



An Evaluation of the ReleasePlanner on the Treatment of Security and Evolving Requirements

Saad Bin Saleem and Yijun Yu and Bashar Nuseibeh and Anne De
Roeck

08 December, 2010

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

<http://computing.open.ac.uk>

Technical Report
An Evaluation of the `ReleasePlanner` on the
Treatment of Security and Evolving Requirements

Saad Bin Saleem, Yijun Yu, Bashar Nuseibeh and Anne De Roeck

December 7, 2010

Contents

1	Introduction	2
1.1	Research Question	3
1.2	Research Methodology	3
2	Background	4
2.1	Security Requirements	4
2.2	Strategic Release Planning	5
2.2.1	ReleasePlanner- a Strategic Release Planning Tool	5
2.2.2	Definitions and Terms	7
3	Case Study - a Meeting Scheduler System	8
3.1	Data Collection	8
3.1.1	Optimize Solution Set	9
3.1.2	Other Contributing Factors	14
3.2	Data Analysis	17
3.2.1	Working of the ReleasePlanner	17
3.2.2	Treatment of requirements	19
4	Discussion	21
5	Conclusion	22
6	Acknowledgement	23
7	Appendix	26
7.1	Case Study Input	26
7.1.1	Customer Demands	26
7.1.2	Requirements	26
7.1.3	Organizational Resource Estimations	27
7.1.4	Requirements Estimations	28
7.2	Instructions for Stakeholders to Use Prioritization Section of the ReleasePlanner	29

List of Figures

1	Alternative strategic release plans.	10
2	Quality evaluation of alternative release plans.	10
3	Stakeholder excitement evaluation for one release plan and multiple stakeholders.	12

4	Stakeholder excitement evaluation for one stakeholder and multiple plans.	12
5	Resource consumption of alternative plans.	13
6	Comparison of structural commonalities between alternative plans.	14
7	Overall resource usage summary.	15
8	A summary of conformance and selected planning criteria. . .	16
9	The most conflicted requirements from stakeholder voting. . .	16
10	The most conformant requirements from stakeholder voting. .	17

List of Tables

1	Functional requirements	27
2	Security requirements	28
3	Evolving security requirements	29
4	Total organizational resource estimations in terms of cost and time	29
5	Functional requirements' estimations	30
6	Security requirements' estimations	31
7	Evolving security requirements estimations	31

Abstract

The `ReleasePlanner` is a state-of-the-art release planning tool which generates the ‘optimal’ plan for developing a given list of system requirements, by ‘maximizing’ the utilization of resources such as developers time and skills within the given constraints such as project cost and duration. For a system that is both security-sensitive and evolving, however, it is unclear whether such treatments of general requirements in the `ReleasePlanner` can still optimize the activities without considering that security requirements are often constraints on functional requirements, and that all these requirements are subject to change. To clarify this, we used the meeting scheduling system for an office building (i.e., the Janie Lee Building) as an example project in the report. Specifically, we listed both security and functional requirements as the input for the `ReleasePlanner` to generate the plans that were ‘optimal’, yet they were found impractical in reality. As a result, we identified several pitfalls that might help adopters of the `ReleasePlanner` to avoid when security and evolving requirements are important to their projects.

1 Introduction

The end-user security is becoming increasingly important and a lot of Scientists emphasized on eliciting security requirements of end-users for the development of secure software systems. There are many different ways to represent security requirements for early analysis of the secure software but existing approaches are only limited to specifications [4]. The management of security requirements is important as security is dynamic element of the system and changes by time. According to Mens et al. incremental/iterative software development methods can play central role in management of any types of requirements [5]. Similarly, in the case of security such methods can become more effective to manage requirements by delivering the most important requirements earlier and to update evolving security requirements due to exposed vulnerabilities.

Release planning (RP) is the most important step in incremental development to decide which requirements to be delivered when. The purpose of RP is to select optimal set of requirements in sequence of candidate releases and plan delivery of each individual release [1]. The selection of requirements in subsequent releases is called strategic release planning and have central place in defining overall goal of the product. There are many approaches of strategic release planning, some are implemented in the form of tool and used by industry, for example Evolve* [8] is a recognized model of release planning and implemented in the form of tool **ReleasePlanner**¹

ReleasePlanner is a decision support tool generates optimal release plans for delivering product in increments. Release-planner is applied in different settings on various sets of requirements to check optimality of the solutions and efficiency of the tool [9]. This tool is never applied to plan a set of security requirements according to best of our knowledge. One can argue that security requirements are subclass of other requirements and already such requirements are handled by **ReleasePlanner** but security requirements are different and more challenging from other requirements according to definition provided by Haley et al [2].

The changes in requirements are very common due to changes in system's context and boundaries. According to Thun et al, the changes in security requirements are more difficult to manage and can impact on the whole system's functionality [11]. The release planning process support adjustment of evolving requirements after post release analysis through re-planning. But, knowing the usefulness of **ReleasePlanner** to adjust changes at strategic

¹<https://www.releaseplanner.com/>

level is still an open question to answer. Specifically, the adjustments of evolving security requirements is of particular interest in this study.

Therefore, purpose of this study is to apply **ReleasePlanner** on a case Meeting Scheduler System to know its working to handle functional, security and evolving security requirements.

The report is organized into following sections. The research question and proposed research methodology is described in this section. The background of release planning and description of basic functions of **ReleasePlanner** is provided in Section 2. Section 3 consists on the data collection, results and analysis of the case study. Discussion and future work are presented in Section 4, while a detail list of input to release-planner is provided in section 5.

1.1 Research Question

The goal of research is to get an understanding of **ReleasePlanner** to develop a strategic release plan in existence of functional, security and evolving security requirements. The following is a research question to fulfil this goal.

RQ: *How ReleasePlanner treats functional, security and evolving security requirements for strategic release planning*

It is important to know the working of **ReleasePlanner** for developing a strategic release plan before understanding how it deals with different types of requirements. Therefore, the following sub-question is formulated for getting an overall understanding of the tool.

SubRQ: How **ReleasePlanner** develops a strategic release plan?

1.2 Research Methodology

An exploratory case study is selected as a method of investigation to understand the working of **ReleasePlanner** in existence of functional, security and evolving security requirements [7]. The data is collected by capturing screen shots of operations and steps of the tool. The different sets of inputs are provided to tool for data collection that includes requirements, requirement's estimations and stakeholder preferences. The results are analyzed qualitatively through interpretation and observations.

2 Background

In this section basic concepts related to security requirements and strategic release planning are discussed, additionally a brief description about the functionality of industrial tool `ReleasePlanner` is also provided.

2.1 Security Requirements

Requirements Engineering (RE) is one of the important phases of software development. The purpose of RE is to facilitate in better understanding of the proposed system through elicitation, specification and analysis of stakeholders' requirements. Recently, the area of Security Requirements Engineering (SRE) emerged by combining RE and security engineering to elicit, analyze and specify application specific security requirements.

