to just 4 percent (1-0.96 = 0.04). The attainment of each requirement is the summation of attainments of risks to that requirement. We can calculate this attainment for each connected risk to R1. So, selection of mitigations can affect risks and indirectly requirements satisfaction. Also, the R1 "at risk" measure is computed by summing the impacts of effects on that.

Feather et al. mentioned the n mitigations can be selected in 2^n ways, and 2^n counted as all the possible ways in the solution space [14]. Based on DDP model, these three mitigations can be selected in 2^3 ways, and 8 strategies can be created by deploying three mitigations [14]. Table 2.4 illustrates these strategies and all the possible strategies. These strategies can be generated in different area with various cost and benefit. Feather et al. believed lower cost and higher benefit are two significant dimensions in evaluating the worth of deploying strategies in a system [14].

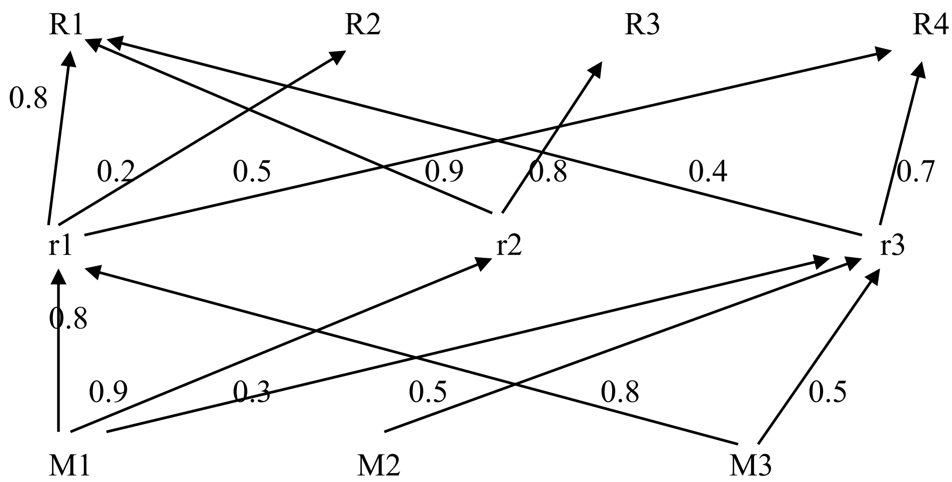

Figure 2.2 Connection of DDP model levels

Benefit of selecting M3 ⇨ r3: 1 – [(1 – 0.7) * (1- 0.4)] = 0.82 (0.18 solve)

⇨ 58 * (1 – (0.18 *0.5)) = 52.78

⇨ r1: 1 – [(1 – 0.8) * (1- 0.2) * (1- 0.5)] = 0.92(0.08 solve)

⇨ 58 * (1 – (0.08 *0.8)) = 54.288

⇨ 52.78 + 54.288 = 107.068

Table 2.5 Strategy Table (1: selected, 0: not selected)

Strategy	M1	M2	M3	Cost	Benefit
S0	0	0	0	0	0
S1	0	0	1	58	107.068
S2	0	1	0	75	68.25
S3	0	1	1	133	175.318
S4	1	0	0	87	249.168
S5	1	0	1	145	356.236
S6	1	1	0	162	317.418
S7	1	1	1	220	424.486

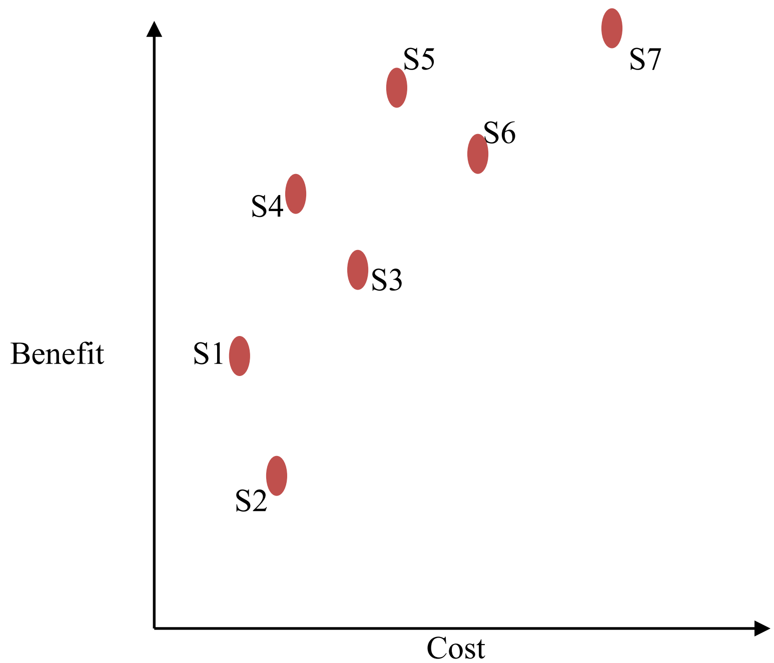

Figure 2.3 Strategy plot

Feather et al. selected 53 mitigations and visualized all the possible probabilities (2^{53} ways) that can come out from their relations. Figure 2.4 shows all these possible connections for n mitigations with black dots in space. The location of these dots (strategies) shows the level of cost and benefit can be added to the software system. They believed sweet spot location is where stakeholders can have more than average benefit and low cost [18].

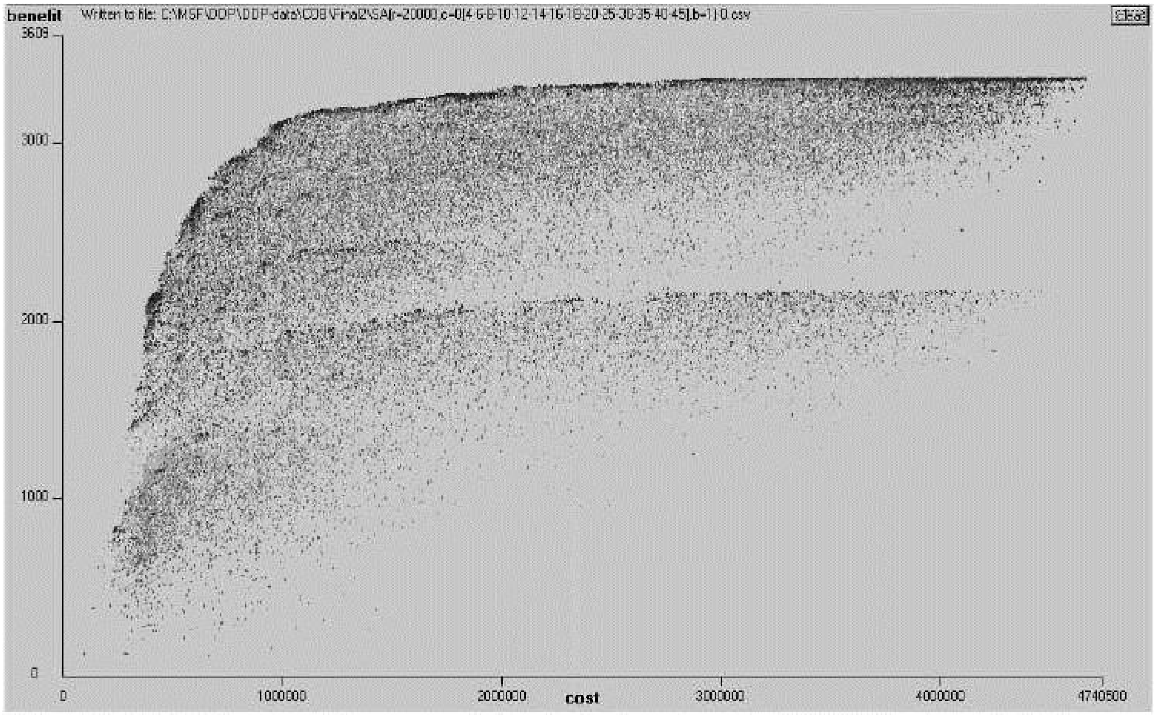

Figure 2.4 Automated search for the cost-benefit tradespace [18]

Although checking all the possible strategies in space can make a good overall idea for users, it is not always necessary because stakeholders just need to select their preferences. DDP behaves as a static model because it does not take user's preference into consideration. Being static not only does not satisfy users in term of time efficiency, but also it needs high power hardware resources for deployment, which is not cost-efficient for different stakeholders.

In case of user interaction for technical and nontechnical users, Visual Analytics as a good solution increases the ability of analyzing and using big data. VA as ‘‘the science of analytical reasoning facilitated by interactive visual interfaces’’ [53] emerges to address different challenges that users may face. Simply, it visualizes data so humans can directly play with data to figure out the insights, and ultimately use their information to make an optimal decision. VA combined different research areas like data mining and statistics. One of the recent approaches is applying VA in requirements engineering. It highlights the relationships and constructs of some information to help human analysts select the right requirements and work with those requirements. VA includes human in data analysis step to leverage creativity, knowledge background and flexibility. Furthermore, integrating VA into DDP anticipated making more interaction between users and system in order to help users make an accurate and informed decision.

