Vision Backlog

Master's Thesis

Nitish Patkar

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science

Thesis Supervisors: Dr. Simon Oberthür and Dr. Stefan Sauer



Department of Computer Science Paderborn University Germany November 20, 2017

Declaration

(Translation from German)

I hereby declare that I prepared this thesis entirely on my own and have not used outside sources without declaration in the text. Any concepts or quotations applicable to these sources are clearly attributed to them. This thesis has not been submitted in the same or substantially similar version, not even in part, to any other authority for grading and has not been published elsewhere.

Paderborn, November 20, 2017

City, Date

Signature

Vision Backlog

Abstract

Requirements elicitation plays a vital role in building a successful software. Incorrect or incomplete requirements lead to an erroneous software and costs a huge amount of rework. Rework costs in terms of money and efforts are usually higher than the early detection of potential flaws in the requirements. This happens because most of the techniques employed to extract requirements fail to understand end user goals. Understanding your users and their goals is important to build a capable, viable and a desirable product or a software system. This thesis attempts to suggest and evaluate an alternative approach to understand your potential users and their goals so that correct and complete requirements can be formulated resulting in a successful software.

Contents

1	Intr	oduction	13			
	1.1	Motivation	14			
	1.2	Objective	15			
2	Fou	undation				
	2.1	Requirements elicitation techniques and approaches	17			
		2.1.1 Interview	18			
		2.1.2 Questionnaire	18			
		2.1.3 Task analysis	18			
		2.1.4 Domain analysis	18			
		2.1.5 Observation	19			
		2.1.6 Protocol analysis	19			
		2.1.7 Prototyping	19			
		2.1.8 Brainstorming	19			
		2.1.9 Card sorting	19			
		2.1.10 Joint Application Development (JAD)	20			
		2.1.11 Scenarios	20			
		2.1.12 Viewpoints	20			
		2.1.13 SWOT analysis	20			
		2.1.14 Theory of change	20			
		2.1.15 Problem definition	20			
		2.1.16 Repertory grids	21			
		2.1.17 Laddering	21			
		2.1.18 Literature review	21			
		2.1.19 Persona	21			
	2.2	Theory of Diffusion of Innovation and User types	21			
	2.3	Goal directed design process	23			
	2.4	Behavior variables and patterns	25			
	2.5	User goals	26			
	2.6	Context	28			
	2.7	Persona	30			
	2.8	Question types	32			
	2.9	Summary	33			
3	Solı	ition	35			
-	3.1	Introduction	35			
	3.2	Building the concept	36			
		3.2.1 Classification of elicitation techniques	36			

	3. 3. 3.3 B ³ 3. 3. 3. 3. 3. 3. 3. 3. 3. 3.	2.2Shortlist and purpose of respective techniques382.3Using shortlisted techniques in Vision backlog392.4Set of questions40uilding the software413.1Question categories and corresponding user interface (UI) el- ements423.2Brief technical introduction443.3Use case diagrams453.4Class diagram473.5EER diagram483.6Frameworks and application architecture503.7A walk through the running application56ummary63
4	Usage 4.1 In 4.2 In 4.3 Ho 4.4 Ho 4.5 Ho 4.6 Su Evalua	Guidelines65troduction65terpretation of the received answers65to identify user types69to be behavior variables can help69to build Personas71ummary73tion75traduction75
6	5.1 In 5.2 E ⁵ 5.3 Q 5.4 E ⁵ 5.5 Su	troduction 75 valuation setup 75 nestionnaire 76 valuation results 77 immary 83 usion 85
7	Future	work 87
	Tutule	
Α	Interv A.1 In A A	ew questions91terview questions911.1 Usability survey911.2 Survey on content quality92
В	Actual B.1 Sa	application screenshots93mple screenshots of the application93

List of Tables

3.1	UI element choices - part 1
3.2	UI element choices - part 2
4.1	User attributes and supporting percentage
4.2	Behavior pattern

List of Figures

2.1	Six phases of goal directed design process [11]	24
2.2	System and context boundary [47]	29
2.3	Why persona is important: different people having different needs [11]	30
2.4	A sample blank Persona explaining user details	32
3.1	Classification criterion for elicitation techniques studied for Vision	
	backlog	38
3.2	Shortlisted elicitation techniques and their purpose in Vision backlog	39
3.3	Actual set of questions - part 1	40
3.4	Actual set of questions - part 2	41
3.5	Vision backlog use case diagram	46
3.6	Vision analytics use case diagram	47
3.7	Class diagram	48
3.8	EER diagram	50
3.9	Full web stack options $[60]$	51
3.10	Simplified overview of the application architecture	52
3.11	Sample model	53
3.12	Sample controller	54
3.13	Sample repository	54
3.14	Sample swagger API	55
3.15	Sample angular service	55
3.16	Sample angular template	56
3.17	Stakeholder profile screen mock up	57
3.18	Actual stakeholder profile screen	58
3.19	Tasks list screen	59
3.20	Actual stakeholder tasks list	59
3.21	Analyst task list mockup	60
3.22	Actual analyst task list screen	60
3.23	Analyst feature list screen	61
3.24	Analyst feature list screen actual	61
3.25	Mockup behavior variables	62
3.26	Actual behavior variable screen	63
4.1	Questions and interpretations - part 1	66
4.2	Questions and interpretations - part 2	67
4.3	Questions and interpretations - part 3	68
4.4	Example behavior variables	71
4.5	Example Persona with filled in details	72

5.1	Set of question - 1
5.2	Set of question for obtaining content quality - 1
5.3	Usability evaluation- Portfolio of results
5.4	Usability evaluation- Diagram of average values
5.5	Usability evaluation- Description of word pairs
5.6	$Usefulness \ evaluation -1 \ \ldots \ \ldots \ \ldots \ \ldots \ 81$
5.7	Usefulness evaluation-2
5.8	Usefulness evaluation-3
5.9	Usefulness evaluation-4
5.10	Usefulness evaluation-5
7.1	Future work
A.1	Set of question - 2
A.2	Set of question - 3
A.3	Set of question for obtaining content quality - 2
A.4	Set of question for obtaining content quality - 3
A.5	Set of question for obtaining content quality - 4
B.1	Stakeholder's view - Profile
B.2	Stakeholder's view - Task creation
B.3	Stakeholder's view - Task list
B.4	Analyst's view Help
	Analyst s view - help

Chapter 1 Introduction

Undoubtedly requirements and especially the elicitation process plays a critical part in the successful software development [4]. If wrong requirements are elicited or critical requirements have not been included, then it can lead to an erroneous software system and cause rework [39]. The rework costs a lot of money and affects project schedule exponentially [39] [58] [57]. Hence, it is important that we elicit correct and complete requirements in the first place. Another great challenge that software industry faces is that the systems tend to be extremely complex these days, with increased number of stakeholders, have a shorter release cycles to evolve and need to adapt to the environment and the users [35] [4].

However it is not always easier to extract requirements for softwares. For example think of the following situation: An assisting software system needs to be built for skilled employees who assemble different products in a smart industry (Industry 4.0). Products to be assembled keep changing one after another, in contrast to traditional routine where same products are repeated over and over. In such a situation, the skilled employee has to exactly remember different assemblies and he must be careful not to make any mistake. An assisting software system will help these employees by making all the required information about the product to be assembled at his fingertips.

In a conventional requirements elicitation setup, requirements for such an assisting software system will be dictated to requirement engineer in a formal meeting. Meetings are always a tough choice as employees are relatively busy resources and are mostly naive about elicitation process. Meetings also face issues like people not being able to best express themselves or do not have enough time [8] [36]. Also, it is totally possible that while working on the assembly line they *remember* or *realize* a very small but fundamental point which could potentially be a new requirement or which can refine an existing one. If not *somehow* noted right away, it is very much possible that they will perhaps forget about it at the end of the day.

One step further, if they can refine their notes/points later to include more contextual information or insights about them, this whole information can help requirements engineer to extract exact needs and expectations or their goals and can also contribute directly to forming actual requirements. When this entire process is done without having to meet requirements engineer very often, we can assume that employees can save a lot of their efforts and time.

The above mentioned situation is a very typical one and instead of a skilled employee, it can be anyone who is an expert in his field seeking a software solution. It is hard in such situations for requirements engineer to extract *needs* or *expectations* or *user goals* from other stakeholders just by observation and especially when there is a limited time for formal meetings.

This thesis concentrates on such specific situations and attempt to precisely formulate the above mentioned problem (see section 1.1) with following key contributions-

- Classification of elicitation techniques as per their usability for the stakeholders to locate and scope problems.
- An approach to enable stakeholders to do elicitation themselves by guiding them through the elicitation process on behalf of the requirements engineer. To accomplish this, develop a concept, which henceforth will be called *Vision backlog* for stakeholders to enable them to independently state a *core goal/problem* which they think should be *tackled*. Part of the concept will be *Vision analytics* which will help analysts to explore stakeholder goals.
- Software implementation of this concept to show that it is technically feasible.
- An evaluation of this conceptual system to show evidences that requirements engineer is going to benefit from it as some of his elicitation work will be taken away. Also to show that when stakeholders concentrate more on the actual problem instead of solutions and do the elicitation themselves, it becomes simpler to extract their actual needs so that a closely fitting solution can be built to solve their problem.