According to Haley et al.[2] security requirements are constraint on the functional requirements to operationalize one or many security goals of the system. A security goal is to protect important assets of the system from potential threats to avoid any harm. Here security analysts can define system's assets, harm and potential threats according to their situation. One can argue that security goal is more like a higher level protection aim, while security requirement can be considered as objectives to achieve the desire aim.

Security requirements are categorized into two groups non-functional(SR) and functional (SFR) by the common criteria [6]. The SR is considered as behavior specification of desire protection to prevent or mitigate the threat, on the other hand SFR is a security mechanism to realize the behavior specification.

The following are some approaches used to represent security requirements for early analysis.

In Haley's approach, the security requirements are elicited by constructing security goals from the owner's perspective. Then security arguments are developed about the elicited requirements to check their usefulness in preventing realization of attacker's goals in operating domain of the system.

Lin et al. [3] have introduced the idea of abuse frames and anti requirements to identify security requirements in the problem world instead of looking everything in the solution world. The approach is useful to analyze security at early stage of the system from system's owner and attacker's perspective.

Another goal oriented approach by Axel et al.[12] helps in identification of security requirement from system's owner and attacker's perspective by

the construction of intentional model/anti models.

2.2 Strategic Release Planning

Strategic release planning is concerned with selection of requirements in subsequent releases to define a product road map. The aim of strategic RP (road-mapping) is not only to select requirements in subsequent releases but balance different constraints or parameters to fulfil the demands of all stakeholders (Customer and development organization). Svahnberg et al have described two types of technical and non-technical constraints are involved during strategic RP [10]. Here technical constraints are referred as requirements dependency, precedence etc, while non-technical means resource, risk constraints etc.

A release plan is refined, updated and re-planned at both levels (strategic and operational) after the delivery of each release through post release analysis. Post release analysis is conducted for taking customer feedback, defect detection and process improvement etc. It can help in re-prioritizing of requirements during re-planning of strategic plan.

There are many ad-hoc (based on human intuition), systematic (based on some computational method) and hybrid (combination of ad-hoc and systematic) approaches to plan and re-plan a release. But some models are validated in Industry and implemented in the form of tool while others have no tool support. Evolve* is one of the many models which is implemented in the form of tool named **ReleasePlanner** for strategic release planning. It is mostly recognized as decision support tool because it generates alternative release plans, which means it act as a facilitator. Eventually, the product managers or people involved with development take the final decisions about selection of requirements. **ReleasePlanner** claimed to be useful for strategic release planning of any planning scenarios including software requirements. The following is a brief description of **ReleasePlanner**.

2.2.1 ReleasePlanner- a Strategic Release Planning Tool

The purpose of this section is to provide details about the interface of **ReleasePlanner**. The interface of **ReleasePlanner** is divided into three main sections for performing different operations.

The purpose of first section “My projects” is to manage new and ongoing projects. In this section, an existing project can be opened by clicking on the desire project. On the other hand for creating new project, there are lot of required steps by the **ReleasePlanner**. The information about max-

imum number of planned releases and object of planning like requirement, feature, service etc are required by the **ReleasePlanner** during first step of the project creation. In second step, it requires to enter desire voting and resource calculation criteria. Then it is necessary to specify the planning criteria for stakeholders' voting in third step for creating the project. There are five planning criteria supported by **ReleasePlanner** including "Urgency", "Value", "Risk", "Time to market" and "Volatility". The assignment of weights to each specified planning criteria on the scale of zero to nine is fourth step. Where zero is no weight and nine is referred to extremely high weight. Fifth and final step consists of assigning weight to each release on the scale of zero to nine similarly as assigning weight to the planning criteria.

The second section "My project information" contains all project relevant information like different inputs before the results or solution stage. In other words, this section includes information about available and required resources, list of requirements to plan, any grouping of requirements, requirement dependencies and one part of section is related to handle stakeholders. Resource type and capacities is one of the important parts of this section where total available cost and time are logged for each release. It is essential step as overall budget of release and selection of requirements are determined later in solution stage based on the input provided here. The next key part of this section is requirement dependencies where **ReleasePlanner** allows you to define coupling and precedence constraints for each requirement. It is also considered as necessary step to fulfil desire goal of release. Another part of this section is stakeholders' area where different information about stakeholder is available and stakeholders can be added or deleted. The precise information about stakeholders' vote based on planning criteria and analysis of each vote for the requirement is important part of this section. In this way, it is a lot easier for development organization to know participation of stakeholders' in prioritization, conformance of stakeholders' voting, any common or conflicting requirements and each stakeholders' excitement level for the voted requirement.

The third section "My project results" is dedicated to generate, view and analyze results of the project. There are three important parts "Optimize solution set", "Analyze solution set" and "What if analysis" in this section. First part allows generating alternative release plans of the selected project, while in second part the deep analysis of generated solution set is possible. Deep analysis of each solution set like number of requirements selected in each release, optimality of each solution, total number of delivered requirements, resources consumed in release and excitement level of stakeholders can be viewed. Third part of this section is important and interesting as it

help to change values of different constraints for the planning and to look how changing value of a constraint can change the whole set of solutions.

2.2.2 Definitions and Terms

The following are some definitions and terms commonly used in the `ReleasePlanner`.

Release A small chunk of complete product or one milestone for product development

Planning Object Any object to plan the releases like requirement, features and services

Resource type and capacities Resource type and capacities: Resources required in different phases of software development can be categorized into different types like design, development, testing resources etc. `ReleasePlanner` allow to define any resource types and also the resource constraints for each type. While capacities are referred as the available resources of the organization. Estimations of required resources for each planning object(requirement)according to resource type are essential for defining resource constraints. Cost and time are commonly defined resource constraints. Risk can be defined as a resource type. Here risk is in terms of implementing a requirement or overall risk associated with the release

Requirements dependencies It is a coupling and precedence relationship between two or more requirements, coupling means that both inter-related requirements needs to be released together, while precedence relationship holds when one requirement can not be released before releasing the other.

Stakeholders People involved with the projet like customers, end-users, other peoples exits in the environment of the system and legal experts.

3 Case Study - a Meeting Scheduler System

The Jennie Lee Building (JLB) is home of computing department at the open university. Currently, there is no automated system in place for scheduling of meetings instead MS office tools like Excel are used for this purpose. Departmental secretaries are responsible for booking and assignment of rooms for any meeting request by the staff or students. The secretaries make sure that meeting rooms can only be accessible during the meeting hours and rooms remain locked in any other timing. They are using door keys for this purpose. Usually, the chair person of meeting takes the key from secretaries and return it after completion of meeting. One staff member of the building is participating as customer in this study. The included staff member have relevant knowledge of the system and he has provided all demands for the new meeting scheduler system. The following operations are performed by on customer demands for preparing inputs for the **ReleasePlanner**. The list of customer demands, requirements, requirement estimations and dependencies can be viewed in appendix 7.