Reddivari et al. have implemented visual analytics framework in requirement engineering. In other words, they create direct interaction with software system for users. They exactly model the "user" to highlight that machine cannot replace with human just because of its augmenting in computations. They add the capability of distinguishing the level of user involvement throughout the requirements elicitation to their framework. They used DDP to gauge their input data in terms of risk assessment in two types of (1) decision in terms of risk assessment, one for "in-scope?" for those subset of assets, risks, and mitigations that are in special assessment cycle, and (2) for in-scope selected subset of mitigations that can address cost-value model[18].

CHAPTER III

VISUAL ANALYTICS ENHANCEMENTS AND PROTOTYPE TOOL

In this chapter, we take a detailed look at visual analytics (VA) as it is applied to requirement engineering. We use the resulting framework to guide the enhancements over the basic DDP model. It is important to explain the term ‘‘visual requirements analytics’’ that we use to refer to the subject matter of our research. This term is derived from a recent article published in IEEE Software [39] where Menzies and Zimmermann presented the guest editors’ introduction to ‘‘software analytics.’’ They defined the emerging field as ‘‘analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions’’ [39]. We therefore believe that if the source of ‘‘software data’’ is requirements- centric as opposed to implementation-centric (e.g., [10]), then ‘‘requirements analytics’’ can characterize the use of analysis, data, and systematic reasoning for making decisions that will benefit managers, requirements engineers, and other relevant stakeholders. Furthermore, if visualization is the primary means by which insights are drawn and shared, then the term ‘‘visual requirements analytics’’ can be used to describe the data-to-decision process. For this reason, we use ‘‘visual requirements analytics’’ and ‘‘VA for RE’’ interchangeably for the rest of the thesis.

3.1 A framework for visual requirements analytics

In essence, VA is aimed at synthesizing the strengths of machines with those of humans [53]. On one hand, modern computers and automated methods, such as data mining [25] and machine learning [2], offer unprecedented computational power to facilitate knowledge discovery. On the other hand, it is indispensable for informed decision making to include humans in the data analysis process to leverage flexibility, creativity, and background knowledge [30]. The specific advantage of making the human–machine synthesis in a visual way is that data analysts, decision makers, and other stakeholders can focus their full cognitive and perceptual attentions on the visualization-enabled analytical reasoning while taking advantage of the automatic data processing techniques [30]. We have developed a visual requirements analytics framework based on the VA literature. Figure 3.1 shows the framework which consists of five components (user, data, model, visualization, and knowledge) and their interactions. Compared with existing conceptualizations (e.g., the ones presented in [53] and [30]), our framework is novel in a couple of aspects. First, it explicitly models the ''user'' to suggest that machine's computations only augment, but cannot replace, human's capabilities to perceive, relate, and conclude in the knowledge discovery and decision making process. Second, our framework distinguishes the degree of user involvement in the VA activities: primary to the user, secondary to the user, or subject to full automation.

These distinctions are made by using different transition types in Figure 3.1. In what follows, we detail the introduction of the proposed framework by discussing the components, the connections between the components, and the different levels of user involvement.

3.1.1 Components

User We choose the term ''user'' to label the human role in Figure 3.1. The rationale is to denote the role as somebody who uses the VA methods, techniques, and tools to carry out RE tasks. In practice, the VA ''user'' can be a requirements engineer, a data analyst, a business manager, a project coordinator, a developer, a tester, a customer, and/or an end user of the software system. In many situations, the ''user'' is not just an individual but a group of stakeholders. For example, using VA's fact-based decision support to answer questions like, ''How much resource is needed for this new feature request and who is most capable of implementing it?'' can help project managers reason more strategically about the importance of the changing requirements, facilitate customer service representatives to better locate technical expertise when answering user queries, and guide sales staff in pricing features by understanding the inherent values and trade-offs. In this sense, analytics is truly about what software projects can learn from themselves and each other, or put it in another way, ''analytics means sharing information'' [39].

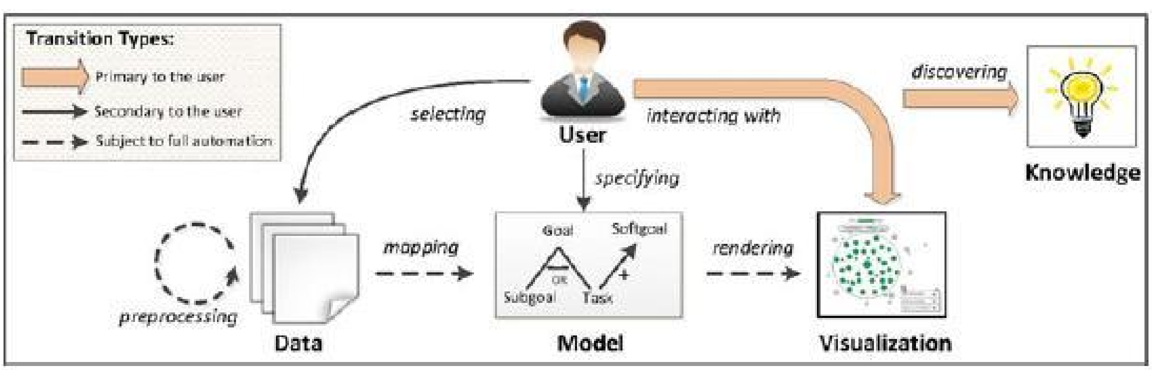

Figure 3.1 A framework that characterizes the key components and their interactions in the visual requirements analytics process[46]

Data Due to the Internet and open source, there is now so much data about software projects that it is impossible to manually browse through it all [39]. Take the SWP project as an example, the focus group meetings helped elicit 113 requirements for only one agency (K-12), but there are five major agencies that the project is aimed to serve. The other four are early childhood, community college, university, and workforce. As mentioned earlier, the requirements for many software projects are of large scale, of different source, of distinct format, and even of various qualities. Therefore, the first step of VA is often to process the raw data in order to extract relevant requirements information for further visual and automatic analyses. The data can be selected manually by the user with the help from automated preprocessing tools and techniques.

Model Continuing with the preprocessed data, the underlying model in Figure 3.1 defines what entities and relationships will be used to support the user's RE task at hand [46]. Goals [29], use cases [40], features [56], problem frames [50], and stakeholder social networks [26] are among the most commonly employed models. Though graphical in some cases, the model is primarily concerned with specifying the problem domain ontology [22], thereby shaping the transformation from data to visualization. In certain approaches (e.g., [46, 10]), the model is only implicit in that the natural language descriptions are extracted and treated as the main requirements constructs. In our current research, DDP serves as the underlying model [46].

Visualization Unlike scientific visualization where the data entities are typically 3D geometries or can be explicitly referenced to time and space [26], the visualization of requirements is a type of information visualization (IV) [8] that deals with abstract data with hundreds of dimensions and no natural mapping to the display. Thus, novel techniques are devised by employing metaphorical [22], quantitative [47], hierarchical [48], relational [37], and other graph-based [27] visual data representations. It is well known in the IV community that, very often, there are many different ways to represent the data under consideration [8]. Searching for the best requirements visualization can be impractical and even counterproductive. It is therefore more valuable to create effective and efficient ways to analyze the data [46]. In this sense, VA is more than just the visualization. It also focuses on how the user interacts with the visualization. Influenced by Shneiderman's celebrated ''overview first, zoom/filter, details on demand'' IV interaction mantra [51], Keim et al. [30] describe the VA interaction mantra to be (1) analyze first, (2) show the important, (3) zoom, filter, and analyze further, and (4) details on demand [46].

Knowledge The interactions with the requirements visualizations shall augment the user's knowledge discovery and lead to actionable decisions; otherwise, they become wasted interactions [46]. However, reaching actionable decisions sometimes also requires new insights and real-time reasoning. It is crucial to note that the knowledge resulting from analytics must be relevant to practitioners (i.e., the ''user'' in Figure 3.1). Only by proving the cost-effectiveness of the VA technique can we address the need for tool support that leverages our knowledge of software engineering to provide more meaningful and less superficial software analytics [39].

In sum, the five components described above form a core set of constructs for the visual requirements analytics framework [46]. Among these components, ''user'' is arguably the most important element as it connects to all other parts and therefore plays an integral role in controlling, monitoring, and adjusting the entire VA process [46]. As pointed out by Menzies and Zimmermann [39], ‘‘users before algorithms’’ is a fundamental principle for software analytics. In our opinion, explicitly embodying ‘‘user’’ in the decision making and knowledge discovery loop is a salient feature that distinguishes VA from IV, and similarly distinguishes VA for RE from REV. For this reason, we will discuss the different levels of user involvement in Sect. 3.1.3, but next, we describe the interconnections of the framework’s nonuser components [46].

3.1.2 Connections

Preprocessing Preprocessing is aimed at cleaning, normalizing, and aggregating data for further processing and modeling [46, 42, 20]. Due to the large volume of data, automated methods are commonly deployed. Goldin and Berry [23] presented a seminal paper in requirements preprocessing where the clerical tool called AbstFinder was introduced to identity important domain concepts from the large mass of natural language text collected from the clients and users. Other preprocessing approaches include our own work on extracting domain-aware lexical affinities [43] as well as our systematic study on indexing where different procedures (e.g., tokenizing, filtering, stop word removal, stemming, etc.) and their interdependencies were organized in a feature model [36].