1.1 Motivation

Stakeholder's contribution in forming requirements for a software is vital and it is now widely being recognized [65] [63]. The reason is, usually the stakeholders have the clearest idea about their own needs and expectations from the perceived system. They posses the *tacit knowledge* and have the better understanding of the problem domain. On the other hand, requirements engineer usually has a very little of it [13]. That's why it makes more sense that stakeholders themselves describe their ideas, needs, expectations so that quality requirements can be educed out of it.

Having identified the importance of quality requirements, people are now trying innovative techniques over the conventional ones [12] [18] [63] [37] for efficiency. The goal of such techniques is to come up with correct, complete, and diverse requirements. *Crowdsourcing* is one such example. Such an approach essentially motivates more and more stakeholders to contribute their ideas and brings in a lot of brainstorming about possible solutions. As a result, developers can build a closely fitting system with a hope of less rework [30]. Considering the above outlined situation, technique like *crowdsourcing* is of little help. Such a situation has challenges of it's own.

- Naive stakeholders have a very little knowledge of the elicitation process, so either they have to learn it or have to keep depending on the requirements engineer. These both solutions are not so practical considering stringent time constraints.
- Elicitation requires special skills and among all the stakeholders only requirements engineer has those [63]. If stakeholders have to depend on him for elicitation all the time, he can become a bottleneck. Furthermore, in small organizations he is a relatively busy resource, and can not be shared among multiple teams.
- Naive stakeholders usually focus more on the probable solution rather than the actual problem they are facing. Most often they are not even aware of the actual problem. Identifying the core of the stakeholder's problem can lead to quality requirements [8].

In addition to these high level challenges elicitation techniques face more concrete limitations-

- 1. Notes, lists, sketches which are mostly used to record needs or expectations of stakeholders are not efficient ways, as they can not naturally be tied to actual requirements [63].
- 2. Formal meetings are not feasible enough to thoroughly extract stakeholder's needs, expectations etc. [63] Stakeholders can get *insights* about their needs literally any time, especially when they are working on it. It could be hard during formal meetings to point out or recollect specific things [8] [5] [37].

1.2 Objective

Conceptualize and build an expert system i.e. Vision Backlog that will:

- capture stakeholder's exact needs and expectations in the form of goals with associated context
- always be available so that stakeholders can benefit from it any time
- leverage from concepts like Personas, SWOT analysis etc. from different domains to improve current requirements elicitation methodology [53]

With a focus that:

- captured knowledge/information through this system will help requirements engineer to extract tacit knowledge and to form more accurate and complete requirements
- all the stakeholders can actively participate in understanding the underlying problem to collectively reach a closely fitting solution

Chapter 2

Foundation

Vision backlog benefits from many concepts from different disciplines and combines them to improve current requirement elicitation process. These concepts help to understand end user goals, relevant contexts, and user's personal attributes. Following sections 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 and 2.8 explain these concepts in detail.

2.1 Requirements elicitation techniques and approaches

The requirements elicitation process represents an early but continuous and critical stage in the development of software systems. It is concerned with learning and understanding end user, customer and/or stakeholder needs and ultimately communicates these needs to the system developers. It achieves its purpose through various activities that involve communication, prioritization, negotiation and collaboration with all the concerned stakeholders [65] [12]. It is requirements analyst's job to carry out the requirements elicitation process by exploring the problem domain, gathering requirements and by prioritizing and negotiating those requirements with other stakeholders. He has to guide and assist the stakeholders or users through the elicitation process is a quite complex one and involves many techniques, approaches and tools most of which are extensively communicative in nature. Many of these techniques are borrowed from different disciplines like psychology or social sciences.

The technique or a set of techniques those are used during the elicitation process primarily decide the quality of requirements (i.e. their consistency, completeness, correctness and relevance). Requirements analysts with extensive experience tend to have ability to choose such appropriate techniques for elicitation. However, in reality most of the requirements analysts working in the industry are less experienced who fail to do the elicitation correctly by employing wrong techniques thus ultimately resulting into poor software products those do not meet end user needs [57]. However, selecting effective elicitation technique alone does not guarantee that we will elicit quality requirements. If a good technique is used poorly then we will end up having poor results as well [27]. The Chaos Report [57] also points out that user involvement into elicitation process is also and rather as important (nearly 16%) reason for software project failure as wrong or incomplete requirements (which is nearly 13%) are.

This urges us to look at the ways how successful analysts are using elicitation techniques that uncover user needs and to encourage user involvement in the elicitation process as argued in 1.1 by partially shifting the work of elicitation from analysts to the users or stakeholders. Following techniques and approaches were studied which are extensively used either independently or in combination by analysts during the elicitation process with the aim of forming quality requirements.

2.1.1 Interview

Interview is an old and widely used elicitation technique that focuses on collecting large amount of data [10]. It is used to collect data about end user goals (see 2.5), it's surrounding contexts (see 2.6), behaviors (see 2.4) etc. The effectiveness of this technique depends heavily on the skills of the interviewer. There are essentially three types of interviews. *Structured*, *Unstructured* and *Semi-structured*. If the interview is conducted with predefined set of questions and a concrete agenda, then it is called structured interview. On the other hand, if neither the content nor the sequence of the topics to be discussed is predetermined, then it is called unstructured interview [29]. Experience shows that structured interviews are more effective than unstructured interviews [10]. Semi structured are combination of both.

2.1.2 Questionnaire

Questionnaires like interviews are also used during initial stages of elicitation and they also focus on collecting information from different stakeholders [17] [65]. To be proven effective as an elicitation technique, it is important to design questions which avoid gathering redundant and irrelevant information. It is hence important for the designer of such questionnaires to understand concepts and boundaries of the problem domain (see 2.6 for more details). Typically they are seen as informal checklists that ensures that all the fundamental points are addressed early that forms the foundation for future elicitation activities.

2.1.3 Task analysis

Task analysis decomposes high level tasks into smaller subtasks until all actions and events performed to carry out those tasks are uncovered [65] [9] [61]. It gives information on the interactions between the users and the system pertaining to a specific contexts. Usually task analysis requires thorough investigation to uncover such details and it is required to define a level of details till which things can be further explored.

2.1.4 Domain analysis

Domain analysis approaches uncover domain knowledge by examining existing product manuals, design documents etc. Domain analysis is usually done in conjunction with other techniques like interview or *Observation*.

2.1.5 Observation

Observation is also a widely used technique to uncover tacit knowledge which usually stakeholders or users can not express easily. Analysts in such cases observes them while performing different activities or while executing a process. This technique is expensive to perform and requires significant efforts from analyst to interpret the actions being performed [65]. This technique is also used in conjunction with other techniques like interview or task analysis.

2.1.6 Protocol analysis

In protocol analysis participants perform an activity and simultaneously describe aloud what and why they are doing [22] [65]. This technique helps analysts to capture tacit knowledge e.g. about the purpose behind any process of the system which stakeholders or users can not express easily.

2.1.7 Prototyping

Prototyping is one of those approaches that is widely used to reduce the rework costs on software [52], typically in early stages. In this approach the system with its different features is presented with different types of prototypes to the customer. The customers can play around with those prototypes and subsequently refine their requirements before committing those requests to production which is costlier than ideation. The purpose of prototypes could also be to explain or explore important concepts or try out different designs etc. There are different types of prototypes like *storyboards, executable* or *throw-away* prototypes which can be used to represent the final system. This approach is also used in conjunction with other elicitation techniques like interview.

2.1.8 Brainstorming

Brainstorming is used in the initial phase of elicitation process to help form a preliminary mission statement of the project. Stakeholders from different groups are encouraged to generate as many ideas as they can think of. The goal is not to explore details of those ideas or critic them but to encourage freethinking to discover new and innovative solutions to existing problems [44] [65].

2.1.9 Card sorting

Card sorting helps to understand and define relation between different domain entities. In this method the stakeholders sort cards having domain entity names into logical groups according to their understanding. They also then need to explain why they grouped those objects like that. Complete understanding of the problem domain and a complete list of domain entities is thus prerequisite for this technique to be effective [65].

2.1.10 Joint Application Development (JAD)

JAD is typically performed after brainstorming as the system goals are already established. All the stakeholders discuss the problems to be solved and different solutions to that problem. Decisions are expected to be made faster as all the concerning stakeholders are present for negotiation [62].

2.1.11 Scenarios

Scenarios describe current and future processes that comprise of actions and interactions between a user and the system. Scenarios are not concerned with internal details of the system rather this technique discovers them during iterations that adds details. All the possible exceptions are considered which would occur during each step [65].

2.1.12 Viewpoints

Viewpoint approaches let us model the problem domain through different perspectives. It aims at developing complete and consistent description of the target system. It is particularly useful for projects with complex entity relationships. These approaches need significant amount of efforts [65].

2.1.13 SWOT analysis

SWOT stands for Strengths, Weaknesses, Opportunities, Threats. This framework lets organizations focus on the most important issues which are responsible for organization's growth and development. It is a collection of internal and external factors which have or may have impact on business. Strengths or weaknesses are always with reference to the competition. It is a fairly simple tool but if used seriously can give insights about probable improvements of existing processes etc. [45]

2.1.14 Theory of change

Theory of change is a rigorous interactive activity where participants or stakeholders together articulate their long-term goals, identifying the necessary conditions which must be met to achieve those goals. These conditions are nothing but *outcomes* of specific actions which measured against success *indicators*. This tool helps the stakeholders to connect their work to a bigger goal and helps them spot potential risks in their plan by making them talk about underlying assumptions. Assumptions are the underlying context for the problem to be solved [55] [14].