- Elicitation and categorization of functional, security and evolving security requirements from customer demands
- Resource estimations for implementing each requirement
- Coupling and precedence relationship between requirements
- The delivery of complete product in total number of releases
- Estimations of total available time and cost can be allocated to each release
- Stakeholder preferences of requirements

In the next stage, a new project "Meeting Scheduler JLB" was created in the **ReleasePlanner** to provide all the inputs for data collection.

3.1 Data Collection

All necessary information for data collection was provided to **ReleasePlanner**, for further details about the inputs look at appendix 7. It was decided to deliver the whole system into five releases. The following data is collected to analyze the generated strategic plan for planning of five releases.

3.1.1 Optimize Solution Set

The **ReleasePlanner** automatically generates an optimized release plan with five different alternatives. It is possible to analyze the generated solution by using “Analyze solution set” section of the tool. The “Analyze solution set” section is divided into five steps. It also generates a report in the pdf version to facilitate analysis of a solution. The following are snap-shoots of each step to explain the results for developing a strategic plan to deliver system into the five releases.

The five alternative strategic release plans are shown in Figure1, one or a set of requirements are selected in each alternative plan. For example most of the requirements are selected to be delivered in first and fifth release in alternative1. The figure is divided into three main columns, first column is showing a unique ID of requirement, a name of requirement is shown in the second column, while third column is showing release plans and further divided into five columns to show all the alternatives. It is important to note that all five releases are represented with a different color from the other. The following is a color scheme for all five releases.

Release1: is represented with a Green color

Release2: is represented with a Orange color

Release3: is represented with a Grey color

Release4: is also represented with a Grey color

Release5: again Grey color is used to represent it

Postponed: is represented with a Dark Grey color

The figure 2 is depicting a degree of optimality, sum of stakeholder’s preferences for all requirements and total number of requirements assigned to each release in alternative plans. Here, it is important to mention that feature points are referred to total number of points given by a stakeholder to all requirements. The term feature and requirements are inter-changeable and have similar meaning in this study. A plan have higher degree of optimality when values of all parameters are equally considered. A green bar in the figure 2 is showing the plan with higher degree of optimality. Similarly, one can clearly see the more stakeholder’s feature points and number of requirements selected in each alternative. The purpose of such quality evaluation of alternative plans is to determine a best alternative among all.

1. Structure of plan alternatives
[\[view legend\]](#)

ID	Requirement	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
1	Fun1 (Room Records)	1	1	1	1	1
2	Fun2 (Participant Records)	1	5	5	5	4
3	Fun3 (Participant Calendar)	1	4	5	1	4
20	SRE10 (Protect Server Intruders)	1	5	1	1	1
22	SRE4 (Issue key Chair meeting)	1	1	1	1	1
16	SRE6 (Issue key Secretaries)	1	1	1	1	1
17	SRE7 (Return key Secretaries)	1	1	1	1	1
18	SRE5 (Return key Chair meeting)	1	1	4	4	1
19	SRE9 (Entry IT People Server room)	1	1	1	1	1
10	Fun10 (Notify Location)	5	5	5	5	5
21	Fun11 (Record Administrative Staff)	5	5	5	1	1
4	Fun4 (Find Room)	5	5	5	5	5
5	Fun5 (Book Room)	5	5	5	5	5
6	Fun6 (Consider Preferences Booking)	5	5	5	5	5
7	Fun7 (Availability Calendar Booking)	5	5	5	5	5
8	Fun8 (Booked Time)	5	5	5	5	5
9	Fun9 (Booked Time)	5	5	5	5	5
11	SRE1 (Entry Attending Participants)	5	1	1	5	5
12	SRE2 (No Entry Non Attending Participants)	5	1	1	5	5
13	SRE3 (Entry Secretaries AnyRoom)	5	5	5	5	5
23	SRE8 (Lock return key Chair meeting)	5	5	5	4	5

Figure 1: Alternative strategic release plans.

2. Quality evaluation of plan alternatives
[\[view explanation\]](#)

Criteria for Planning	Explanation	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
S/Urgency = 5 Value	Degree of optimality	100.0%	97.0%	96.8%	94.7%	93.7%
= 5 Risk	(Stakeholder feature points)	(24929)	(24184)	(24142)	(23600)	(23348)
= 5 Viability						
= 5 Necessary						
Number of features assigned to...	Release	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
	Release 1	9	8	8	8	8
	Release 2	0	0	0	0	0
	Release 3	0	0	0	0	0
	Release 4	0	1	1	2	2
	Release 5	12	12	12	11	11
	Total	21 of 21	21 of 21	21 of 21	21 of 21	21 of 21

Figure 2: Quality evaluation of alternative release plans.

The figure 3 is showing an excitement level of participating stakeholders for all the planning criteria. The different planning criteria are already mentioned in background 2.2.1. The purpose of such analysis is to predict a reaction of stakeholders about the particular alternative plan. For example, the excitement level of two stakeholders is shown in the figure, each of them have a neutral excitement level for the alternative1, a value of neutral excitement level of each stakeholder is also shown. It indicates that both stakeholders can behave differently on this assignment of requirements in the alternative1. The figure is divided into three columns, where first column is describing an excitement score, the second is explaining value of excitement (adding excitement of all planning criteria) and third column is showing the identity of a stakeholder. The following is an explanation of different excitement scores as described in the `ReleasePlanner`².

Very Excited: Stakeholder is expected to feel very excited about this assignment

Excited: Stakeholder is expected to feel excited about this assignment

Neutral: Stakeholder is expected to feel indifferent about this assignment

Disappointed: Stakeholder is expected to feel disappointed about this assignment

Very Disappointed: Stakeholder is expected to feel very disappointed about this assignment

Surprised: Stakeholder is expected to feel surprised about this assignment

Very Surprised: Stakeholder is expected to feel very surprised about this assignment

N/A Not available. (User did not perform any voting on this feature)

The figure 4 is showing an excitement level of an individual stakeholder for all alternative plans. In this way, an excitement score of each alternative is clearly visible and helps in determining the stakeholder preferences for all alternatives. The meaning of each excitement score is similar as described above.