Mapping While data preprocessing can result in many constructs, showing these constructs in a visual form needs an underlying model that specifies ‘‘what’’ to be visualized and ‘‘how’’ to visualize them [46]. We call this transformation ‘‘mapping’’ in Fig. 1. In i* [58], for example, ‘‘what’’ to be visualized consist of actors, goals, softgoals, tasks, and resources, whereas the strategic dependency and strategic rationale models define ‘‘how’’ to visualize these constructs. Models based on use cases [40], on the other hand, require the mappings of ‘‘actors’’ and ‘‘use cases’’ and define ‘‘uses’’ and ‘‘extends’’ as basic ways to link the constructs. Note that each model focuses on certain constructs and ignores many others. Therefore, the VA approach equipped with an extensible model will allow new constructs (e.g., ‘‘aspects’’ in use case maps [40]) to be integrated in the visualization, thereby facilitating fresh insights to be generated [46].

Rendering The key for visual requirements analytics is to produce interactive visualizations for the users to leverage their cognitive and perceptual skills to perform reasonings, draw insights, and make decisions [46]. The visual aspects of RE models, however, have received surprisingly little attention in the literature [34]. Moody et al. [34, 33] filled the gap by proposing a set of principles for cognitively effective visual notations [34], including semiotic clarity, perceptual discriminability, and graphic economy. While a recent empirical study [7] on i* shows that the visual notations designed by following the principles are more semantically transparent than those originally proposed [58], the work is in line with the REV theme of striving for the best set of static notations. Our focus, in contrast, is on creating dynamic visualizations to support analytics [46].

3.1.3 User involvement

As shown in Figure 3.1, a novelty of our framework is the distinction of different levels of user involvement in the visual requirements analytics process [46]. This section groups the descriptions based on the three categories: primary to the user, secondary to the user, and subject to full automation [46].

Primary to the user We highlight in Figure 3.1 that it is through the interactive visualizations that important insights are gained, efficient reasonings are performed, defensible assessments are made, and optimal analysis results are arrived at [46]. In software analytics, data are abundant, and most managers and engineers are technically and analytically skilled, but these stakeholders typically do not have sufficient time to dig into the details [46]. As a result, they need visual approaches to fully grasp the findings. Graphs and charts produced by statistics and spreadsheet tools are a good start, but more research is needed on how to bring the message out of the software analytics to those who make decision based on them [46]. The visualization is what will make software analytics powerful [39], and our research is precisely focusing on this essential issue.

Secondary to the user Two transitions in Figure 3.1 are secondary to the user: selecting the input data and specifying the model elements [46]. A principal guideline of data selection is to go mining with the data in hand, not the data that one might want or wish to have at a later time [39]. The reason for that is because one may not have control over how data is collected, which makes data cleansing and spurious data removal particularly important preprocessing steps [51]. As for model determining, a trend in software analytics is to shift from searching for global models that can cover many situations to tailoring local models and then sharing the lessons learned [39]. We adopt this view in our work so that different underlying models can be used to tackle different RE tasks in a customized and complementary way [46].

Subject to full automation As mentioned earlier, the use of advanced machine learning and statistical methods in software repository mining has resulted in numerous tools. In fact, the application of automated data mining techniques in software analytics has become a resounding success [39]. The emphasis of all automation in software analytics, however, should be put on supporting the generation of real-time, shared, and actionable decisions [39].

It is worth pointing out that, in our framework, the VA path from data to decision is not strictly linear but highly iterative and incremental with feedback loops between and within the stages [46]. For example, a visual comparison may generate new hypotheses to test, which in turn triggers the user to scrutinize certain preprocessing procedures and to refine the underlying data model [46].

3.2 Using the framework to enhance the basic DDP model

The main objective of the proposed framework is to assess existing VA approaches in RE [46]. This not only substantiates the value of the framework, but also suggests potential tool integration and guides further tool development in a principled manner [46]. The five components presented in Figure 3.1 represent the key areas and thus the conceptual goals that a visual requirements analytics approach shall satisfy. It is this straightforward mapping that motivates the application of the goal question metric (GQM) [3] paradigm in our work. The top row of Table 3.1 lists the conceptual goals. In GQM, a goal needs a purpose, issue, object, and viewpoint [3]. Take the ''user'' goal as an example; here the need is to assess (the purpose) the adequacy (the issue) of user satisfaction (the object) from the VA tool provider's perspective (the viewpoint) [46]. In order to derive the operational questions associated with each goal, we performed an extensive analysis of the literature in the area of requirements engineering visualization with special emphasis on analytical solutions [46]. When reviewing Gandhi and Lee's seminal work [46, 22], for instance, we noted that a real-world security certification and accreditation scenario could involve over 500 requirements. Thus, the question ''Does the VA approach support large-scale inputs?'' (D1 in Table 3.1 [46]) was posed.

Continuing in a like manner yielded all the questions for use in GQM. Table 3.1 groups and labels each goal's operational questions [46].

Table 3.1 Five conceptual goals and their operational questions to be addressed by a visual requirements analytics approach

User		Data		Model		Visualization		Knowledge	
U1	Multiple stakeholder roles	D1	Large-scale inputs	M1	Explicit model representation	V1	Multiple views	K1	Anomaly detection
U2	Usage without heavy training	D2	Heterogeneous input types	M2	Automatic model construction	V2	Inter-view navigation	K2	Detailed explanation
U3	Real-time performance	D3	Automatic preprocessing	M3	Model extension and customization	V3	Browsing	K3	Hypothesis-based reasoning
U4	Integration into existing software development environment			M4	Model traceability	V4	Searching	K4	Scenario-based reasoning
U5	Practitioner-oriented guidelines					V5	Query-drilling	K5	Actionable decision
						V6	Filtering		
						V7	Annotation		

In our work, we focus on importing 6 features over the basic DDP model. Next we discuss these enhancements in detail.

3.2.2 Multiple Stakeholder roles (U1)

In accordance with Table 3.1, one of the VA enhancements over DDP is *multiple stakeholder roles* feature [46]. Figure 3.2 clearly shows that by each requested modification in a project multiple departments are satisfied to decide whether they need to do the modification or not [46]. In order to make modification in a development process, the tool should be sufficient to be used by all the departments with different knowledge [46].

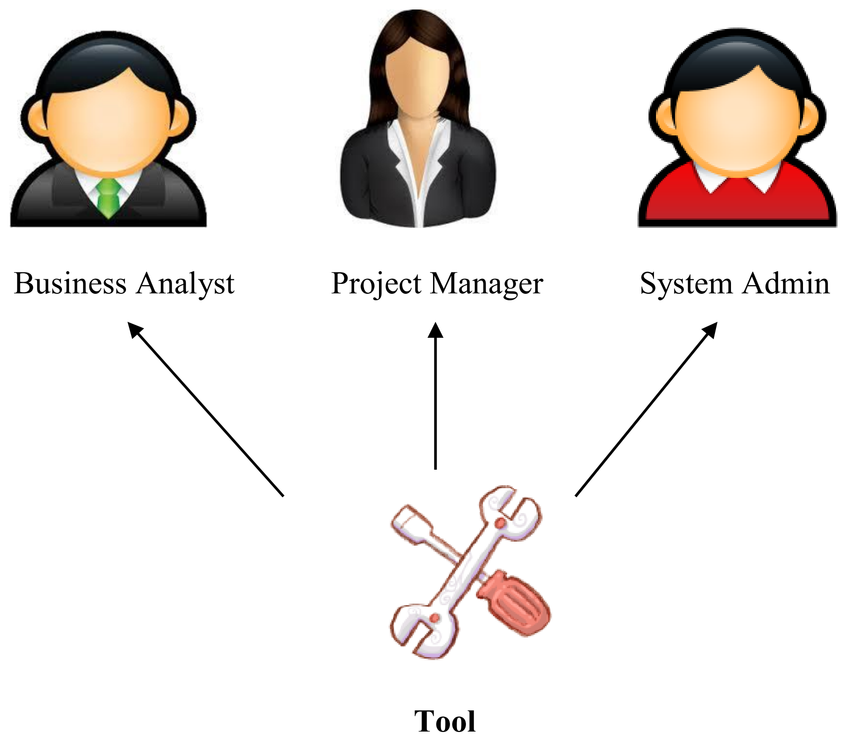

Figure 3.2 Multiple Stakeholder roles

3.2.3 Users without heavy training (U2)

Users without heavy training is known as another feature of our framework for stakeholders without any knowledge about the nature of this visual analytic processes [46]. We can notice from Figure 3.3 that a business analysts do not need to fully undrestand the grasp of internal process which get them into visualization step. By selecting requirements and pressing button they are expecting to see the visualized data and start analysing the prons and cons of their needs [46].