2.1.15 Problem definition

Problem definition helps stakeholders to explore a specific problem from different angles with associated contexts and issues. It is a narrow down approach that makes stakeholders think through complex interconnected issues to get to the key issue that can be improved. Participants are encouraged to internalize on the problems and provides them space to rethink those problems in ways (which could be new or controversial) which could potentially change them [34] [14].

2.1.16 Repertory grids

Repertory grid is an interview technique to explore ways in which a person constructs his experience. In this technique stakeholders are asked to construct their imagination of a problem domain through a set of attributes for that domain. Values are then attached to such identified attributes. The result is a model of a system in the form of a matrix that categorizes domain entities into different categories. The goal behind categorizing entities is to explore similarities and differences between those entities. Because of its complexity, it is mostly used by domain experts [33] [19].

2.1.17 Laddering

Laddering is essentially a one-to-one in-depth interview technique where interviewee identifies different product attributes and later arranges them according to their significance to him to form a ladder. The interviewer asks "*Why is this attribute important?*" until all the aspects of the ladder are explored [28] [59]. This technique is based on a theory that users choose a product because it contains attributes which help them in achieving something while fulfilling their values [25].

2.1.18 Literature review

Literature review is typically done at the beginning of the elicitation process to illuminate the problem at the hand by finding out what others have already thought or done about it. This helps to formulate a clear idea about the problem we are trying to solve [23]. Literature review helps analysts to understand domain better.

2.1.19 Persona

Personas are essentially portraits of fictional but realistic people which are used as common reference point by all the stakeholders in the team to focus their design and development efforts for [14]. By building a Persona for a specific group of users/people that exhibit similar characteristics, a common understanding is developed within stakeholders to favor and prioritize decisions. This technique is described in detail in following section 2.7.

Subsection 3.2.1 proposes a classification criteria and classifies the above elicitation techniques so that a scheme could be proposed to increase stakeholder participation in elicitation by making them use some of these techniques to locate and scope their own problems, see subsections 3.2.2 3.2.3.

2.2 Theory of Diffusion of Innovation and User types

The theory of *Diffusion of Innovation* proposed by Rogers [49] tells us that any innovation is adopted by its users (i.e. diffusion of innovation) in a specific sequence. Some users adopt it earlier as compared to others and the adoption process continues until a critical mass i.e. your entire target audience has adopted it. Factors like

uniqueness of the innovation, communication channels available and social system at that time, decide to which extent the innovation spreads among its users. It sustains only if it is widely adopted. Users judge an innovation on its relative advantages over existing tools or procedures. Compatibility with existing systems, complexity, testability are few characteristics which users like to evaluate before adopting it. Ability and motivation also play an important role for users to adopt an innovation.

To prove this, Rogers points out specific observations or characteristics of the users. These characteristics help us to distinguish target users into different categories called *adopter categories*. The theory suggests that, diffusion of innovation among adopter categories happens in a sequence marked by distinct stages [41] [32]. Rogers suggests five stages of this adoption process:

- 1. **Knowledge**: during this stage, the individual hears for the first time about an innovation and he has very little information about it. He is not enough motivated to know more about this innovation.
- 2. **Persuasion**: during this stage the individual has gained some interest in the innovation, and as a result he actively seeks more information about it.
- 3. **Decision**: during this stage the individual explores advantages and disadvantages of the innovation and decides to either accept it or reject it.
- 4. **Implementation**: once the user accepts the innovation, he uses it to a varying degree. During this stage the individual also determines the usefulness of the innovation and may search for further information about it.
- 5. **Confirmation**: during this stage the individual decides to continue using the innovation.

along with following five adopter categories:

- 1. **Innovators** are those people who are eager to try new ideas, are usually daring and tend to take risks. They also can bear financial setback. Innovators play an important role in launching a new idea into a social system.
- 2. Early adopters are those people with individual opinions and they are seen as role models in their social system. They play an important role to decrease uncertainty about a new idea by adopting it, and spread it within their interpersonal network.
- 3. Early majority are the people to adopt a new idea just before reaching out to the most common people in their social system. They take more time to take favorable decision about a new idea than early adopters.
- 4. Late majority are the people in a specific social system adopt new changes as a necessity or because of increasing social pressure.
- 5. Laggard are the last people in a specific social system to adopt an innovation. Laggards are suspicious of innovations and usually feel comfortable with traditional values and the past.

The diffusion of innovation is essentially a decision making process [51] and interestingly an individual can decide not to use an innovation at any time during the process. Rogers also points out that, innovations that are less risky are adopted faster. On the other hand, if the innovation appear to be obstacle into routine tasks, it is not adopted easily in spite of having high relative advantages.

Adoption of any new idea is a result of human interaction through interpersonal network. In the context of Vision backlog, the stakeholders belong to such a specific social system. Identifying individual stakeholders using *Vision backlog* software to be of specific adopter category, can help analysts to identify and focus innovative ideas or improvements. This is explained in detail in subsection 4.3.

2.3 Goal directed design process

Goal directed design process keeps end user or stakeholder goals at the center of requirements elicitation. It argues that most digital products fail because they fail to understand end user goals [11].

Digital products fail because the planning of complex digital products; especially ones that interact directly with humans, require significant upfront efforts by professional designers and other stakeholders. In case of digital products, this planning involves understanding how the humans using the product live and work i.e. context. This helps to design product's behavior and form that supports and facilitates those human behaviors.

Developing digital products involve equal efforts from developers and market experts. Market experts usually have less involvement in requirements elicitation process and they end up giving requirements which have very little to do with user goals. Their requirements focus more on competition and statistics based on market research. This results into digital products that irritate users, reduce productivity, and fail to meet user goals. Furthermore, these products often blame users for making mistakes which are not their fault. There are certain reasons why these products behave so bad [11]:

- **Ignorance about users**: most softwares are built without much understanding of their users. Stakeholders fail to understand what exactly it is that makes users happy while using their product.
- **Conflicting interests**: usually people who build products i.e. developers are the same people who also design them. Even with appropriate skills and the best intentions, it is simply not possible for programmer to be responsible effectively for the user, the business, and the technology all at the same time.
- The lack of process: there is no standardized design/elicitation process that can be followed to build a successful software

Goal directed design process which Cooper calls *interaction design* [42] is an approach that overcomes above mentioned challenges. It combines multiple techniques of requirements elicitation which are used to understand end user goals and uses

them at different stages of requirements elicitation to understand different aspects of end users and system contexts (see 2.6). It gives solutions that meet the user goals and cater to business and technical constraints simultaneously. The process is roughly divided into six phases [11] as shown in figure 2.1:



Figure 2.1: Six phases of goal directed design process [11]

1. Research:

This phase is aimed at gathering and providing qualitative data (qualitative research) about the potential and/or actual users of the (software) product being built. It involves many techniques like observation, contextual interviews, competitive product audits etc. to collect qualitative data about users and their goals. As an outcome of employing these techniques we obtain **Behavior patterns** (explained in subsection 2.4) which tell us how the product will be used and what goals or motivation lie behind using the product. These Behavior patterns and the goals which are associated with them help us build Personas in the next phase (explained in subsection 2.1.19).

2. Modeling:

The data obtained in research phase is then used to build *domain* and *user* models in the modeling phase. Domain models plot insights about domain in simplified formats such as workflow or information flow diagrams. While the user models form Personas. *Personas* are built From the identified groupings of behaviors, attitudes, aptitudes or skills of the users. A specific Persona tells us what a specific group of users expect and how will they use the proposed product. By prioritizing Personas we can very much prioritize what needs to be focused in the proposed product.

3. Requirements:

This phase forms the bridge between modeling and framework definition phase. From the identified Primary and subsequent Personas and their goals, corresponding functional needs can be formed and prioritized. Also, we decide which tasks are more important than others based on the Persona's motivations and goals. The output of this phase is the **Requirements definition** that considers business and other constraints like user skills, his physical capabilities to state what needs to be accomplished. This requirements definition forms the basis for deciding and designing the framework of use in the next phase.

4. Framework:

In this phase designers develop the overall product concept, and define how the product should behave, its appearance etc. Designers use *Interaction framework* to define the possible interactions with the product. For that they use

interaction design principles [11] or *patterns*. Once interaction framework is emerging, a visual framework that defines products visual form is produced.

5. Refinement:

This phase refines what is produced in previous phase. It produces a *Form* and behavior specification which is essentially a detailed documentation of the design i.e. icons, styles etc.

6. Development support:

This phase essentially helps developers in case of doubts as it is very much possible to have made some mistakes in design specifications or there could be some confusions.

Vision backlog uses first two phases i.e. *Research* and *Modeling* of Goal directed design process to understand problem domain and personal attributes of stakeholders. Vision backlog combines different elicitation techniques as explained in Goal directed design process to gather data about users and domain. This is explained in detail in section 3.3.6. Another software called Vision analytics further analyzes the gathered data to build a *Persona*, which is explained in subsection 4.5.