In the fifth step of "Analyze Solution Set", one can compare total resource consumption in a release with total available resources for a release as shown

²<https://www.releaseplanner.com/>

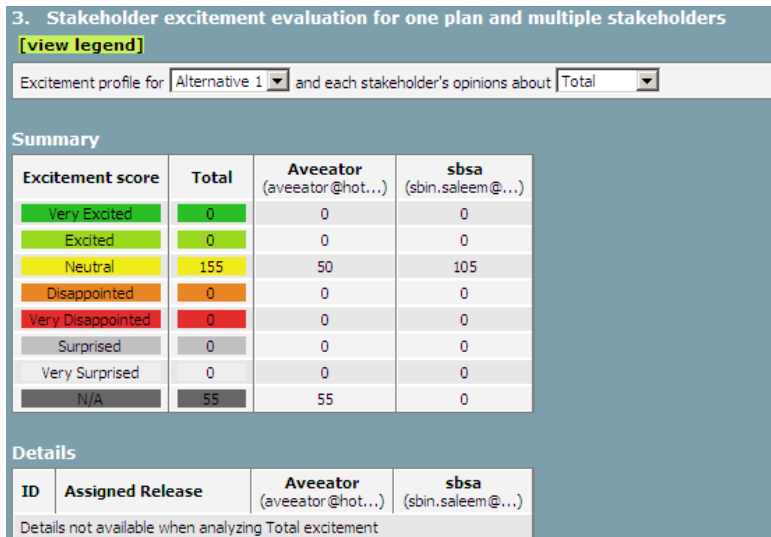


Figure 3: Stakeholder excitement evaluation for one release plan and multiple stakeholders.

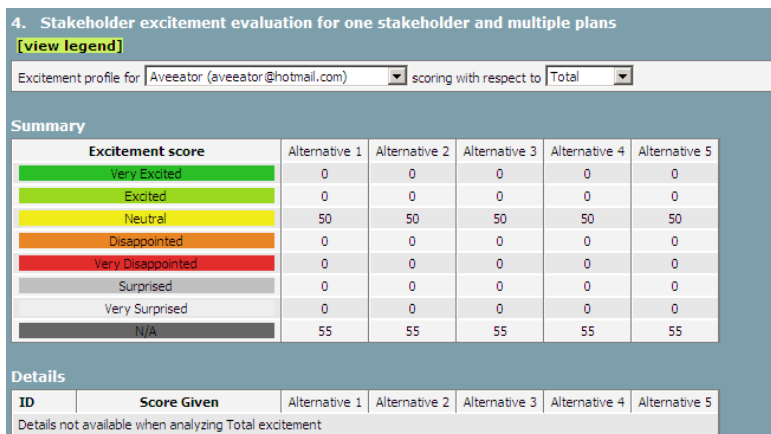


Figure 4: Stakeholder excitement evaluation for one stakeholder and multiple plans.

in the figure 5. The consumption of resources is indicated in a percentage for each release, while a bar is also showing the level of resource usage, the brighter the bar is the higher the rate of consumption. We would like to mention that two resource types "cost" and "time" are used in this study. Therefore, in the figure 5 a resource consumption of both types is shown separately. **ReleasePlanner** provides the facility to even see the resource consumption at requirements level by highlighting the total cost and time allocated to implement each requirement. But such diagrams are not shown in this study for the sake of simplicity.

5. Resource consumption of plan alternatives [view explanation]						
Resource	Release Name	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
Total Development Cost (Includes all phases of Software Development)	Release 1	52.0% 	49.0% 	47.5% 	44.0% 	45.0%
	Release 2	0.0% 	0.0% 	0.0% 	0.0% 	0.0%
	Release 3	0.0% 	0.0% 	0.0% 	0.0% 	0.0%
	Release 4	0.0% 	20.0% 	24.0% 	48.0% 	54.0%
	Release 5	41.1% 	40.1% 	40.4% 	39.1% 	37.9%
	Release 1	100.0% 	94.1% 	91.2% 	86.5% 	84.1%
	Release 2	0.0% 	0.0% 	0.0% 	0.0% 	0.0%
	Release 3	13.3% 	18.9% 	21.7% 	26.1% 	28.3%
	Release 4	100.0% 	100.0% 	100.0% 	100.0% 	100.0%
	Release 5	100.0% 	100.0% 	100.0% 	100.0% 	100.0%
Total Development Time (Includes all phases of Software Development)	Release 1	100.0% 	100.0% 	100.0% 	100.0% 	100.0%
Release 2	0.0% 	0.0% 	0.0% 	0.0% 	0.0% 	
Release 3	13.3% 	18.9% 	21.7% 	26.1% 	28.3% 	
Release 4	100.0% 	100.0% 	100.0% 	100.0% 	100.0% 	
Release 5	100.0% 	100.0% 	100.0% 	100.0% 	100.0% 	

Figure 5: Resource consumption of alternative plans.

The figure 6 structure commonalities between alternative plans is showing a list of requirements which are most commonly selected in all the five alternative plans. The figure is divided into two columns requirement and conformance. A requirement with the most occurrences in all the five alternative plans is represented through a dark green color and the red color is categorizing the least occurrences of a requirement. The conformance of requirements is represented with the following color scheme.

High conformance: Dark Green color

Medium to High conformance: Yellow color

Medium conformance: Orange color

Medium to low conformance: Light Green color

Low conformance: Red color

6. Comparison of structural commonalities between alternative plans		
(view legend)		
ID	Requirement	Conformance %
1	Fun1(Room_Records)	100%
10	Fun10(Notify_Location)	100%
4	Fun4(Find_Room)	100%
5	Fun5(Book_Room)	100%
6	Fun6(Consider_Preferences_Booking)	100%
7	Fun7(Availability_Calendar_Booking)	100%
8	Fun8(Booked_Time)	100%
9	Fun9(Notify_Time)	100%
13	SRE3(Entry_Secretaries_AnyRoom)	100%
22	SRE4(Issue_key_Chair_meeting)	100%
16	SRE6(Issue_key_Secretaries)	100%
17	SRE7(Return_key_Secretaries)	100%
19	SRE9(Entry_IT_People_Server_room)	100%
20	SRE10(Protect_Server_Scheduler)	100%
23	SRE5(Look_return_key_Chair_meeting)	100%
21	Fun11(Record_Administrative_Software)	100%
2	Fun2(Participant_Records)	100%
11	SRE1(Entry_Attending_Participants)	100%
12	SRE2(No_Entry_Non_Attending_Participants)	100%
18	SRE8(Return_key_Chair_meeting)	100%
3	Fun3(Participant_Calendar)	0%

Figure 6: Comparison of structural commonalities between alternative plans.

3.1.2 Other Contributing Factors

The overall resource consumption and stakeholder's voting have a central role in developing a strategic plan. Therefore, the following results of overall resource usage and about analysis of stakeholder's voting are collected from the tool. We would like to mention that for the sake of simplicity, only sample diagrams (not complete) of different stakeholder's voting analysis's steps are presented in this study.

The figure 7 is showing a summary of total available and required resources in terms of cost and time to implement all the requirements. A resource utilization factor indicates the difference between available and required resources. The figure can be divided into two parts, first part in column and rows is showing the given time and cost to implement each requirement, while in the second part all the cost and time resources are summed up to get total resources.