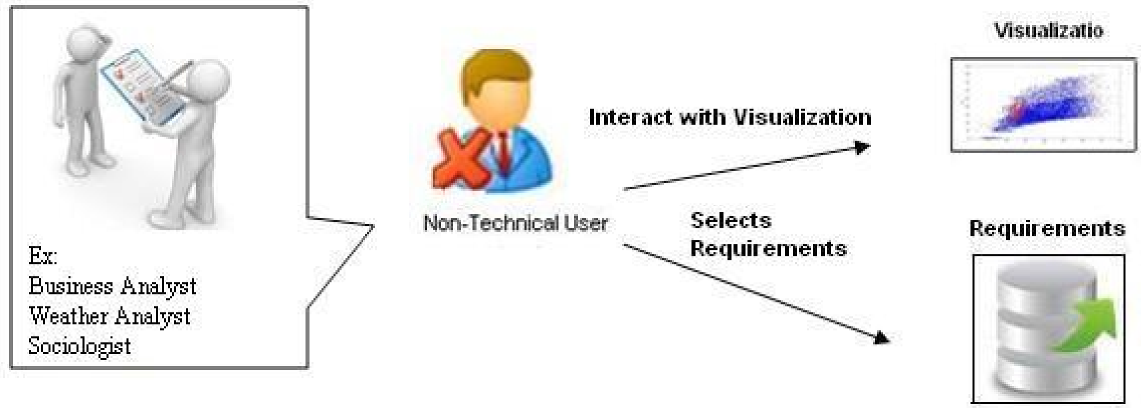

Figure 3.3 Non-Technical Users

The fourth goal, visualization, can be supported by our framework through some features: Filtering (V6), Annotation (V7).

3.2.4 Filtering (V6)

Filtering feature supports selecting preferred requirements. It helps to ignore unwanted requirements and to delete noises to return what exactly users want to see. Our enhancement on top of DDP enables users to select their preferences and the enhanced DDP will just visualize their needs [46].

3.2.5 Annotation (V7)

Annotation means attaching name and label to the data with their current status. Actually, annotation helps to separate different sets of data for further reviews and modifications [46].

The fifth goal, Knowledge, is the most important goal in the VA requirements analytics process: *scenario-based reasoning (K4) and actionable decision (K5)* [46].

3.2.6 Scenario-based reasoning (K4)

Scenario-based reasoning (K4) feature helps to analyze the ''what-if'' scenarios. For example, a manager uses the VA support to compare the cost and benefit of implementing different subsets of requirements as alternatives, and then scenario-based reasoning is performed. With this enhancement over DDP, implementing requirements with evaluated cost and benefit becomes possible [46].

3.2.7 Actionable decision (K5)

Actionable decision The insights, explanations, and reasonings shall all contribute to making decisions that are actionable (K5). For example, based on DDP, cost and benefit are the two key dimensions that guide user to decide to select or to omit requirements [46].

In sum, it is our hypothesis that implementing these six features VA enhancements (U1, U2, V6, V7, K4, and K5) over DDP leads to increase interactivity of requirements visualization to more accurate, informed, and defensible decisions. The next chapter describes a case study to test our hypothesis [46].

CHAPTER IV

CASE STUDY

We report in this section an exploratory case study [29] that we have worked with the SLDS (Statewide Longitudinal Data System) team at nSPARC to collect data. Our overall purpose is to qualitatively assess our VA support and gain operational insights of requirements-driven risk mitigation in practice. Thus, we first explain the case study design (Section 4.1). We then present the findings (Section 4.2) and discuss the threats to validity of our study (Section 4.3) [46].

4.1 Case Study Design

The main reason that we chose a case study as the basis for our experimental design is that the investigation of an existing element is suitable for addressing the 'how' and 'why' questions that can otherwise be difficult to answer through controlled experiments [13]. Essentially, the pros and cons of using VA in RE are only likely to be evident for the continuing real-world project under conditions that cannot be repeated in the lab. Peculiarly, the study of applying VA in RE cannot be independent from the organizational context, and the effects may take weeks or months to appear [46].

4.1.1 Rationale

We have designed an exploratory case study in collaborating with SLDS project team. According to Yin [46, 57], an exploratory case study is proper for preliminary inquiries in which it is not yet clear which phenomena are important or how to measure these phenomena. In our case, we were interested in understanding the practical impacts of VA on the RE tasks. To understand more about the support of visualization for RE tasks, it would be premature to try to measure the cost/benefit trade-off and the statistical significance of certain variables [46]. For our exploratory study, we expect to answer the following questions: 1) what RE tasks are in need of VA support; 2) how VA supports RE tasks; and 3) what final benefits can be achieved.

4.1.2 Objective

Our research objective is to make a more interactive tool by implementing the visualization features, such as filtering (V6) and annotation (V7), which are currently under supported. Our tool aims to support producing end-to-end, from-data-to decision values to its users. The development of the tool has been firmly coupled with the nSPARC SLDS project. Table 6 provides the basic information about the development efforts of the tool [46].

4.1.3 Procedure

Throughout February 2014, we held 4 meetings in nSPARC's workplace. Each meeting included one SLDS project member and one system administrator member; the requirements analyst participated in all the meetings. We considered these collaborative efforts as being similar to joint application development (JAD) [55] workshops where "knowledge workers and IT specialists meet to define and review the requirements for the system" [46]. In our study, reviewing and analyzing the requirements for nSPARC simultaneously has helped to define and clarify the requirements for our tool. This helps us to deploy the best desired features during meetings and to assess how the tool supported the RE tasks in short cycles [46].

Table 4.1 Joint application development for the tool[46]

Preparation by the research team	**Meeting date And duration**	**nSPARC participant(s)***	**Main Activities**
- Demo our tool on SLDS Life Track project	Feb 5,2014 and 1 hour	SLDS Project manager, System Admin. Requirements Analyst	• Explain about the Tool • Get their Feedback • Gather SLDS requirements by Project Manager (Appendix, Table 7) • Gather SA mitigations for those requirements (Appendix, Table 9)
- Run the tool on their requirements for single user scenario - Identify preferences	Feb 10,2014 and 2 hours	System Admin. Requirements Analyst	• Detect & act on extremity • Elicit RE tasks (Appendix, Table 7, 9)
Implement multiple stakeholders scenario Compare Multiple Stakeholder concerns	Feb 16,2014 and 1.5 hours	SLDS Project manager, System Admin. Requirements Analyst	• Diagnose & handle outliers • Elicit RE task (Appendix, Table 7, 9)
Perform Analysis - Make multiple step comparison - Support exploratory reasoning	Feb 25,2014 and 1 hour	SLDS Project manager, System Admin. Requirements Analyst	• Relate multiple artifacts (Appendix, Table 10, 11) • Refine the tool design

4.2 Findings

We collected 16 assets (Table A.1), 16 Risks (Table A.2), and 15 mitigations (Table A.3) mainly through observations, and during the 4 meetings with project manager, system admin, and requirement analyst (Table 4.1). Figure 4.1 shows the visualized solution space of all the possible ways that elicited mitigations can be selected based on DDP model (Chapter 2).

We took notes from meetings and transcribed all the interviews. We then collectively applied qualitative data analysis [49] to code and categorize the data. For coding, we segmented and allocated units of meaning to the location of data collected. For categorizing, we interpreted and assigned these units for answering our research questions. The qualitative data analysis was implemented by two researchers manually in a collaborative fashion [46].

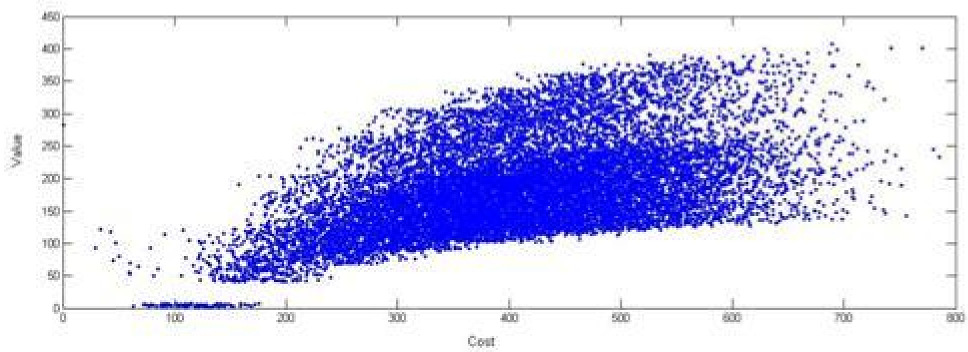

Figure 4.1 DDP solution space

We came up with two scenarios, single stakeholder and multiple stakeholders.

4.2.2 Single stakeholder

4.2.2.1 Non-Technical user

For single stakeholder scenario, we asked the SLDS project manager as the non-technical person to check preferred assets among all. Thus, the project manager decided to select A3, A4, A5, A12 and A15 and run the tool. Figure 4.2 shows the way the tool plots satisfied strategies in the six different independent regions or sub solution spaces. In other words, these six regions covered all the possible ways that these five assets were visualized, which provided huge difference in comparison with DDP model.