2.4 Behavior variables and patterns

Behavioral variables and patterns are design tools described by Allan Cooper [11] which are essentially helpful to build Personas (see subsection 2.7 for more about Persona). As seen earlier in subsection 2.3, qualitative research is the first step in goal directed design process and it plays an important role in gathering data about users, their goals and motivations. Qualitative research has an advantage over quantitative research, that it helps analysts to understand problem domain, system contexts (see subsection 2.6) and product constraints with different and more useful perspectives. It also helps analysts to identify patterns of behavior among existing and potential users of a product much more quickly and easily than it would be possible with quantitative approaches [11]. There are many types of qualitative research e.g. stakeholder interviews, Subject Matter Expert (SME) interviews, user and customer interviews etc. which could be employed to gather qualitative data.

It is important, that the stakeholders designing a digital product identify an appropriately diverse samples of users and user types when planning a series of interviews in qualitative research phase so that an entire range of user behaviors regarding a product can be captured. Based on the information gathered during research, designers (in vision backlogs case analysts) need to create a hypothesis that serves as a starting point in determining what sorts of users and potential users to interview. This hypothesis is called Persona hypothesis which is explained in detail in 2.1.19.

The persona hypothesis is the first step at defining the different kinds of users and customers for a product. The hypothesis serves as the basis for initial interview planning; as interviews proceed, new interviews may be required if the data indicates the existence of user types not originally identified. The persona hypothesis attempts to address, at a high level, these three questions:

- 1. What different sorts of people might use this product?
- 2. How might their needs and behaviors vary?
- 3. What ranges of behavior and types of environments need to be explored?

For business products roles, common sets of tasks and information requirements relates to distinct classes of users. Unlike business users, consumers don't have concrete job descriptions, and their use of products may cross multiple contexts. That's why in addition to roles, a Persona hypothesis should be based on variables that help differentiate between different kinds of users based on their needs and behaviors. These variables are called *behavior variables*.

The qualitative data one obtains in the research phase of goal directed design process, can be observed to discover a few *behavior patterns*. Behavior patterns emerge from Behavior variables, and behavior variables are nothing but distinct aspects of the data on which we can categorize our users e.g. education etc. There are few variable types identified in [11], which give us distinct behavior patterns:

- Activities: find out those variables which deal with user activities e.g. frequency of performing certain task etc.
- Attitudes: find out those variables which can deal with user's opinions about the product, workflow etc.
- Aptitudes: find out those variables which can give you some meaningful information about user's aptitude e.g. his qualification etc.
- Motivations: find out those variables which can tell you something about user's motivations behind using the product e.g. necessity or entertainment etc.
- **Skills**: find out those variables which can categorize your users based on the skills they posses.

Once one identifies such behavior variables and plots users against those, he can see that few of his users appear together to suggest a pattern which is nothing but a *behavior pattern*. Behavior patterns together with the goals of those users help analysts to build a *Persona*. For enterprise applications, behavioral variables are often closely associated with job roles. For each job role, variables are collected separately. Although the number of variables will differ from project to project, it is typical to find 15 to 30 variables per role.

Subsection 4.4 explains how sample variables can be identified from that data gathered and patterns can be observed with an example.

2.5 User goals

As mentioned earlier in section 2.3, end user goals should be focus of elicitation activities as they help us identify underlying problems better. It is important to

understand what really a goal means.

There is a fine difference between goals, activities and tasks. A goal is an expectation of an end condition, whereas both activities and tasks are intermediate steps that help someone to reach a goal or set of goals [11]. For years, digital products have been developed keeping tasks in mind which has proven inadequate. Analysts ask questions like *What are the tasks that you perform?* during interviews to form a basis of digital product design. The information obtained gets the work done but may end up being only incremental improvement which will not provide a solution that differentiates your product in the market, and very often will not really satisfy the user.

Goals on the other hand are the drivers behind user actions or behaviors [11]. Thus, the function and behavior of the product must address goals via tasks, typically as few tasks as absolutely necessary. People usually can not tell their goals directly, that's why analysts need to carefully reconstruct goals from observed behaviors, answers to other questions, nonverbal cues, and clues from the environment. User goals tell us what the users are trying to achieve. They tell us user's intentions behind performing certain things.

There are three types of goals suggested which give us some meaningful insights about user's intentions in specific contexts [42] [11]:

1. Experience goals:

These goals essentially tell how the user wants to feel when he is using our product. They tell us their expectations from the quality of interaction with the product. They are simple and personal and hence hard to express. Designers are supposed to choose the form, behavior, motion and auditory elements that communicate the proper feel and effect as expected by the user. We consider these goals while building distinct Personas.

2. End goals:

End goals are nothing but user's motivations for performing tasks associated with using the product. They form the basis for the product's behavior, look and feel and they should be the focus in determining the overall product experience

3. Life goals:

Life goals represent long term desires, motivations, self-image attributes that connect users to the product. They go beyond the context of the product being designed, and tell us *why* the user is trying to accomplish the end goals. They form the basis for product's branding and marketing strategy. These user attributes are helpful to connect distinct Personas to connect to product.

Apart from end user goals there are other types of goals which also need to be taken into account. They are customer goals, business goals, technical goals. Typically, these goals also must be acknowledged and considered, but they do not form the basis for the digital product design. Although these goals do need to be addressed, they must not be addressed at the expense of the user goals.

• Customer goals:

Customers can be different than end users i.e. parents can be customers who can buy a product for their kid who is an end user. They can be consumer customers like parents or enterprise customers like IT managers. Both of them have different concerns while purchasing a product. There will be Personas for customers as well having their own experience or end goals.

• Business goals:

Businesses or organizations have their own requirements for products, services or systems which should be considered while designing a solution. Business goals are partially covered in user or customer Personas. We can also exclusively specify them.

• Technical goals:

Technical goals essentially reduce programmers' tasks. They are important to the developers. These goals must serve business and user goals.

Identifying end user and customer goals along with business goals is the core of goal directed design process which allows us to take best design decisions regarding a product under construction. Vision backlog uses a combination of elicitation techniques to synthesize stakeholder end goals along with their life and business goals which is explained in sections 2.8 and 3.2.4.

2.6 Context

In requirements engineering, identifying all the material and immaterial aspects that affect the system or a product is as much important as identifying user or stakeholder goals. To identify such aspects, the future system when it comes into reality is imagined. This helps to identify parts of the real world those can influence the system. To form complete and correct requirements along with correct user goals, it is important to precisely identify relationship between material and immaterial aspects. Those aspects are called *System context*. Few examples of such aspects of reality that influence the system or its requirements are:

- People (stakeholders or a group of stakeholders)
- Systems in operation (other technical systems or hardware)
- Processes (technical or physical processes or business processes)
- Events (technical or physical)
- Documents (e.g. laws, system documentation)

If the system context is incorrectly or incompletely considered during requirements engineering, it can result in incomplete and erroneous requirements ultimately resulting in failed system.

The requirements for a specific system are always bound to the stakeholders for whom it is being developed, or standards or legal guidelines it has to adhere to. Which means, a requirement is defined for a specific context and can only be interpreted correctly in regard to this specific context. It is requirements engineer's responsibility to define system context properly. To do that he has to separate the system from its context and from the parts of the reality that are irrelevant for the system, see figure 2.2.



Figure 2.2: System and context boundary [47]

The context boundary in case of Vision backlog is already outlined in section 1. In case of Vision backlog, the software itself along with the technology related decisions, infrastructure used fall inside the system boundary.

As already mentioned, every action we perform is bound to a specific context. The contexts affect the system or it's use in many ways [21]. All the actions users perform, they are scoped in a specific physical, environmental contexts and are valid in that scope. Understanding that context is as essential as understanding the user goals.

There are three types of contexts suggested to which user goals are bound [11]:

- 1. **Technical** context tells us details like what technical competency user is required to have, his technical skills etc. to accomplish a certain specific task.
- 2. **Business** context tells us the business value and the business drivers for performing a certain task and user's and organization's motivation behind it.
- 3. Environmental context tells us the correlation between the user performing a certain task and the physical space around him, co-workers etc. It also tells the frequency and importance of tasks those are being performed.

Understanding the contexts associated with user goals help analysts to build a better vision for the product with concrete reasons for choosing certain design over others. To understand contexts pertaining to specific stakeholder goals and for the analysts to be able to clearly define system context, Vision Backlog uses a combination of elicitation techniques which is further explained in subsection 4.2.

2.7 Persona

It is insane to think that if you build a product that has functions that caters to a vast set of people, would be a successful product. It is a wrong assumption as it does not work so in reality. When you broadly and randomly extend a product's functionality to include many functionalities, you increase the cognitive load and navigational overhead for all users [11]. This is explained with the following figure 2.3:



Figure 2.3: Why persona is important: different people having different needs [11]

As can be seen in the picture, if one tries to design a car that pleases many people, you will have a car that has all the functions everybody wishes, but it wouldn't please anybody. Hence it is important to realize that a product should be designed with specific types of individuals with specific needs. Personas tell us those specific types of users and their specific needs. In Goal directed design approach we represent those users through a Persona whose needs best represent the needs of a larger set of key constituents. However, it does not mean that we compromise the needs of other users. Those less significant users are also represented through secondary Personas.

As described earlier in subsection 2.3 one of the principal outcomes of qualitative research is a rising set of behavior patterns i.e. identifiable behaviors that help categorize ways of using a potential or existing product. These patterns suggest goals and motivations of different types of users. In business and technical domains, these behavior patterns tend to map into professional roles; for consumer products, they tend to correspond to lifestyle choices. Behavior patterns and the goals associated with them are the key in creation of Personas in the modeling phase of Goal directed design process. Market research helps to select and filter valid Personas that fit our business requirements.