In the **ReleasePlanner** stakeholder's voting is performed based on the five planning criteria, where a stakeholder needs to priorities all requirements based on a planning criteria. The five types of planning criterion are discussed in the background section 2.2.1. Based on this mechanism of voting, the analysis is also performed for each planning criteria. In that analysis an average weighed value of a requirement, prioritization rank in reference

resource usage summary			
ID	Requirement	Total Development Cost (Includes all phases of Software Development)	Total Development Time (Includes all phases of Software Development)
1	Fun1 (Room_Records)	15,000.00	125.00
2	Fun2 (Participant_Secords)	17,000.00	145.00
3	Fun3 (Participant_Calender)	10,000.00	100.00
4	Fun4 (Find_Room)	16,500.00	155.00
5	Fun5 (Book_Room)	22,000.00	250.00
6	Fun6 (Consider_Preferences_Booking)	14,000.00	150.00
7	Fun7 (Availability_Calender_Booking)	16,000.00	165.00
8	Fun8 (Booked_Tag)	8,000.00	70.00
9	Fun9 (Hosts_Time)	7,000.00	80.00
10	Fun10 (Hosts_Location)	7,000.00	90.00
11	SRE1 (Entry_Atending_Participants)	20,000.00	180.00
12	SRE2 (No Entry_Non Attending_Participants)	10,000.00	70.00
13	SRE3 (Entry_Secretaries_AnyRoom)	19,000.00	170.00
16	SRE6 (Issue key_Secretaries)	8,000.00	60.00
17	SRE7 (Return key_Secretaries)	8,000.00	60.00
18	SRE8 (Return key_Chair meeting)	12,000.00	80.00
19	SRE9 (Entry_IT_People_Server room)	11,000.00	90.00
20	SRE10 (Protect Server_Prividers)	9,000.00	55.00
21	Fun11 (Record_Administrative_Stuff)	13,000.00	110.00
22	SRE4 (Issue key_Chair meeting)	14,000.00	135.00
23	SRE5 (Log return key_Chair meeting)	12,000.00	100.00
24	EV1 (Lock_Door)	10,000.00	110.00
25	EV2 (Allow_Access_Valid_Key_Holders)	18,000.00	155.00
26	EV3 (Video_Recording_Important_Assets)	22,000.00	160.00
		Total Development Cost (Includes all phases of Software Development)	Total Development Time (Includes all phases of Software Development)
Total Resource Usage:		318,500.00	2,795.00
Total Available Resources:		860,000.00	3,750.00
Resource Utilization Factor:		37.0 %	74.5 %

Figure 7: Overall resource usage summary.

to other requirements and standard deviation of different stakeholder's priorities for a requirement are provided for a planning criteria as shown in the figure 8. For example, one can see the average weighed value, prioritization rank and standard deviation of of "Urgency" (a planning criteria) in the column "Average urgency", "Rank of the average urgency" and "Standard deviation of the average urgency" of the figure. Here, the higher value of "Average urgency" of a requirement shows the higher importance of a requirement on weighted average value. The meaning of rank 1 of a planning object (requirement) refers to the higher importance than other planning objects and more lower value of rank shows less importance, while less value of standard deviation shows a conformance between stakeholder priorities. Similarly, the overall priority of requirements by adding the values of all planning criteria is also shown in the figure 8. For example, the column "Overall priority" is showing the weighted priority of all planning criteria, the rank of a requirement among all planning criteria is highlighted in the column "Rank of the overall priority", while the column "Standard deviation of the overall priority" gives information about the standard deviation of stakeholder's priorities for all planning criteria. It is consider important to account the weight of each planning criteria for calculation of overall priority as planning criteria can have different weights. For example, "Urgency" can have weight on the scale of 0 to 9, where nine is the highest weight and certainly 0 is the lowest.

The purpose of diagram 9 is to show the most conflicted requirements among all stakeholder's with their overall and according to each planning criteria priority. It has the same level of information as provided in the

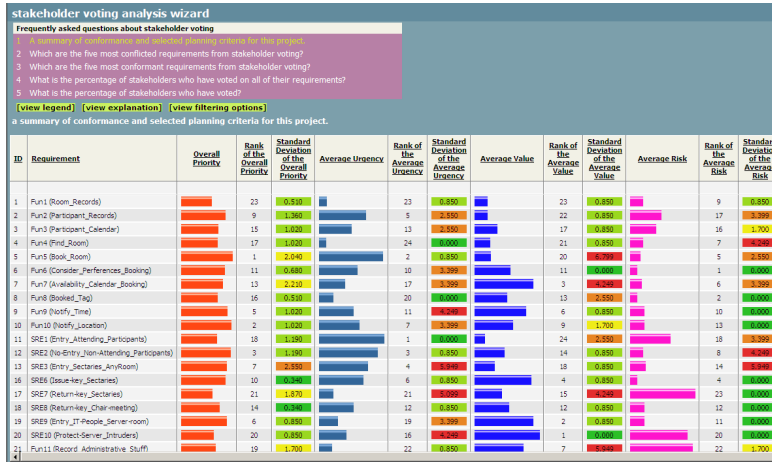


Figure 8: A summary of conformance and selected planning criteria.

above diagram 8.

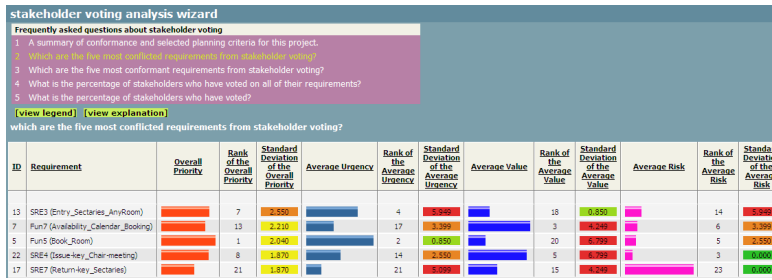


Figure 9: The most conflicted requirements from stakeholder voting.

In this figure 10, the most commonly votes requirements by all stakeholders are shown. Similarly with above diagram 9, it is has same level of information provided in the figure 8.

In this way, it is easier to analyze the impact of stakeholder’s voting on the selection of requirements in subsequent releases. An analysis of results is provided in the following section.

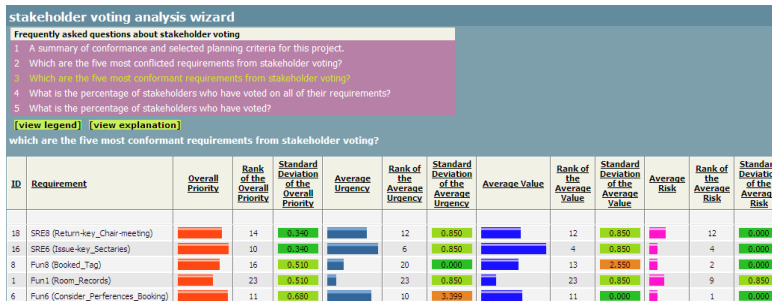


Figure 10: The most conformant requirements from stakeholder voting.