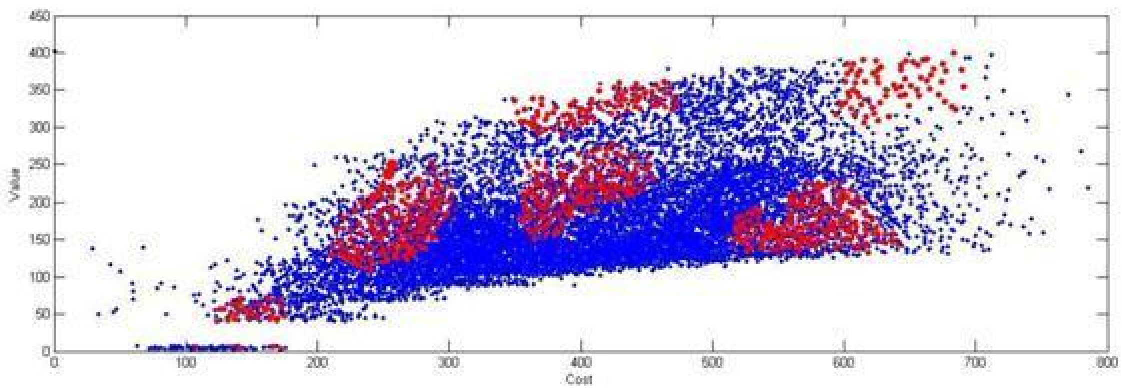

Figure 4.2 Six independent regions from

Human knowledge counts as part of the decision. At the same time, automatic analysis offered valuable options such as labeling the regions or annotation feature (V7) to provide straightforward analysis over visualization. Figure 4.3 presents this annotation.

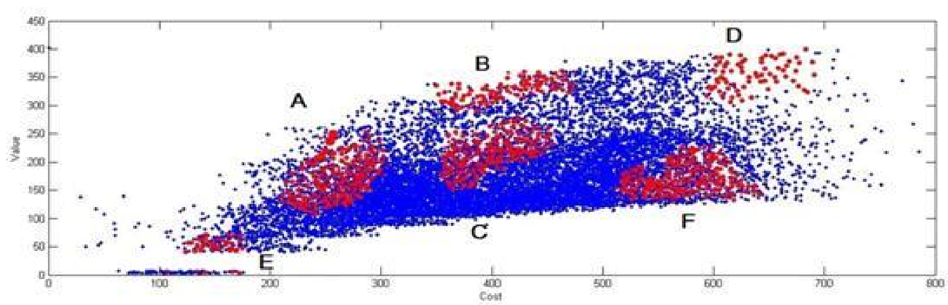

Figure 4.3 Using annotation in DDP

Table 4.2 shows the regions, their included assets and the satisfied assets for each region.

Table 4.2 Included assets in each region with its satisfied assets

Region	Included Assets	Satisfied Assets
A	A3, A4, A12, A15	A1, A5, A7, A11, A13, A14, A16
B	A3, A4, A15	A1, A7,A10, A16
C	A5,A8	A7, A10
D	A4, A5,A15	A7, A8, A10, A11,A12, A16
E	A5	A8, A10, A14
F	A4, A5, A15	A7, A8, A10, A11, A13, A16

We started comparing these regions based on cost and value in our second meeting to inform requirements analyst about the capability of our tool. Each region characterized different costs and values; furthermore, to provide valuable information about the region of interest (the sub set with low cost and high benefit for the software system) among these six regions, we started the evaluation with the presence of an analyst. Based on Table 4.2, region *E* provided the lowest value, and it just contained A5. Therefore, the analyst decided to delete A5 because it was covered by other regions. Region *F* also did not provide good values and efficient costs; furthermore assets A4 and A15 were covered with more normal costs and values by other regions.

Figure 4.3 shows the updated status of left regions. The tool allowed analyst to either continue the evaluation or confirm these three regions. The analyst decided to find out what the point is in keeping *D* while it brings the highest level of the cost this project. In accordance with Table 4.2, region *D* contained assets A4, A5, and A15, which were covered by A and B both, so we decide to delete D.

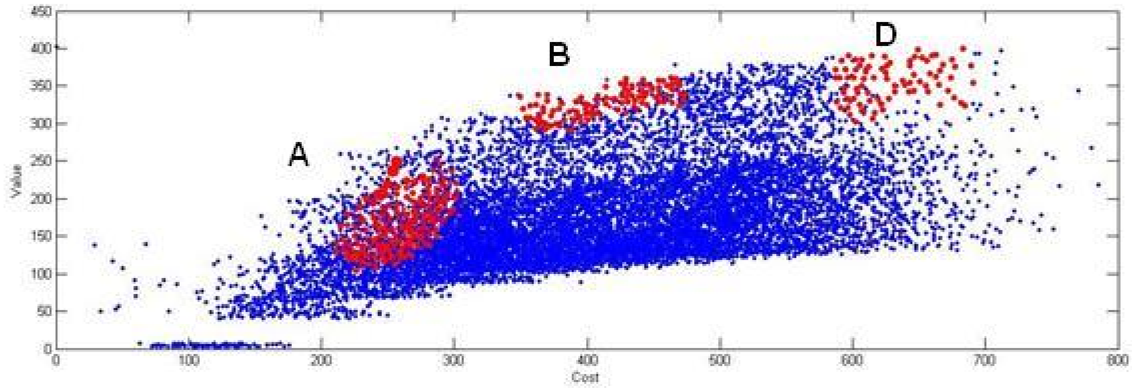

Figure 4.4 The left regions after third negotiation

The important point was, all the selected assets expected that assets A5 are covered by region *A* directly and A5 is one of the satisfied assets. In other words, we had all the preferred assets directly and indirectly at *A* which was located in the sweetspot [12] or region of interest based on DDP. Region A delivered high value and low cost while it covered the entire user's preference at the same time. The analyst believed keeping region *A* made up project inputs that were shown in Figure 4.4.

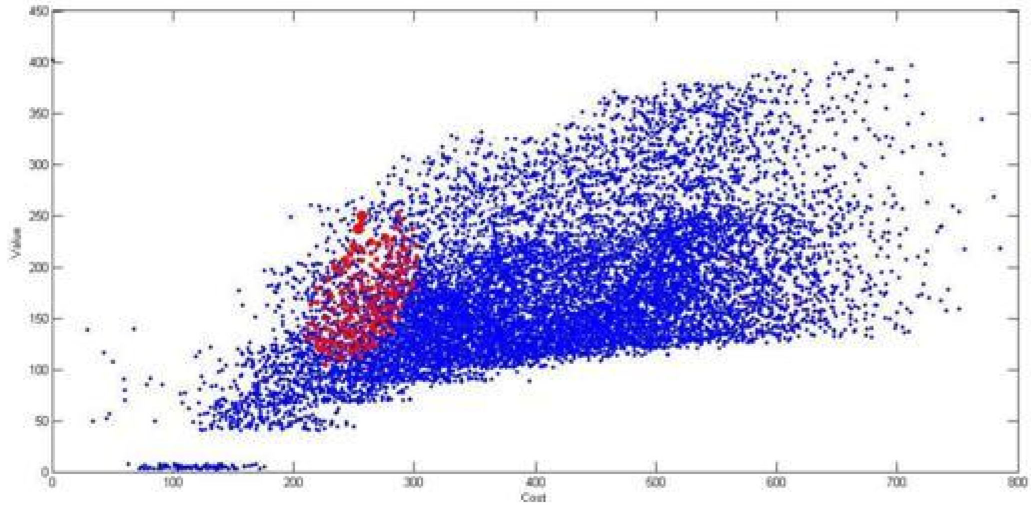

Figure 4.5 The selected region of interest by analyst

4.2.2.2 Technical user

For the second part of single user scenario, we held a meeting with system administrators to check the technical needs that could the protect system against failures during implementing assets for an SLDS team. The first meeting helped us to elicit all mitigations (Table A.3) by presence of a system administrator, a requirement analyst, and a project manager (Table 4.1).

Our second meeting discussion started from selecting preferred mitigations by a system admin. The system administrator deselected M5, M6, and M15. These holes show their deselected associated strategies. Figure 4.5 shows the visualized plot of the deselected mitigations.

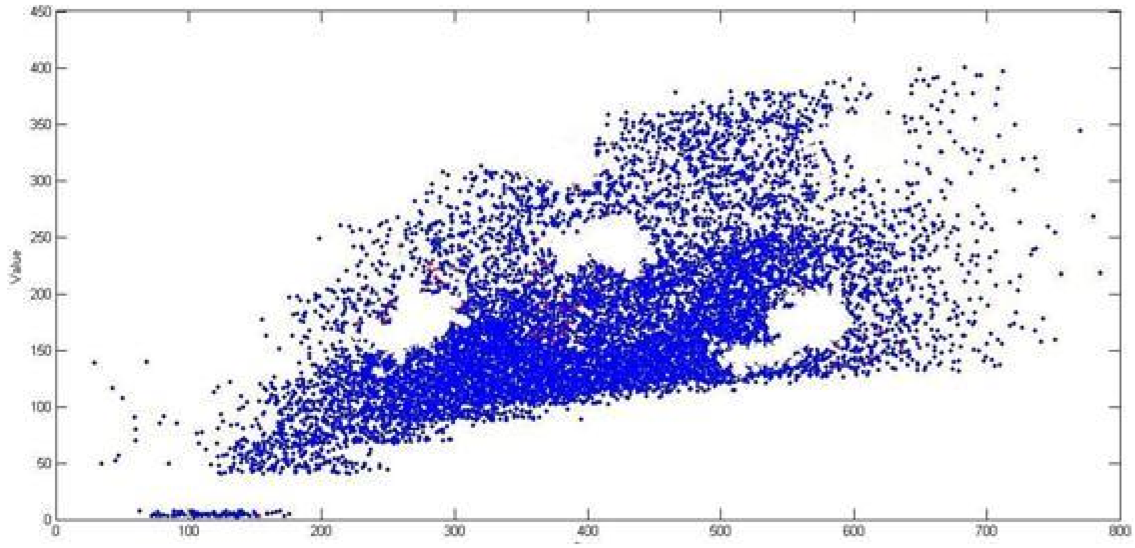

Figure 4.6 Preferred mitigations plot

Figure 4.5 illustrated the deleted strategies which did not contain any of these five mitigations.