A Persona essentially represents a user model. How users behave, how they think, what they wish to accomplish, and why such sort of questions can be better understood and communicated through a Persona. Personas are not real people, but they are based on the behaviors and motivations of real people we have observed. From the behavior patterns we identify and the user goals we obtain, we can see that there are certain users which exhibit similar behavior and share similar goals. They represent distinct groupings of behaviors, attitudes, aptitudes, goals and motivations. Persona hence becomes an important tool in a narrative, scenario based

design approach to incrementally and iteratively generate fitting design decisions in the later Framework definition phase of Goal directed design process. Persona also serves as a powerful communication tool between developers, managers and other stakeholders that helps them to understand the logic behind design decisions so that they can prioritize the features.

Personas also help us address following three main design issues which arise typically during product development:

1. The elastic user:

Every member on development team usually has his own understanding of a potential user and his needs for the product being developed. When discussing product's design related decisions, each member ends up talking through his own perspective on the user and hence the potential user becomes *elastic*. Lack of precision about the user can lead to a lack of clarity about how the product should behave.

2. Self-referential design:

Many programmers and designers end up projecting their own ideas, motivations, goals onto the product being developed. They fail to understand what exactly is needed by the users, their goals and motivation behind using the product being developed. Such products would satisfy only a subset of potential users that have similar needs or goals as of programmers or designers.

3. Edge classes:

Edge classes are those users whose goals or motivations are not too clear and focusing too much on such users can lead to bad design decisions. Persona avoids such situations by making the members on the product development team to ask, if a specific Persona would perform a specific function often so that they can prioritize functions with clarity.

Ideally multiple such groups of our potential users can be identified and each one of them would signify a distinct Persona. We can then decide which group of users is the most important to us through market research and consider corresponding Persona as our *Primary Persona* so that our designers and developers have a common understanding of what are the most important user needs and hence what needs to be prioritized concerning designs and features. Similarly *Secondary Personas* can be chosen to further refine your design decisions for your system to be more versatile. Figure 2.4 shows a sample empty Persona:

	Age	
	Education	
	Role	
	Main Responsibilities	
	Expert in	
	Technical competency	
	Probable user type	

Figure 2.4: A sample blank Persona explaining user details

On the left hand side, a short life story of our Persona is written along with a dummy picture of him/her. On the right hand side, important attributes are listed with their corresponding values to be filled. These attributes could also correspond to behavior variables. The last cell tells what probable adopter category this persona might fall into. Subsection 4.5 explains with an example how such a Persona can be created.

2.8 Question types

Ethnographic techniques like stakeholder or subject market expert interviews play a vital role in acquiring as much qualitative data as possible. We need this data to understand the problem domain, stakeholder's/end user's needs and contexts of use of the product that is being developed. It is important to avoid a fixed set of questions during such interviews or contextual inquiries as analysts can potentially miss out a lot of valuable information just because they presumed something. Interviews must be based on a premise that analysts know very little about the problem domain and must learn what is important from the people they interview. It is good to have an evolving list of topics depending on the problem domain that analyst wish to cover during interviews [11].

Cooper classifies questions into following different categories [11].

1. Goal oriented:

These type of questions are aimed at gathering information about the user goals, their importance etc. Try to gather information about opportunities for improvements, about priorities regarding product features by asking questions like What activities currently waste time? or What helps to take a decision? etc.

2. System oriented:

These type of questions are aimed at gathering information and more insights about the existing systems, products etc. and about users' interactions with such systems or products. Ask questions about product functions, preferences or failures like *What are the most common operations you do with the product?* or *Which parts of the product are used most?* or *How do you rectify errors or problems?* etc.

3. Workflow oriented:

These type of questions are aimed at understanding how users perform certain tasks, steps they follow, any bottlenecks in the processes etc. Ask questions like What do you do first and next everyday? or How often you would do a certain thing? or What can go wrong while doing something? etc.

4. Attitude oriented:

These type of questions are aimed at gathering information about personal attributes of the users like their aspirations, personal choices, interests etc. Ask questions like *What are your personal goals for next five years?*, or *What do you like most about your job?* etc.

5. Aptitude oriented:

These type of questions are aimed at gathering information about skills, education, special training users have which help them in performing specific tasks. Ask questions like *Highest education you have?* or *Skills you typically need to carry out certain task?* etc.

6. Context oriented:

These type of questions are aimed at gathering supporting information that helps analysts to understand user's environment, his role etc. Ask questions like What role you are playing while performing a certain task? or Who else is involved in performing this task? etc.

Categorizing questions into different types help analysts to cover all the important details of all the major and necessary aspects of products and surrounding contexts. It helps stakeholders or users to think thoroughly about their daily activities, priorities and preferences for using a certain product or reasons for following a specific workflow. Subsection 3.2.4 lists all the questions that I have asked and classifies them according to the types mentioned above.

2.9 Summary

This section described all the major concepts that have been used to build Vision backlog. It also outlined their importance in Vision backlog. Following chapter 3 explains how the concept of Vision backlog is built using above concepts and further gives technical details.

Chapter 3

Solution

3.1 Introduction

Section 1.1 makes it clear that the stakeholders should be able to state their goals, needs and expectations themselves along with all the important contextual information around it. They should be able to state those things whenever they want and not just during meetings, a question must be asked- how can they? Another challenge is to make them focus on the problems they are facing rather than the probable solutions. A simple answer to such questions would be by building a software that is always available and which lets the stakeholders state their goals, needs, expectations regarding a product or a process effortlessly and effectively; by focusing them on the underlying problems. For requirements analysts, another software must be developed that analyzes the data entered by the stakeholders and helps them to understand stakeholder goals and subsequently to build Personas. The first software developed is called Vision backlog, and the analytics software is called Vision analytics.

It is important to note that stakeholders would not completely take over requirements elicitation activity, instead it will positively be shared among stakeholders and analysts. Vision backlog would not replace requirements analysts but it will ease their work by reducing efforts and time spent on eliciting correct and complete requirements. It combines examined elicitation techniques in such a way that stakeholders can use them without having to learn anything new with the goal of providing analysts with rich informative data.

Requirements elicitation techniques and methods (described in 2.1) are further classified and shortlisted to be used in vision backlog to extract user goals, surrounding contexts and users' personal attributes- refer subsections 3.2.1 and 3.2.2. The theory of *diffusion of innovation* is used to identify adopter category of stakeholdersrefer 4.3. Vision backlog concentrates on first two phases of Goal directed design; shortlisted elicitation techniques in 3.2.2 are used to collect data about stakeholders' goals, whereas adequate analytics are provided to the analysts to build Persona- see 4.5. Few sample behavior variables are identified and presented to the analysts so that they can see behavior patterns. This would be an input to built Personas- refer 4.4. Shortlisted elicitation techniques are also used to understand underlying context of stakeholder's goals and to extract different types of stakeholder's goals 3.3 and 3.4. Section 3.2 explains Vision backlog conceptually and section 3.3 explains its technical details.

3.2 Building the concept

This section explains Vision backlog conceptually. Subsection 3.2.1 proposes a criteria so that elicitation techniques listed in 2.1 can be classified. Subsection 3.2.2 shortlist elicitation technique based on the criteria already established and describes the purpose for which they will be used. Subsection 3.2.4 lists all the questions asked in Vision backlog; each of which refers to a specific shortlisted technique. It also describes it's type and what it will capture.

3.2.1 Classification of elicitation techniques

The techniques listed in subsection 2.1 come from different disciplines like social science, design thinking etc. They are used in different situations and for different purposes and not all the purposes identify with the problem I am solving. Which means that not all the techniques listed above are of help to build Vision backlog. To understand and choose what would be helpful to me, it is necessary to classify them in such a way that selected techniques echo with Vision backlog's purpose.

These techniques can be classified in many ways. One way is to classify them according to the means of communication they involve: *conversational, observational, analytic* and *synthetic* [64]. The conversational method is based on verbal communication between two or more people. Methods in this category are called verbal methods. The best example is Interview. Observational method is based on understanding problem domain by observing human activities. There are requirements which people can not verbally articulate properly. Those are acquired through observational methods. Example of such methods is Protocol analysis. Analytic methods provide ways to explore the existing documentation of the product or knowledge and acquire requirements from a series of deductions which help analysis capture information about application domain, workflow and product features. Examples include card sorting. Synthetic methods into a single method. The provide models to explore product features and interaction possibilities. Example is Prototype with Storyboards.

Although the above mentioned techniques make sense, these schemes are not much of a help considering vision backlog's context. The primary concern is that the stakeholders should focus on the problems they are facing and should not get distracted by solution or implementation details. Another challenge is that a technique should be imitated in a software. Examining the literature that describes these techniques [27] [65], following classification criteria is established. Direct answers to the following questions 1 and 2 can be found in the above mentioned literature. Considering how much in depth knowledge is required to use a specific elicitation technique, whether it can be used by stakeholders is indicated by question 3, keeping in mind that stakeholders must not learn anything new. Question 4 classifies techniques as
per their ability to be imitated as a software. There already exist softwares which implement certain techniques [1] [40] [43] [7]. Other techniques are used in Vision backlog and answer to the question how? can be found in following subsection 3.2.4. Techniques which fulfill a specific criteria are marked with a *tick*. Techniques which fulfill all the four criterion, are marked with brown color:

- 1. A: is the technique intended to locate and scope problem and not about solution oriented?
- 2. *B*: can techniques be used individually and it is not performed as a group activity?
- 3. C: can the technique used by both stakeholder and analysts?
- 4. D: can the technique be imitated as a software?