3.2 Data Analysis

3.2.1 Working of the ReleasePlanner

The **ReleasePlanner** automatically generates five different alternatives of a release plan instead of generating one solution. In this way, it is easier for decision-makers to choose an optimal plan of their choice. It not only generates a set of alternative plans but also provides the facility to analyze the results. The tool is based on an optimization algorithm proposed in the strategic release planning model *Evolve** [8], therefore user do not need to do anything for generation of a plan except providing the desire input. In this study, we have provided a set of inputs to **ReleasePlanner** for developing a strategic plan to deliver a meeting scheduler system into five releases. The results in terms of five different alternative plans and analysis of each alternative are gathered in data collection phase. Each analysis step guide a decision-maker understanding any advantages or disadvantages of selecting a plan. In the following paragraph the steps to analyze a generated solution are discussed in detail.

The **ReleasePlanner** has a dedicated section “Analyze solution set” to analyze the generated solution, for further details about different sections in **ReleasePlanner** look at 2.2.1. There are five steps to analyze the generated solution, all the five steps provide a different information from each other. For example, the purpose of step one is to show five different alternative release plans, while a user can compare all alternative plans in step two. The step two provides an information about the most and least alternatives calculated using utility function according to the stakeholder’s expectations. It gives an information about number of requirements assigned to each release as already discussed in the above section 3.1.1. The purpose of third and fourth step is to help decision-makers in knowing the preferences of stakehold-

ers about each alternative plan. In the fifth step, a resource usage summary of each release is presented, while last step shows the conformance of all requirements in different alternatives. Look at the data collection section 3.1.1 for further details about the each step. It is observed that aim of such analysis is to determine the best alternative among all, therefore below we are discussing a possible best alternative for a meeting scheduler system.

The structure plan 1 of alternative1 shows the selection of requirements in first and fifth release only, which means that there is no need of second, third and fourth releases. It also means alternative1 can become more useful for an organization which want to deliver the system into less number of releases. One can see that alternative1 has a higher degree of optimality and high rate of feature points from stakeholders in comparison to all the other alternatives. But, the excitement level of stakeholders for alternative1 is neutral, which may have advantages and disadvantages. Now look at the resource consumption 5, no cost resources are used in second, third and fourth releases but some time resources of release three and four are utilized to adjust the time overhead of the first and fifth release. Such consumption may have advantages and disadvantages depending upon the situation of the organization. In the following paragraphs, we have discussed other parameters which can help decision-makers for selection of an alternative.

The requirements dependencies have a great influence on the assignment of requirements in subsequent releases. Such influence is observed from the results of this study. For example, the Fun5 (a functional requirement) is selected to deliver in the fifth release with its interrelated requirements Fun6 and Fun7 in alternative1. This coupling dependency was originally provided as input to the **ReleasePlanner** as shown in the table 5, this dependency is respected by the tool as shown in figure figure 8. Similarly, SRE1 (a security requirement) is interrelated with SRE2 through a coupling relationship, which is also respected by the tool. The precedence relationship between Fun1 and Fun4 is realized by selection of Fun1 in first and Fun4 in fifth release.

The stakeholder's preferences are one of the key factors in selection of requirements. This fact is observed during this study as one can clearly see the high value of stakeholder's feature points for alternative1 in comparison to other alternatives as shown in the figure 2. The excitement level of stakeholders also favor the alternative1.

The efficient utilization of resources is another goal to achieve from a strategic plan. Now, look at the figure 5 and see that resources are used efficiently in alternative1 comparing to the other alternative plans. But, we can not say that resources are utilized in best possible way in alternative1

as it depends on the organization's situation and nature of the project. It also depends on the constraints and inputs provided to the `ReleasePlanner` for developing a plan. We have observed that in such sequential way of development the even distribution of different resources (allocated to a release plan) is challenging because a resource can be more or less efficiently used compare to the others. For example, one can analyze that cost (a resource) is not utilized at all in the forth release but time resource of forth release is completely utilized to deal with the time overheads of first and fifth release.

3.2.2 Treatment of requirements

In the `ReleasePlanner` all requirements are treated as planning objects and logged into the "Requirements" section of the tool. The requirements can be grouped according to their type like "functional", "security" and "evolving security" but such grouping have no influence on the planning. The only purpose of grouping is to categorize alike requirements together for better understanding. It is observed that all the functional requirements are treated equally but requirements dependencies and stakeholder's priorities make a requirement more important or less important to deliver in a release. Such importance of requirements from the stakeholder's point of view can be analyzed through voting analysis as shown in the figures 9 and 10. Similarly, the resources required to implement a requirement also contribute to define the priority of a requirement but there is no other differentiation is made between functional requirements.

The security requirements are constraints on the functional requirements as defined by Haley et al.[2]. Based on this definition a coupling relationship exists between security and other functional requirements for developing them together in a release. It means both requirements need to develop in the same release as one can see in the figure 1 a security requirement (SRE1) is selected to deliver in the same release with its corresponding functional requirement (FUN5). Here SRE1 is to "protect a booked room", while FUN5 is to "book room". Now the developers needs to construct both requirements together in a release to fulfill the constraint that have some disadvantages. First, the functional requirement is positive constraint of the system, while security is a negative. Such relationship of positive constraints with negative constraints don't explicitly visible in the `ReleasePlanner`, which can mislead a developer. Second, the selection of such related requirements together in a release can affect the selection of other important requirements. Third, developers may need to wait for the development of a functional requirement to develop its corresponding security requirement, which can affect the

utilization of resources. Forth, the development or deliver of both requirements together do not mean that they are functioning together as well. In other words, consider someone have ordered you to develop a door (functional requirement) and lock (security requirement). Now, you have planned to deliver the lock and door together, which does not make any sense for the customer as he has nothing to do with the door and lock as separate entities. Actually, he wants you to develop them in any order but deliver a door with a lock fitting on it for protecting the room or whatever way he want to use the door. Similarly, the treatment of security requirement alike any other requirement is not practical as the attributes protection, asset and harm of a security requirement distinguishes it from any other types of requirements.

The security requirements are dynamic and changes overtime like any other kinds of requirements. Therefore, it is considered important to in-line a strategic plan according to the changes in security requirements. During this study, we have not found any explicit way to deal with evolving security requirements by the **ReleasePlanner**. Therefore, this tool may require a special treatment of evolving security requirements than other kinds of requirements. The reason is that security requirements have different attributes from other requirements and evolvement of one attribute of security requirement can influence a change on security as constraint and security functional requirement (mechanism to realize the constraint).

4 Discussion

ReleasePlanner is an effective tool to generate optimal strategic release plans. It is really useful to balance the different constraints involved in strategic RP. The optimality of a generated solution is totally dependent on the provided input. Therefore, a great care is required while making estimations, analysis of requirements, determining requirement dependencies and stakeholders' votes. It is observed that getting a stakeholder's preferences of requirements is one of the key function of the **ReleasePlanner**.