The system analyst decided to divide the new sub solution space vertically based on low cost, medium cost and high cost, and picked one tradespace of strategies with the highest value from each. This separation helped to figure out the reason of applying strategies with even the highest cost for the software system. Figure 4.6 presents the new hypothesis.

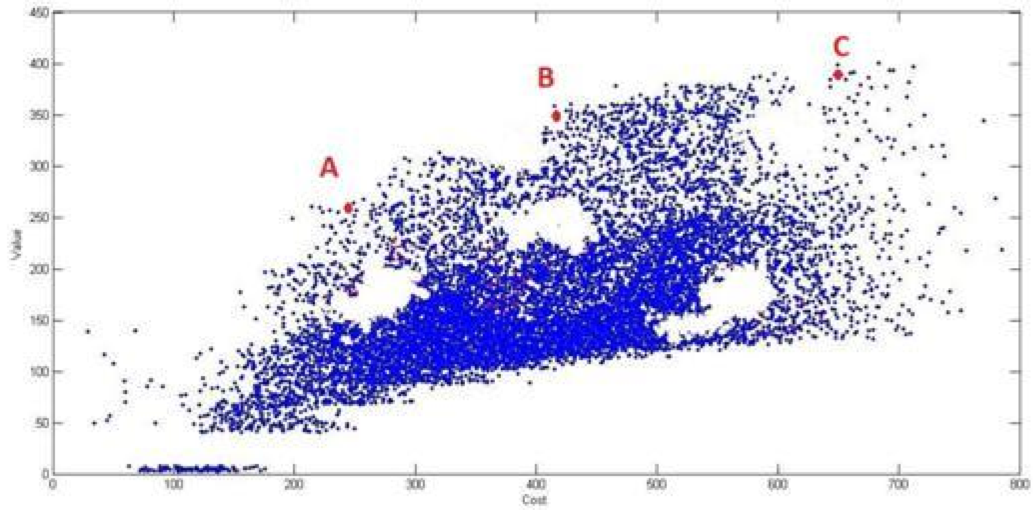

Figure 4.7 Divided sub solution space vertically with selected three strategy with highest value from each

Table 4.3 Selected strategies and their included mitigations

Region	Strategy	Mitigations
A	S10260	M1, M3, M4, M8, M9, M11, M12,and M14
B	S20682	M1, M4, M10,M11, M12,and M13
C	S18460	M2, M4, M7, M10, M11, and M13

Each of these strategy dots was derived from some selected mitigations. The tool enabled us to check the exact number of strategies and their satisfied mitigations for each of these dots in sub solution space. Region *A* provided high benefit and low cost; furthermore, it is located in the region of interest. Region *B* contained high benefit with medium cost, which was somehow valuable after region *A*, but region *C* was related to the highest cost and highest value. “Why do we need this region if we have to omit this part from evaluation every time? “

The analyst needed to figure out “what is the important reason in which keeping strategy with highest cost becames critical?” In other words, “which mitigations must be selected even if they need high cost for implementation?”

We found that some mitigation were common among these three strategies, selecting them could satisfy all the three regions, Figure 4.7. M4 and M11 are the mitigations that must be selected regardless of their location.

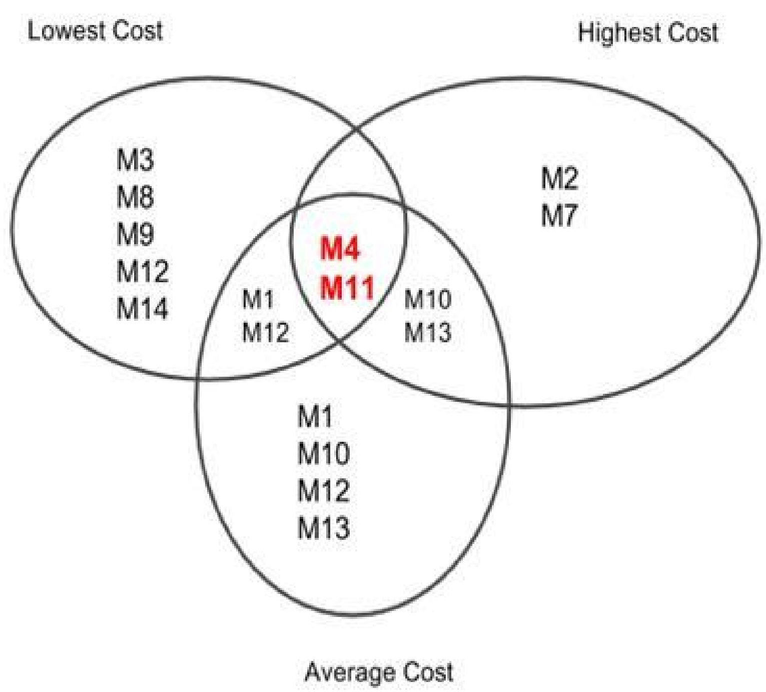

Figure 4.8 Common mitigations of three selected strategy in Table 13

4.2.3 Multiple stakeholders’ scenario

Our third meeting started with a question from the requirements analyst. The analyst stated that if each of these strategies has derived from some selected mitigations, then what happens if selecting these common mitigations satisfies the same strategies as what we selected in Figure 4.4?

We believed that the rationale behind selecting common mitigations was finding all the mitigations that are critical to select. In case of the first scenario by both project manager and the system administrator as our "user," we tried to reach three goals: 1) select requirements instead of mitigations performed a convenient way that does not face non-technical person with an ambiguous environment at first place; 2) both technical and non-technical stakeholders work with what they are familiar with; 3) they assess risk in SLDS project indirectly just by working on their professions.

If we wanted to imply the advantage of deploying common mitigations from the second meeting in Figure 4.4, we just needed to keep strategies from region *A* that are satisfied from selecting the common mitigations (M4, M11). Figure 4.8 shows the visualized plot of common strategies that satisfies the selected preferred assets and mitigations by both technical and non-technical users.

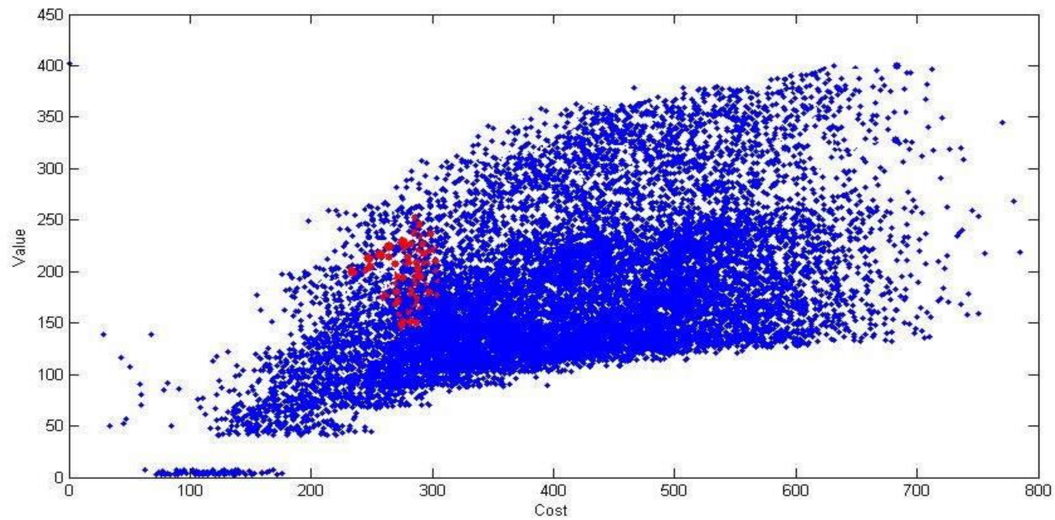

Figure 4.9 Combination of technical and non-technical single stakeholder scenario preferences

4.3 Threats to Validity

Several factors can disturb the efficiency of our exploratory case study. Construct validity concerns launching correct operational metrics for the concepts being studied [57]. The fundamental constructs in our case study consists of 'VA supports' and 'keeping assets on track in practice.' Our first construct, the VA supports are incorporated in our tool, which is developed with the objective of enhancing the state-of-the-art in visual requirements analytics (Chapter 3). The fact is, various VA tools have different stability and vulnerabilities, and we are not confident about if other tools may support RE. Our second construct, the interpretation of 'keeping requirements on track' is fixed in the actionable decisions (K5) made during the VA process [46].

A major limitation with our study design is the analysts and developers of the tool. Besides, the experimenter bias issue, researcher changes the study to find the expected result. In terms of experimenter bias problems, we mitigated such threads by using exploratory case study instead of exploratory and casual study. Also, we worked on pre-defined data analysis (coding and categorizing) with group meeting to apply more than one researcher analysis.