Following figure 3.1 plots techniques listed in subsection 2.1 against above mention criteria.

Technique	A	В	С	D
Interview	~	~	~	~
Questionnaire	~	~	~	~
Task analysis	~	v	~	~
Domain analysis	~	V	~	v
Observation	~	V		
Protocol analysis	V	V		
Prototyping			~	
Brainstorming			~	~
Card sorting		~	~	~
JAD			~	
Scenarios		~		
Viewpoints		~		
SWOT analysis	~	v	~	~
Theory of change	V	~	V	~
Problem definition	V	~	V	~
Repertory grids		✓	~	✓
Laddering		~	~	~
Literature review	~	~		
Persona	~	V	~	V

Figure 3.1: Classification criterion for elicitation techniques studied for Vision backlog

Techniques of interests should be those which fulfill all the four criterion.

3.2.2 Shortlist and purpose of respective techniques

Classification alone isn't sufficient as the software system can not really imitate techniques like *Observation*. Thus, they are further shortlisted to the final list which can be used in our software solution. Figure 3.2 lists those techniques along with their purpose in Vision backlog.

Nr	Technique	Purpose
1	Interview	I used structured stakeholder interviews to gather information about their personal attributes, their goals, business drivers behind them, underlying contexts and the domain
2	Questionnaire	Similar to the interview
3	Domain analysis	Used along with interview to gather domain specific information like vocabulary, specific terms used during execution of a specific process or a task
4	Task analysis	Used to gather information about the tasks stakeholders perform, subtasks and concrete steps, with contextual information like specific skills required
5	Problem definition	Used along with questionnaire to gather information about stakeholder's problems with current processes or tasks along with possible alternatives
6	Theory of change	Used along with interview to make stakeholder think about their high level goals with tasks those help them achieve the goal, and potential risks etc.
7	SWOT analysis	Used to gather information about possible improvements, challenges and alternatives to the tasks stakeholders performs
8	Persona	Provide all the project stakeholders with a common understanding of their target user

Figure 3.2: Shortlisted elicitation techniques and their purpose in Vision backlog

As a whole all these techniques gather data about stakeholder attitudes, aptitudes, their skills, their tasks, motivations behind those tasks and their surrounding contexts.

3.2.3 Using shortlisted techniques in Vision backlog

The shortlisted elicitation techniques in subsection 3.2.2 intend to gather diverse information at different stages of elicitation process. In reality these techniques are practiced in different formats with different surrounding environment settings. For them to be usable in a software a common factor which is 'questioning' that each of these techniques does, is taken a basis. Vision backlog presents stakeholders with many questions, each of which corresponds to a specific elicitation technique fulfilling it's status purpose in figure 3.2. The questions are intended to ask information about stakeholder goals, activities, aptitudes, attitudes, skills etc.

Following subsection 3.2.4 consolidates all the questions that have been asked along with their type, what it is intended to capture and to which elicitation technique it belongs to.

3.2.4 Set of questions

Nr	Question	Туре	What will it capture	Which technique
1	My name	-		1 & 2
2	My age	-		
3	What education I have	Aptitude oriented	Aptitude	
4	My short story	Goal oriented	-	
5	I work in this company as	Context oriented	context	
6	My interests	Aptitude oriented	Aptitude and attitude	
7	My technical skills	Aptitude oriented	Skills	
8	I am a rational thinker	Attitude oriented	Attitude	
9	I like to welcome good new changes	Attitude oriented	Attitude	
10	I like to embrace science	Attitude oriented	Attitude	
11	I am socia	Attitude oriented	Attitude	-
12	I am cosmopolitan	Attitude oriented	Attitude	-
13	I seek latest information	Attitude oriented	Attitude	-
14	This is what is exciting about my job	Attitude oriented	Attitude	
15	This is what I don't like about my job	Attitude oriented	Attitude	
16	How I see myself in 5 years	Attitude oriented	Attitude	

Figure 3.3: Actual set of questions - part 1

Nr	Question	Туре	What will it capture	Which technique
17	What task I perform	Goal oriented	Activity	4
18	How often do I perform it	Workflow oriented	Behavior	4
19	How important it is		Context	4
20	What role I am playing while performing this task	Context oriented	Context	5
21	What steps I perform while doing this task	Workflow oriented	Activity	6
22	Why do I perform this task	Goal oriented	Motivation	6
23	Why is it important to perform this task	Goal oriented	Motivation	7
24	What tools you need to perform this task	Context oriented	Context	4
25	Do you need any special training to perform this task	Context oriented	Context, Aptitude	3
26	What kind of training you need to perform this task	Aptitude oriented	Still	3
27	Can something go wrong while performing this task	System, workflow oriented	Context	3
28	What are the effects of performing this task	Context oriented	Context	4
29	Do you use any special vocabulary related to this task	Context oriented	Context	6
30	Who else is involved in performing this task	Context oriented	Context	3
31	What should happen before and after performing this task	Workflow oriented	Activity, Context	4
32	Can you think of any improvements to this task	Goal oriented	Activity	6
33	How important you think is this improvement	Goal oriented	Context	4
34	Main benefit of this improvement	Goal oriented	Motivation	7

Figure 3.4: Actual set of questions - part 2

3.3 Building the software

This section gives technical details of Vision backlog and Vision analytics softwares. Subsection 3.3.1 associates questions listed in 3.2.4 to UI elements. Subsection 3.3.3, 3.3.4 and 3.3.5 contain Use case diagrams, Class diagram and ERR diagram for Vision backlog respectively. Subsection 3.3.6 describes in detail frontend and backend of the Vision backlog and Vision analytics web applications.

3.3.1 Question categories and corresponding user interface (UI) elements

The intention behind collecting data through Vision backlog is that it can further be processed in Vision analytics which analysts can benefit from. For the data to be processable, it has to be captured with an appropriate structure. Presenting user with a series of random questions also isn't what is intended, questions have to guide stakeholders through elicitation techniques smoothly to fulfill it's purpose. Hence, questions have to follow a certain sequence and a structure. The questions are classified into two groups so that stakeholders can follow essentially two sequences.

- 1. Questions from 1 to 16 in figure 3.3 focus on stakeholder's life goals. They capture data about user's life goals and his attitude and aptitude. Answers to these questions gives input to understand what type of user he is during Persona synthesis.
- 2. Questions from 17 to 34 in figure 3.4 focus on stakeholder's end, business and experience goals. They capture data about the actual tasks the user performs and his goals, motivation behind performing those tasks, other contextual information like frequency, importance etc. Answers to these questions gives qualitative data that can be used to identify behavioral variables and patterns during Persona synthesis.

Along with a proper logical and engaging sequence of the questions, a proper format of the questions to capture structured answers is also important. A wise selection of UI elements should help to restrict stakeholders to limit their answers to specific measurable values. Tables 3.1 and 3.2 cover all the questions along with their corresponding UI element choice.

Nr	Question	UI element choice
1	My name	Text field
2	My age	Text field
3	What education I have	Radio button
4	My short story	Text area field
5	I work in this company as	Select box
6	My interests	Text area field
7	My technical skills	Select boxes
8	I am a rational thinker	Radio button: Yes No
9	I like to welcome	
	good new changes	Radio button: Yes No
10	I like to embrace science	Radio button: Yes No
11	I am social	Radio button: Yes No
12	I am cosmopolitan	Radio button: Yes No
13	I seek latest information	Radio button: Yes No
14	This is what is exciting	
	about my job	Text area field
15	This is what I don't like	
	about my job	Text area field
16	How I see myself	
	in 5 years	Text area field

Table 3.1: UI element choices - part 1

Nr	Question	UI element choice
17	What task I perform	Text field
18	How often do I perform it	Select box
19	How important it is	Select box
20	What role I am playing	
	while performing this task	Select box
21	What steps I perform	
	while doing this task	Text field
22	Why do I perform this task	Text area field
23	Why is it important	
	to perform this task	Text area field
24	What tools you need to	
	perform this task	Text area field
25	Do you need any special	
	training to perform this task	Radio button: Yes No
26	What kind of training you	
	need to perform this task	Text area field
27	Can something go wrong	
	while performing this task	Text area field
28	What are the effects of	
	performing this task	Text area field
29	Do you use any special	
	vocabulary or terms those are	
	related to this task	Text area field
30	Who else is involved in	
	performing this task	Text area field
31	What should happen before	
	and after performing this task	Text area field
32	Can you think of any	
	improvements to this task	Text area field
33	How important you think is	
	this improvement	Select box
34	What is the main	
	advantage of this improvement	Select box

Table 3.2: UI element choices - part 2

3.3.2 Brief technical introduction

As mentioned in section 3 that Vision backlog and Vision analytics are softwares, it was also important to decide what sort of software they should be that serves the purpose. One of the goals of Vision backlog is that it should always be available to use and that stakeholders should keep entering and updating the data whenever they wish and from wherever they wish. Analysts are also involved who are going to benefit from this data in analytics form.