Stakeholders' voting is one of the important element in the whole release planning process. Therefore a proper knowledge of the tool is very important for stakeholders to perform voting correctly. The most of the participants involved in this case study have complained about complexity of the voting process. Consequently, only two people participated as stakeholders, which is a limitation of this study. However, a complete written guidance to use the tool was provided to all the participants, which is available in appendix 7.2. But, we do want to mention that participants were not pushed for joining the study due to ethical reasons. Therefore, it is recommended to ensure a proper training of stakeholders before inviting them to use the stakeholders' voting section of the tool. It can be interesting to know that how to get a stakeholder's preferences for security requirements as these are different from other requirements. The current criteria of stakeholders' voting in the **ReleasePlanner** can be improved further to make voting process simpler for the stakeholders.

Requirements dependencies are one of the important constraints in the whole release planning process. There is no such support by the **ReleasePlanner** to capture requirement dependencies. Currently, the tool only allows to log the dependencies in its "Requirements dependency" section. But, the correct identification of such dependencies is equally important for developing an effective strategic plan. Especially, the identification of security requirements related dependencies is still unknown. Therefore, any work in such direction to capture the security requirements dependencies will contribute in the development of better strategic plans.

The **ReleasePlanner** treats all the requirements as planning objects and there is no support to deal security differently from the other kinds of requirements. It is found through analysis that there is a need to develop such a mechanism in the **ReleasePlanner** to manage security requirements for strategic release planning. One can argue that requirement dependencies can be used for managing security and functional requirements relationship. But, firstly security is dynamic and completely different from other func-

tional and quality requirements. Secondly, one security relationship can have a deep impact on the security of the whole System. Thirdly, the security requirement has different attributes protection, asset and harm than any other requirement as described earlier 3.2.2. Therefore, it is an interesting research direction to look release planning from the angle of security requirements.

The **ReleasePlanner** does not explicitly deal with evolving security requirements. It categorizes all the requirements equally which means no differentiation between evolving or changing requirement as discussed in the analysis 3.2.2. For example, SRE1 (a security requirement) is delivered in the first release but later the requirement changes completely. Now, a developing organization wants to update the strategic plan, which is impossible in between of the project and for making any change the organization need to wait until the post release analysis. The description of SRE1 can be changed but we are referring about the forms of SRE1 before and after the delivery of a release and its impact on the other requirements. Therefore, the handling of evolving security requirements is another future research direction.

5 Conclusion

We have conducted a case study on the commercial tool **ReleasePlanner**. The motivation of study is to know the treatment of functional, security and evolving security requirements by the tool. The study was designed to collect data through capturing screen-shots of different steps of the tool. One can argue about design of this study as we have not had a preliminary knowledge about working of the tool. But we have used the user manual and notes provided on the site of the tool to mitigate this limitation. A gap to manage security requirements dependency (e.g a relationship between security functional and other functional requirement) after the analysis of results is found. Moreover, we observed that sequential development of such dependencies can cause delay in delivering the software product release. The researcher involved in this study is not a native speaker, therefore, interpretations of the results is also a threat to validity. But, the results was discussed with another scientist to reconfirm the interpretations. It is also a matter of fact that **ReleasePlanner** is a complex tool and require good background knowledge before conducting such study.

6 Acknowledgement

We would like to highly acknowledge Dr.Charles Haley for his support at the early stage of this research. Moreover, we cannot forget the generosity of Dr.Guenther Ruhe and the team of Expert Decisions Inc.³ for providing us access of their commercial tool `ReleasePlanner`.

³<https://www.releaseplanner.com/>

References

- [1] D. Greer and G. Ruhe. Software release planning: an evolutionary and iterative approach. *Information and Software Technology*, 46(4):243–253, 2004.
- [2] C.B. Haley, R. Laney, J.D. Moffett, and B. Nuseibeh. Security requirements engineering: A framework for representation and analysis. *Software Engineering, IEEE Transactions on*, 34(1):133–153, 2008.
- [3] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett. Introducing abuse frames for analysing security requirements. In *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International*, pages 371–372, 2003.
- [4] Daniel Mellado, Carlos Blanco, Luis E. Sanchez-nchez, and Eduardo Fernandez-Medina. A systematic review of security requirements engineering. *Computer Standards & Interfaces*, 32(4):153–165, June 2010.
- [5] T. Mens, M. Wermelinger, S. Ducasse, S. Demeyer, R. Hirschfeld, and M. Jazayeri. Challenges in software evolution. In *Principles of Software Evolution, Eighth International Workshop on*, pages 13–22, 2005.
- [6] Common Criteria Sponsoring Organizations. Common criteria for information technology security evaluation part1: Introduction and general model, version 3.1, rev 3. Institute of Standards and Technology, CCMB-2009-07-001, 2009.
- [7] G. Rugg and M. Petre. *A gentle guide to research methods*. Open Univ Pr, 2007.
- [8] Guenther Ruhe and An Ngo. Hybrid intelligence in software release planning. *Int. J. Hybrid Intell. Syst.*, 1(1-2):99–110, 2004.
- [9] Omolade Saliu and Guenther Ruhe. Supporting software release planning decisions for evolving systems. In *Proceedings of the 29th Annual IEEE/NASA on Software Engineering Workshop*, pages 14–26. IEEE Computer Society, 2005.
- [10] Mikael Svahnberg, Tony Gorschek, Robert Feldt, Richard Torkar, Saad Bin Saleem, and Muhammad Usman Shafique. A systematic review on strategic release planning models. *Information and Software Technology*, 52(3):237–248, March 2010.

- [11] Thein Than Tun, Yijun Yu, Charles Haley, and Bashar Nuseibeh. Model-Based argument analysis for evolving security requirements. In *2010 Fourth International Conference on Secure Software Integration and Reliability Improvement*, pages 88–97, Singapore, Singapore, 2010.
- [12] Axel van Lamsweerde. Elaborating security requirements by construction of intentional Anti-Models. In *Proceedings of the 26th International Conference on Software Engineering*, pages 148–157. IEEE Computer Society, 2004.

7 Appendix

In this section, a preliminary data used at different stages of this case study is provided.

7.1 Case Study Input

This section contains all the inputs provided to **ReleasePlanner** for developing a strategic release plan. At the first stage, a list of customer demands was prepared. Then each customer demand was analyzed until it becomes a unique testable requirement. The functional, security and evolving security requirements are gathered from a set of unique testable requirements and grouped into functional, security and evolving security requirements. In next stage, a cost and time resource to implement each requirement are estimated and logged into the table. Similarly, the dependency relationship between all requirements are captured through human judgement.

7.1.1 Customer Demands

The following are demands by the customer to develop a meeting scheduler system.

- A meeting proposed shall be able to find a room based on FIFO principle
- All participants' calendar/availability and preferences are collected into a database
- All participants are notified (by email) about the meeting time/location
- The meeting room doors should be accessible by the attendants at given time
- The meeting room doors should not be accessible by the non-attendants at given time

7.1.2 Requirements

In this section, the natural language description of functional, security and evolving security requirements are provided.