4.4 Learned Lessons from Case Study

Based on our case study, increasing interactivity between users and requirements visualization shall augment the user's knowledge discovery to leads more actionable decisions [46]. The lessons derived from this case study clarified the importance of user's capabilities to observe, relate, and ultimate goal, which is discovering the operational insights of requirements-driven risks mitigations and making the actionable and informed decisions. We learned that although machine's computations can augment, but still human is playing critical rules in making informed decision based on the situation and elicited data [46]. Therefore, taking users considerations into account helped us to elicit critical data among all by using a tool [46]. This tool helped to increase interactivity.

CHAPTER V

CONCLUSIONS

This thesis has proposed a framework to characterize and promote use of visual requirements analytics [46]. We apply our framework to define existing VA for RE solutions, which in turn helps identify areas for improvement. Based on this understanding, we develop a tool to enhance requirements-level Defect Detection and Prevention (DDP) with visual analytics and interactive visualization supports to RE practitioners. We further manage a case study to figure out how our tool might qualitatively assess VA support. By using two VA features: filtering (V6) (Chapter 3, section 3.2.3) and annotation (V7) (Chapter 3, section 3.2.4) in our tool, we could explore two features as knowledge: 1) scenario-based reasoning (K4) (Chapter 3, section 3.2.5); 2) actionable decision (K5) (Chapter 3, section 3.2.6) which these visualization and knowledge features increases visual interactivity could lead to actionable decisions [46].

From our experience, we understand that VA is valuable in servicing requirements analysts, decision makers, and other stakeholders to rapidly extract insights from the flood of data. We see how using our tool helps to limit the solution space and make a sub space from all critical data. Using the sub solution space for stakeholders and analysts is more efficient in terms of time, and cost. We decide to polish our tool development and to improve the extensibility of the underlying visualization models as our future work. We also plan to conduct more in depth empirical studied to explore the cost and value tradeoffs in requirements- driven risk mitigation in practice, based on our future collaboration with nSPARC SLDS team. Finally, we want to study the possible usage limitations and find fundamental strategies to overcome the barriers in order to deliver the full potential of VA approaches in RE [46].

REFERENCES

[1] Y. Akao, Quality Function Deployment, Productivity Press, Cambridge, Massachusetts, 1990.

[2] E.Alpaydin, " Introduction to machine learning," The MIT Press, Cambridge, 2004

[3] V. Basili, G. Caldiera, and RH. Dieter, "The goal question metric paradigm," *Encycl Softw Eng,* 2:528–532, 1994

[4] B. Boehm, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, ACM, V.11(4), pp.14-24, Aug. 1986.

[5] RPL. Buse and T. Zimmermann, "Information needs for software development analytics," *International conference on software engineering (ICSE),* Zurich, Switzerland, pp 987–996, 2012

[6] A. Cailliau and A. V. Lamsweerde, "A probabilistic framework for goal-oriented risk analysis," *the Int'l Requirements Engineering Conference*, pp. 201-210, 2012.

[7] P. Caire, N. Genon, P. Heymans, and M.D. Laurence, "Visual notation design 2.0: towards user comprehensible requirements engineering notations," *International requirements engineering conference (RE)*, Rio de Janeiro, Brazil, 2013

[8] S.K. Card, J. Mackinlay, and B. Shneiderman, "Readings in information visualization: using vision to think," Morgan Kaufmann, Los Altos, 1999.

[9] P. Caserta and O. Zendra, "Visualization of the static aspects of software: a survey," *IEEE Trans.Vis.Comput.Graph*, V.17 (7), pp.913–933, Aug. 2010.

[10] J. Cleland-Huang, and R. Habrat,"Visual support in automated tracing," *International workshop on requirements engineering visualization (REV),* article no. 4. New Delhi, India, 2007.

[11] JR. Jr. Cooper, S. W. Lee, RA. Gandhi, and O. Gotel,"Requirements engineering visualization: a survey on the state-of-the-art," International workshop on requirements engineering visualization (REV), Atlanta, GA, USA, pp 46–55, Sept. 2009.

[12] S. L. Cornford, M. S. Feather, and K. A. Hicks, “DDP - A tool for life-cycle risk management”, IEEE Aerospace Conference, Big Sky, Montana, Colorado, pp. 74-83, Mar 2001.

13] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian ,“Selecting empirical methods for software engineering research”, *in Guide to Advance Emprical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer, 2007, pp. 285–311..

[14] M. S. Feather ,“A Quantitative Risk-based Model for Reasoning over Critical System Properties”, *Requirements for High Assurance Systems Workshop*, Essen, Germany, Sept. 2001.

[15] M. S. Feather, T. Menzies, and J. R. Connelly, Matching software practitioner needs to research activities, APSEC 2003: 6-16.

[16] M. S. Feather, K. A. Hicks, R. M. Mackey, and S. Uckun, “Guiding Technology Deployment Decisions using a Quantitative Requirements Analysis Technique” RE, pp.271-276, 2008.

[17] M. S. Feather and J. Wilf, Visualization of Software Assurance Information, HICSS 2013: 4948-4956.

[18] M. S. Feather, S. L. Cornford, and K. A. Hicks, “ Descoping”, Proceedings 27th Annual NASA Goddard/IEEE International Software Engineering Workshop, Pasadena, CA, USA, Dec 2002 , pp.107-113, Dec 2002.

[19] M. S. Feather, and S.L. Cornford, et al., “Experiences using Visualization Techniques to Present Requirements, Risks to Them and Options for Risk Mitigation,” REV 06, 14th RE, pp. 10, 2006.

[20] R.Gacitua, P.Sawyer, and V.Gervasi, “Relevance-based abstraction identification: technique and evaluation,” *Requir Eng,* 16(3):251–265, 2011

[21] K. Gallagher, A. Hatch, and M. Munro, “Software architecturevisualization: an evaluation framework and its application,” IEEE Trans Software Engineering, V.34(2), pp.260–270, March/April.2008.

[22] RA. Gandhi, and S-W. Lee , “Visual analytics for requirements driven risk assessment,” *International workshop on requirements engineering visualization (REV)*, article no. 6, New Delhi, India, 2007.

[23] L. Goldin and DM. Berry, “AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation,” *Autom Softw Eng,* 4(4):375–412, 1997.

[24] S. Haag, M. Cummings, and D. J. McCubbrey, Management Information Systems for the Information Age. McGraw-Hill, 2003.

[25] J. Han and M. Kamber, “Data mining: concepts and techniques,” Academic Press, London, 2000.

[26] CD. Hansen and CR. Johnson, “Visualization handbook,” *Academic Press*, London, 2004.

[27] P. Heim, S. Lohmann, K. Lauenroth, and J. Ziegler, “Graph-based visualization of requirements relationships,” *International workshop on requirements engineering visualization (REV)*, Barcelona, Spain, pp 51–55, 2008.

[28] C. Heitmeyer, B. Labaw, and D. Kiskis, “Consistency Checking of SCR-Style Requirements Specifications”, Proceedings 2nd IEEE International Symposium on Requirements Engineering, York England, pp.56-63, Mar 1995.

[29] J.Horkoff, and E.Yu, “Visualizations to support interactive goal model analysis,” *International workshop on requirements engineering visualization (REV),* Sydney, Australia, pp 1–10, 2010

[30] J. Karlsson, and K. Ryan, “A Cost-Value Approach for Prioritizing Requirements,” *IEEE Software*, pp. 67-74, Sept /Oct. 1997.

[31] DA. Keim, F. Mansmann, J. Schneidewind, JJ. Thomas, and H. Ziegler“Visual analytics: scope and challenges. In: SJ. Simoff, MH. B. hlen, A. Mazeika (eds) Visual data mining—theory, techniques and tools for visual analytics,” Springer, Berlin, pp 76–90, 2008.

[32] I. Kwan, D. Damian, and M-A.Storey, “Visualizing a requirements- centered social network to maintain awareness within development teams,” *International workshop on requirements engineering visualization (REV)*, article no. 7 Minneapolis/St. Paul, MN, USA, 2006.

[33] M. D. Laurence, P. Heymans, and R. Matulevicius, “Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation,” Requirement Engineering Journal, Springer London, V.15 ,pp.141–175, June. 2010.

[34] M. D. Laurence, P. Heymans, and R. Matulevicius, “Improving the effectiveness of visual representations in requirements engineering: an evaluation of i* visual syntax,” *International requirements engineering conference (RE),* Atlanta, GA, USA, pp 171–180, 2009.

[35] MD. Laurence, “The ‘‘physics’’ of notations: toward a scientific basis for constructing visual notations in software engineering,” *IEEE Trans Software Eng,* 35(6):756–779, 2009

[36] A. Mahmoud and N. Niu, "Source code indexing for automated tracing," *International workshop on traceability in emerging forms of software engineering (TEFSE)*, Honolulu, HI, USA, pp 3–9, 2011.

[37] R. Matulevicius and P. Heymans, "Visually effective goal models using KAOS," *International workshop on requirements, intentions and goals in conceptual modeling (RIGiM),* Auckland, New Zealand, pp 265–275, 2007.

[38] S. McConnell, "From the editor – an ounce of prevention", IEEE Software, 18(3): 5-7, 2001.