There are two choices one can have in such a situation- a desktop application that

is baked up by the cloud and a web application hosted on the server. Desktop application was not chosen due to following reasons:

- 1. Operating system (OS) dependency: desktop applications are OS dependent i.e. it is required to develop separate applications for Windows and Mac
- 2. Desktop application must be installed on the user's personal computer. For a few users it could be an overhead, considering factors like storage space they have on their computer or it's configuration
- 3. Updates: to update a desktop application one might explicitly need to ask the user to turn the updates on. In case the user forgets to turn on the updates or simply ignores it, then he will end up missing new features

Considering above mentioned points, Vision backlog is built as a web application which can be accessed over Internet any time and from anywhere. Regular updates can be made available to the user without him having to do anything about it. Fine tuning web application for different web browsers is easier than building separate applications for different operating systems. A web application can be rendered on tablet PCs with same experience as on laptops by simply making it responsive.

3.3.3 Use case diagrams

A use case describes *who* can do *what* with the concerned system. It is a technique that captures system's behavior based on scenarios that represent a specific functionality the system should support. Typically use case describes a specific functionality to achieve a specific objective, and therefore many use cases can be required to understand a set of behaviors that comprise a whole system. Actors are external entities which could be a user or a group of users or other systems which interact with the system to accomplish a certain task. There are three types of relationships possible between use cases. First, a use case may *include* another use case, which means behavior of the included use case is a part of including use case. Second, a given use case may extend another use case, which means the extending use case is dependent on the extended use case. Third, a use case be a generalized form of another use case [20].

Following figures 3.5 and 3.6 depict respectively use case diagrams for Vision backlog and Vision analytics. The rectangle represents a system boundary. The ovals represent specific use cases. The arrows represent relationships between use cases. In figure 3.5 stakeholder is an actor whereas in figure 3.6 an analysts is an actor. Arrow with jjextend¿¿ notation represents an extend relationship between two use cases e.g. view profile makes sense only create profile is executed.



Figure 3.5: Vision backlog use case diagram



Figure 3.6: Vision analytics use case diagram

3.3.4 Class diagram

Class diagram is a UML structure diagram that represents static structure of a system. It shows the types being modeled within the system which could be a class, an interface, a data type or a component. A class can have attributes and methods. Attributes and methods have access identifiers like public, private and protected. Classes can be related to each other through relationships [6].

Following figure 3.7 depicts Vision backlog modeled as a class diagram. The rectangle represents a class whose name is written at the top of the rectangle. Second row in each rectangle shows attributes of that class along with their data types and their access modifiers. + means public whereas - means private. Arrows between classes represent a relationship with cardinality mentioned on both the ends.

ApplicationUser is a super class and Stakeholder inherits from it. The relationship between ApplicationUser and Profile is composition with cardinality 1..1 which means an ApplicationUser can have only one Profile and when you delete ApplicationUser, Profile will also be deleted.



Figure 3.7: Class diagram

3.3.5 EER diagram

The enhanced entity relationship (EER) model is an extension of entity relationship (ER) model, developed to precisely describe properties and constraints of more complex databases. It improves ER model's presentational capabilities. It includes all the concepts of ER model, and additionally concepts like *subclass, superclass, specialization* and *generalization* [15]. It includes EER diagrams that accurately represent the requirements of complex databases. Following figure 3.8 represents EER diagram of Vision backlog. It is generated through MySQL workbench.

Rectangles are database tables with their name at the top e.g. table profiles. Table columns and their data types are listed bellow the table name. Yellow key symbol represents primary key e.g. for profiles table ProfileId is the primary key. Filled blue

diamond represents not null simple attribute e.g. profiles table has Age which is a not null attribute. Filled brown diamond represent not null foreign key e.g. profiles table has StakeholderId as a not null foreign key. Empty blue diamond represent simple attribute which can be null e.g. in profiles table, attribute Designation can be null. Empty brown diamond represent a foreign key which can be null. Arrows represent relationships which have cardinality and modality. Cardinality can be 1 or Many and the symbol is placed on the outside ends of the relationship line, closest to the entity. Modality can be 1 or 0 and the symbol is placed on the inside, next to the cardinality symbol. For a cardinality of 1 a straight line is drawn. For a cardinality of Many a foot with three toes is drawn. For a modality of 1 a straight line is drawn. For a modality of 0 a circle is drawn. For example, exactly one profile entity can have one or more skills.



Figure 3.8: EER diagram

3.3.6 Frameworks and application architecture

Developing a web application involves many technologies. A typical web application has a backend (web server) that handles the business logic, a frontend (cliend-side) that interacts with the user and a database that stores the data. There are wide range of technologies that can be used for backend, frontend and database. The decision to choose a specific set of technologies usually depends on business requirements and also personal experiences. The main concern while developing Vision backlog is that, in the future it should be possible to integrate it with other systems or at least expose its APIs to other applications. Another important concern is to keep client-side technology independent of server side technology as in the future either of them can be changed independently of the other if need arises.

Choosing a set of technologies i.e. technology stack for a web application essentially involves choosing a technology for client-side to go along with HTML and CSS and choosing a server-side technology. Following figure 3.9 illustrates few sample technologies that are used on client and server side in a typical web application.



Figure 3.9: Full web stack options [60]

Typically on the client-side we require HTML, CSS and perhaps a styling framework like Bootstrap or Material Design Lite (MDL). Javascript libraries like ReactJs or a Javascript framework like Angular 2 are also used on the client-side. There are many server-side frameworks based on a specific programming language like Django is python based, Rails is Ruby based etc. Database choices include MongoDB or MySQL etc.

Overview

Vision backlog and Vision analytics are web applications that share a common backend. Backend and frontends are built as separate independent projects so that in case either of them can switched to a new or a different technology regardless of the other. Backend is a REST API with JSON as request and response format. Frontend makes requests to this REST API and processes the received response.

The backend is developed using .NET Core. Two frontends i.e. Vision backlog and Vision analytics are built using Angular 2 and MDL as a styling framework. MySQl is used as a database. See following figure 3.10



Figure 3.10: Simplified overview of the application architecture

All the necessary help required to use the applications- their purpose, their advantages, workflow, explanation of the terms used, usage guidelines are included in the applications itself. Mock-ups were created before starting the actual development of the applications, those are included in subsection 3.3.7. Subsections 3.3.6 and 3.3.6 describe in detail how backend and frontend are designed and developed.

Backend

The backend serves a REST API and it is built with .NET core V1.1. It follows repository pattern [56]. Entity Framework Core V1.1 takes care of database migrations and updating the database. Models are POCO classes [46] which define the properties of the data that will be stored in the database. Models used in Vision Backlog are depicted in the below figure 3.11:

```
7
   ι
10
       public class User_Task
11
       {
12
            [Key]
13
            public long TaskId { get; set; }
14
            public string What { get; set; }
            public string How_often { get; set; }
15
            public string How_important { get; set; }
16
17
            public string User_role { get; set; }
18
            public string Why { get; set; }
            public string Why_important { get; set; }
19
20
            public string Tools { get; set; }
21
            public bool Is_training { get; set; }
            public string What_training { get; set; }
22
            public string Errors { get; set; }
23
            public string Effects { get; set; }
24
            public string Vocabulary { get; set; }
25
26
            public string Peers { get; set; }
27
            public string Pre_post_conditions { get; set; }
28
            public string Improvement { get; set; }
29
            public string How_imp_improvement { get; set; }
            public string What_advantage_of_improvement { get; set; }
30
31
32
            [ForeignKey("FeatureId")]
33
            public long? FeatureId { get; set; }
34
35
            [ForeignKey("StakeholderId")]
            [Required]
36
            public long StakeholderId { get; set; }
37
38
39
            public virtual ICollection<Step> Steps { get; set; }
40
41
            public User_Task()
42
            {
43
44
            }
45
46
       }
```

Figure 3.11: Sample model

Controllers are responsible for handling HTTP requests, responses and they contain the business logic. Each model has its own controller. Figure 3.12 shows one of the actual controller class.



Figure 3.12: Sample controller

Repositories are responsible for storing and retrieving data from the database with the help of Entity Framework Core [16]. Each Model has it's own repository. Figure 3.13 shows one of the actual repository class.

```
8
    namespace Vision_backlog_backend.Repository
        public class User_TaskRepository : IUser_TaskRepository
10
11
            private readonly WebAPIDataContext _context;
12
13
14
            public User_TaskRepository(WebAPIDataContext context)
15
                 _context = context;
16
17
18
            }
19
20
            public IEnumerable<User_Task> GetAll()
21
22
                 return _context.User_Tasks.Include(task => task.Steps).ToList();
23
            }
24
25
            // Method that returns all the tasks of a specific user i.e. logged in
              used making the request
26
27
            public ICollection<User_Task> GetUserTasks(long stakeholderId)
28
29
30
                 var userTasks = _context.User_Tasks
.Include(task => task.Steps).ToList()
31
32
                      .Where(task => task.StakeholderId == stakeholderId).ToList();
33
34
                 return userTasks;
            }
35
36
            public void Add(User_Task item , long stakeholderId)
37
38
                 item.StakeholderId = stakeholderId;
39
40
                _context.User_Tasks.Add(item);
41
                _context.SaveChanges();
42
            }
43
```

Figure 3.13: Sample repository

A token based authentication is enabled and on successful login, the backend returns a JSON Web Token (JWT) [31] which is used to authorize a user on subsequent requests. Users can register themselves with simple registration process which requires email, first name, last name and a password. Swagger [54] enabled to test the API.