The table 1 describes a set of functional requirements to develop a meeting scheduler system. A unique requirement's identifier is provided with each requirement.

Identifier	Functional Requirement
Fun1	The system shall keep record of all meeting rooms in JLB.
Fun2	The system shall keep record of all participants' preferences for meeting (means time).
Fun3	The system shall have calendar information of all participants.
Fun4	The system shall find a room for any proposal of meeting by any participant.
Fun5	The system shall book an available room for the proposed meeting based on the FIFO principle.
Fun6	The system shall consider each participant's preferences before booking of meeting.
Fun7	The system shall consider availability of all participants based on their calendar information before booking of meeting.
Fun8	The system shall tag "booked" after booking a room for the proposed meeting.
Fun9	The system shall notify about the time of meeting to each attending participant through email.
Fun10	The system shall notify about the location (room) of meeting to each attending participant through email.
Fun11	The system shall maintain the records of administrative stuff that includes secretaries and IT people of the building or department.

Table 1: Functional requirements

The table 2 explains a set of security requirements of a meeting scheduler system.

The list of evolving security requirements are provided in the table 3.

7.1.3 Organizational Resource Estimations

There are certain resources required for developing a system, therefore it is very important for an organization to determine what type of resources required to develop a certain system. In this study, we have assumed that only cost and time are two types of resources required for the development of meeting scheduler system. In the next step, it is important for an organiza-

Identifier	Security Requirement
SRE1	The system shall provide access of booked meeting room to attending participants of the meeting.
SRE2	The system shall not provide access of booked meeting room to non-attending participants of the meeting.
SRE3	The system shall provide access of any meeting rooms to sectaries through a valid key.
SRE4	The system shall have a mechanism to issue valid key of the room to chair of the meeting.
SRE5	The system shall have a mechanism to log return of valid key of the room from chair of the meeting.
SRE6	The system shall allow only sectaries to issue key of the meeting room.
SRE7	The system shall allow only sectaries to log key return of the meeting room.
SRE8	The system shall have a mechanism to insure that chair of meeting should return key of the meeting room to sectaries.
SRE9	The system shall allow only IT people to enter in the server room through a valid key.
SRE10	The system shall have a mechanism to protect servers from other than IT staff of building.

Table 2: Security requirements

tion to estimate the total available resources and allocate a certain amount of each resource to a release. In this study, we have planned to deliver the meeting scheduler system into five releases. Therefore, the following table 7 is presenting an amount of cost and time allocated to each release.

7.1.4 Requirements Estimations

There are certain resources required to implement each requirement. A list of cost and time required to develop requirements of a meeting scheduler system are provided in this section. The coupling and precedence relationship between requirements are also described in this section.

The table 5 is showing the cost and time implement functional requirements with a coupling and precedence relationship of a requirement with the other requirements.

Identifier	Evolving Security Requirement
EV1	The system shall keep lock the doors all time.
EV2	The system shall allow access in the building only to valid key holder of all the doors.
EV3	The system shall have a mechanism of video recording of important assets of building.

Table 3: Evolving security requirements

Release	Total Available Cost	Total Available Time
Release1	£200000	850hours
Release2	£100000	600hours
Release3	£110000	900hours
Release4	£50000	400hours
Release5	£400000	1000hours

Table 4: Total organizational resource estimations in terms of cost and time

In the table 6, a cost, time, coupling and precedence information of security requirements are provided.

The table 7 contain a cost, time to implement evolving security requirements. There was no coupling and precedence relationship was found.

7.2 Instructions for Stakeholders to Use Prioritization Section of the ReleasePlanner

The following list was provided to the stakeholders (those have participated in this study) as a guidance for using the stakeholders' prioritization area of theReleasePlanner.

- You will see update user information page at the first place, if you want to change any of the personal details then you can fill the form and click button "Update user information" otherwise click button "Skip".
- You will reach in the main page titled "Existing projects" after clicking any of the above buttons. Here you will see project name "Meeting scheduler JLB". Please click on the project name and you are in the page titled "Prioritization". Now you will see two types of tabs named "Criteria" and "Groups". The criteria's are "Urgency", "value", "Risk", "Volatility" and "Necessary" which will be used for priority of

Identifier	Cost	Time	Precedence	Coupling
Fun1	£15000	125hours	Fun4	No
Fun2	£17000	145hours	Fun6	No
Fun3	£1000	100hours	Fun7	No
Fun4	£16500	155hours	No	No
Fun5	£22000	250hours	No	Fun6 and Fun7
Fun6	£14000	150hours	No	Fun5
Fun7	£16000	165hours	No	Fun5
Fun8	£8000	70hours	Fun9 and Fun10	No
Fun9	£7000	50hours	No	Fun10
Fun10	£7000	50hours	No	Fun9
Fun11	£13000	110hours	No	No

Table 5: Functional requirements' estimations

certain requirement in two “Groups” of requirement “Functional requirements” and “Security requirement”.

- By clicking on any of the above tab like “Urgency” and then clicking green button “View criterion Description” you will find the purpose of that particular tab. Similarly, by clicking the green button “View requirements details” and then clicking any requirement from the “List” on your left side you can see description of that requirement.
- Now chose any type of priority by clicking any tab and click your choice of requirements group either “Functional requirements” or “Security requirements” for voting.
- For example, I clicked on the “Urgency” tab and then “Security requirements” tab. Below, I can see list of “Requirements” on my left side and nine relative values from “Extremely low” to “Extremely high”. Now I thought that “SRE6” requirement seems in between “Low” to “Average” level of urgency for me. Therefore, I just pick that requirement from the “List” and drop it in the box “4”. I can see the message “Vote saved”. It’s done.
- Then, I performed the same procedure on all set of requirements until my requirement list is empty.
- At-last, I just clicked the button “Submit priorities” to submit my vote for all set of requirement in one group “Security requirements” for one level of priority like “Urgency”

Identifier	Cost	Time	Precedence	Coupling
SRE1	£20000	180hours	SRE4	SRE2
SRE2	£10000	70hours	No	No
SRE3	£19000	170hours	No	No
SRE4	£14000	135hours	SRE5 and SRE8	No
SRE5	£12000	100hours	No	No
SRE6	£8000	60hours	No	SRE8
SRE7	£8000	60hours	No	SRE5 and SRE8
SRE8	£12000	80hours	No	No
SRE9	£11000	90hours	SRE10	No
SRE10	£9000	55hours	No	No

Table 6: Security requirements' estimations

Identifier	Cost	Time	Precedence	Coupling
EV1	£10000	110hours	No	No
EV2	£18000	155hours	No	No
EV3	£22000	160hours	No	No

Table 7: Evolving security requirements estimations

- It is not necessary that you need to prioritize all groups of requirements or all priority levels like "Urgency". Rather, it is your own choice that you feel what is important.