[39] T. Menzies and T. Zimmermann, "Software analytics: so what?," *IEEE Softw,* 30(4):31–37, 2013

[40] G. Mussbacher, D. Amyot, and M. Weiss, "Visualizing early aspects with use case maps," *Trans Aspect-Oriented Softw Dev,* 3:105–143, 2007.

[42] J. Mylopoulos, L. Chung, S. Liao, Wang, H., and E. Yu, "Exploring Alternatives during Requirements Analysis", IEEE Software 18(l), pp. 92-96, Jan/Feb 2001.

[42] J. Nattoch Dag, V. Gervasi, S. Brinkkemper, and B. Regnell, "A linguistic–engineering approach to large-scale requirements management," *IEEE Softw,* 22(1):32–39, 2005.

[43] N. Niu and S. Easterbrook, "Extracting and modeling product line functional requirements," *International requirements engineering conference (RE),* Barcelona, Spain, pp 155–164, 2008.

[44] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap", Proc. of the Conference on the Future of Software Engineering, pp. 35-46, 2000.

[45] O. Raz, and M. Shaw, "An Approach to Preserving Sufficient Correctness in Open Resource Coalitions", Proceedings of the 10" International Workshop on Software Specification and Design, San Diego,California, pp. 159-170, Nov. 2000.

[46] S. Reddivari, S. Rad, T. Bhowmik, N. Cain, and N. Niu, "Visual requirements analytics: a framework and case study," Requirement Engineering Journal, Springer London, Nov 2013.

[47] S. Reddivari, Z.Chen, and N.Niu,"ReCVisu a tool for clustering based visual exploration of requirements," *International requirements engineering conference (RE)*, Chicago, IL, USA, pp 327–328, 2012.

[48] T. Reinhard, S. Meier, and M. Glinz, “An improved fisheye zoom algorithm for visualizing and editing hierarchical models,” *International workshop on requirements engineering visualization (REV)*, article no. 9, New Delhi, India, 2007.

[49] L. Richards, Handling Qualitative Data: A Practical Guide. Sage. 2005.

[50] R. Seater, D. Jackson, and R. Gheyi,” Requirement progression in problem frames: deriving specifications from requirements,” *Requir Eng*, 12(2):77–102, 2007.

[51] MJ. Shepperd, Q. Song, Z. Sun, and C. Mair, “Data quality: some comments on the NASA software defect datasets,” *IEEE Trans Softw Eng,* 39(9):1208–1215, 2013.

[52] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” *IEEE symposium on visual language (VL)*, Boulder, CO, USA, pp 336–343, 1996.

[53] JJ. Thomas, KA. Cook, "Illuminating the path: the research and development agenda for visual analytics," IEEE Press, Alamitos, 2005

[54] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, “Fault Tree Handbook”, U.S. Nuclear Regulatory Commission NUREG-0492, Jan 1981.

[55] J. Wood and D. Silver, Joint Application Development. Wiley, 1995.

[56] K. Wnuk, B. Regnell, and L. Karlsson, “What happened to our features? Visualization and understanding of scope change dynamics in a large-scale industrial setting,” *International requirements engineering conference (RE)*, Atlanta, GA, USA, pp 89–98, 2009.

[57] R. K. Yin, Case Study Research: Design and Methods, Sage Publications,2003.

[58] SK.Yu Eric, “Towards modeling and reasoning support for early-phase requirements engineering,” *International symposium on requirements engineering (RE)*, Annapolis, MD, USA, pp 226–235, 1997

APPENDIX A

TABLE OF ELICITED ASSETS, RISKS, MITIGATIONS, RISK-ASSETS, AND MITIGATION-RISK

Table A.1 Assets table

Assets	Description	Value
A1	Conduct an early childhood education and healthcare system	90
A2	Count the population of children from low income family	41
A3	Improve healthcare system	76
A4	Conduct an immunization service	73
A5	Get a monthly health check certificate	30
A6	Be aware of prematurity and genetic conditions	90
A7	Be aware of physical health condition of disabled children during their first year	79
A8	Check mother's health before and after having baby	89
A9	Need early competent learning system for children under school age	80
A10	Check the concentration and mental health	75
A11	Check the basic skills of children every two months before school age	55
A12	Check communication skills of children during kindergarten and elementary school ages	51
A13	Evaluate the quality of the relationship between teacher and children	50
A14	Developing children from different perspective (cognitive, social. emotional, creative, and mental)	80
A15	Hire highly qualified and trained teachers	70
A16	Train teacher on communication with children and teaching skills	73

Table A.2 Risks table

Risks	Description
R1	Grow up in a family with low financial and emotional support
R2	Grow in a environment without attention to the early childhood development
R3	Don't learn how to interact and communicate with others in a committee
R4	Deploy a weak healthcare system
R5	Have children with less that 20% of life in healthy condition
R6	See unmoral low weight for new born baby
R7	See children with weak level of vitamin, and basic health elements
R8	Need to know the low level of concentration as an issue on children because of being in high level of energy
R9	Check if educational system strategy is not a good selection for group of children
R10	No checking the basic skills doesn't let parents and practitioners to understand wrong behavior, emotion, and wrong development characteristic of children
R11	Deploy low-quality educational system causes making weak communication between teacher and student in which teachers will not be able to transfer their knowledge to children
R12	Make a low knowledge foundation for children
R13	Hire low qualify teacher doesn't make any special outcome result from children
R14	Grow up weak children from different perspective (cognitive, social. emotional, creative, and mental)
R15	Having an addicted family to the drug
R16	Born baby from an physical or mental ill mother

Table A.3 Mitigations table

Mitigations	Description	Cost
M1	Set an unique standard for educational system	80
M2	Classify children in the related group	50
M3	Hire teachers that pass an standard test which their knowledge, skills, and health condition be checked through that test	61
M4	Advertise some workshop to update teachers and practitioners with new teaching and communication skills	45
M5	Classify teachers in some level of proficiency and set a workshop for parent	38
M6	Provide governmental funding to the local healthcare systems to be facilitate with different testes which is necessary for children	30
M7	Set some medical tests that children should pass every 2 years	49
M8	Make an interactive educational system	75
M9	Check the relationship between children and teachers	70
M10	Teach children how to have social engagement and encourage them for further learning	24
M11	Have some opening for noncitizens children in educational system	78
M12	Provide some insurance coverage for health check	69
M13	Offer individual learning to children with especial need to encourage children in order to improve their self-confidence and motivation	76
M14	Have some special teaching strategy which is a good step in accordance with the last approach in that area	40
M15	Support poor families financially by government in order to help them grow better generation	65

Table A.4 Assets-Risks table

Assets	Connected Risks	Weight
A1	R1, R2, R5, R6, R7, R14	0.1, 0.5, 0.4, 0.2, 0.3, 0.5
A2	R2, R4, R6, R7, R14, R15, R16	0.3, 0.7, 0.4, 0.4, 0.5, 0.1, 0.8
A3	R4, R5, R6, R7, R16	0.3, 0.5, 0.2, 0.4, 0.1
A4	R4, R5, R6, R7	0.8, 0.2, 0.3, 0.4
A5	R4, R5, R6, R7, R16	0.7, 0.1, 0.2, 0.4, 0.5
A6	R4, R5, R6, R7, R16	0.9, 0.2, 0.1,0.4, 0.3
A7	R2, R4, R6, R7, R14, R15, R16	0.4, 0.2, 0.3, 0.5, 0.1, 0.3, 0.7
A8	R5, R6, R7, R16	0.4, 0.2, 0.5, 0.1
A9	R2, R3, R10, R11, R12	0.3,0.2,0.5,0.8,0.3
A10	R1, R2, R4, R8, R14	0.1, 0.3,0.4,0.7,0.5
A11	R8, R12, R14	0.3, 0.5,0.2
A12	R2, R3, R10, R13	0.3, 0.2, 0.5, 0.6
A13	R11, R12, R13	0.6, 0.5, 0.4
A14	R11, R14	0.3, 0.4
A15	R8, R 11, R12	0.8, 0.6, 0.7
A16	R8, R9, R11	0.3, 0.2, 0.4

Table A.5 Risks-Mitigations table

Risks	Connected Mitigations	Weight
R1	M1, M2, M6, M8, M12, M13, M15	0.5, 0.3, 0.7, 0.4, 0.7, 0.4, 0.5
R2	M1, M2, M4	0.2, 0.6, 0.7
R3	M8, M10	0.4, 0.5
R4	M6, M7	0.4, 0.5
R5	M6, M7	0.6, 0.6
R6	M6, M7	0.6, 0.7
R7	M6, M7	0.4, 0.5
R8	M1, M2, M4, M9	0.3, 0.5, 0.4, 0.2
R9	M1, M3, M5, M13	0.4, 0.2, 0.5, 0.1
R10	M5, M8, M14	0.3, 0.1, 0.6
R11	M2, M3, M5, M8, M9	0.2, 0.4, 0.6, 0.7, 0.8
R12	M1, M3, M5, M13, M14	0.2, 0.3, 0.1, 0.4,0.2
R13	M2, M3	0.4, 0.7
R14	M4, M5, M7	0.2, 0.3, 0.1
R15	M15	0.2
R16	M6, M7, M12, M15	0.7, 0.6, 0.5, 0.3