My API

Account	Show/Hide	List Operations	Expand Operations
POST /api/Account			
Auth	Show/Hide	List Operations	Expand Operations
Post /api/Auth/login			
Feature	Show/Hide	List Operations	Expand Operations
GET /api/Feature			
Post /api/Feature			

Figure 3.14: Sample swagger API

Frontend

Both the frontends i.e. Vision backlog and Vision analytics are built with Angular 2 [2] and MDL [38] used for styling. Both the applications are modularized with different components taking care of different functionalities. Dedicated services communicate with the backend to fetch and send data. Figure 3.15 shows one of the actual service:

```
11 @Injectable()
12 export class FeatureService {
13
14
      private base_url = 'http://localhost:4784/api/';
15
      constructor (private http: Http) {}
16
17
18
      private extractData(res: Response) {
19
20
        let body = res.json();
        return body || { };
21
22
      }
23
      private handleError (error: Response | any) {
24
        \ensuremath{{\prime}}\xspace // In a real world app, you might use a remote logging infrastructure
25
26
        let errMsg: string;
        if (error instanceof Response) {
   const body = error.json() || '';
27
28
29
           const err = body.error || JSON.stringify(body);
           errMsg = `${error.status} - ${error.statusText || ''} ${err}`;
30
31
        } else {
32
           errMsg = error.message ? error.message : error.toString();
33
        }
34
        console.error(errMsg);
35
        return Observable.throw(errMsg);
36
      }
37 }
```

Figure 3.15: Sample angular service

Templates render the corresponding HTML. Figure 3.16 shows one of the actual service:

```
...s-analyst-frontend\src\app\feature\feature.component.html
                                                                                         1
 1 <div class="feature-form-container" *ngIf="tasks">
 2
        <!-- create a feature or update existing one -->
 3
        <span style="float: left;">
 4
 5
            <button class="mdl-button mdl-js-button mdl-button--raised" (click)
              ="openFeatureList()"> Add to existing</button>
 6
        </span>
        <span style="float: right;">
 7
            <button class="mdl-button mdl-js-button mdl-button--raised" (click)
 8
                                                                                        P
              ="openCreateFeature()"> Create new</button>
 9
        </span>
10
11
        <router-outlet></router-outlet>
12
13 \langle div \rangle
14
15
```

Figure 3.16: Sample angular template

In Vision backlog application users/stakeholders can register and login. Once they login, they can create profile, add tasks or edit existing tasks etc. At each login they get to see their own data i.e. tasks, profile. In Vision analytics application requirements analysts can see consolidated data as entered by stakeholders. The data is presented in nice analytics form and required help is provided which will guide them to create their own Personas.

3.3.7 A walk through the running application

Vision backlog's main aim is that stakeholders should be able to enter data about the tasks they perform, and more contextual data like why they perform those tasks, what tools or knowledge they require to perform those tasks, if there are any alternatives to those tasks etc. Along with this information, they also would provide personal information like education they have had, their job designation, skills they posses and so on. Important thing while designing this application is how these questions are presented to the user so that they can efficiently and effectively answer them.

Main aim of Vision analytic is that requirements analysts will be able to take a quick look at the consolidated data and subsequently build Personas from the data that is entered by stakeholders in Vision backlog This data will be presented to them in the form of suitable analytics. Analysts should be able to sort, filter data according to different properties. Additionally, they should be provided with as many behavioral variables as possible so analysts can see clearer patterns which eventually will help them to build Personas. The challenge here would be to display analytics in a suitable form that reduces analysts efforts and time.

Following subsections 3.3.7 and 3.3.7 explain in detail how both the frontend applications i.e. Vision backlog and Vision analytics work. Mockups and actual screenshots of the application are also included for readers to get a clear idea.

Stakeholder's view

Stakeholder's view is called *Vision backlog*. The user needs to register himself and login to start using the application. On the very first login the user is presented with a Help pop-up that explains the purpose of the application and different functionalities available and their importance. The user has options to see/edit his profile, to see the help section and to create a task or see existing tasks to edit further. The application works with three main functionalities: *Profile creation*, *Task creation* and *Task edition*.

During **profile creation** user creates his user profile which collects data about his personal attributes necessary to guess the adopter category he belongs to. This section asks questions from 1 till 16 from figure 3.3. The user fills the information about his personal aspirations, interests, skills etc. This information will help analysts to understand his life goals. Wherever possible a structured data will be obtained. Additionally, it makes the user familiar with the look and feel of the application. Each step describes at a maximum possible details, what that step is and why it is important to perform that step.

Vision backlog	⑦Help Profile Logout(→
Name Age Role you play select V	
	Save
	"

Figure 3.17: Stakeholder profile screen mock up

♠ Vision Backlog				0	θ	Э
				\mathbf{x}		
	1 That's me!	2 What am I like?	I am at my workplace!			
	The information that you enter in t you can handle. How and why you	this and next two tabs tell us what is it like to be 'You'l What are your inter are involved in this job, what you like about it, what you don't. We keep thi	rrests, what kind of latest technologies is information confidential.			
	This help us understand what kind For example, should the proposed	d of user you are, which in turn makes easier for us to take certain decision system use touch controls etc.	ons about the system we are building.			
	l am	Nitishame? Dummy name is fine!	-			
		28 _{/ age}				
	I have	Vocational training Diploma Bachelor N	Masters Doctoral			

Figure 3.18: Actual stakeholder profile screen

During **task creation** user creates tasks he performs as a part of his job which he thinks are important. This section asks questions from 17 till 34 from figure 3.4. User provides details such as how frequently, how important it is, the reason behind performing that task, any improvement he can think of etc. user can add as many details as he wants at a time and save it. He can edit/add information anytime later. Structured data is obtained wherever possible. Each step describes what it is about and why is it important to perform it. Additionally, an assistant is always present on the screen which takes a sample example and explains it with possible answers.

During *edition of tasks* user edits existing tasks. A list of all the tasks is always present and the user can open individual tasks and make changes of his wish. He can see how many tasks are present and can navigate through them. Figure 3.19 and 3.20 show mockup and actual screenshot of task list screen. Tasks are displayed as sticky notes.

	vision-back	klog	
Vision backlo	9		
Total Applies A	≔ 0		
Task one Edit Task two Edit	Tab1 Tab2 What task How important	Tab3	
			Save
			"

Figure 3.19: Tasks list screen

♠ Vision Backlog					0	θ	€
and the second second				an a			
Total number of tasks are 2 Nitish task one	1 Quick Intro Give us few short answers	Think about it Why is it important		Supporting details Let's uncover some important details	Finishing details Few more details to voice your opinion		
EDIT Nilish task two	The information that yo oftenly you perform it a use. Apart from this, it existing workflow taking	u enter in this and following two tabs gives us all the nd how important it is. Also, it gives us more detailed gives you an opportunity to suggest any other way into consideration your valuable suggestions- those	important details about the task th d information like- if there is any spe ys of performing this task which co you have acquired through everyd	at you perform. It helps us unders scial traning required to perform th uld be more efficient. The entire ay experience on the floor.	stand why you need to perform it, how his task or any special vocabulary you information will help us improve the		
EDIT	The understanding we other important activitie	develop from this information can give us hints like s can be focused that improves efficiency, etc. Take	whether any activities that you per your time but fill this information in (arform as a part of this task can l detail. You can always add/edit th	be dropped or redesigned or perhaps is information afterwards.		
	What task I perform?	Nilish:task:09eu do, try to be precise	Tell us which main steps you p while doing this task?	Consider a studen media department has taken an initia professors of differ	Your assistant t who works part time in the of the university. His department tive where lectures from different rent departments will be		

Figure 3.20: Actual stakeholder tasks list

Analyst's view

Analyst's view that is *Vision analytics* is responsible for displaying analytics to the analysts. Whatever data stakeholders enter can not be displayed as it is as it is a mixture of various concepts and captured as a series of answers to questions. For it to be useful to the analysts, it has to be first processed to segregate meaningful information chunks.

Analyst's goal would be to have all the information readily available that will help them identify user types and the information that will help them build Personas. *User types* will help them to prioritize the improvements suggested by the users. To build Personas, they will need assistance in identifying and creating *Behavior variables* [11]. Additionally a list of tasks performed by the users and their motivations behind them would also be helpful. Keeping this in mind, in Vision analytics we have showed the *task list, feature list, profiles, and behavior variables*.

Task list section essentially lists all the tasks all the participating stakeholders have created. This list is sortable according to the frequency of performance, it's importance, importance of the improvement as suggested by the stakeholders, the reason stakeholder mentions for the improvement. Analysts can also combine tasks which they think have the same goal or which are nothing but similar; just reported by different stakeholders to form a *Feature*. Additionally it is also possible to open a specific task from the list to explore all its details. Figures 3.21 and 3.22 show mockup and the actual screenshot of the task list. As can be seen in figure 3.22, task list can be sorted according to different criterion.

Tasks	Features	Profiles Behavi	or variables	
			Ad	d to feature
Select	Task	Importance 🕈	Frequency	Feature \$
	Task one	Very Imp	Daily Once	1
	Task two	Rather Imp	Daily Once	2
С	Task three	Very Imp	Weekly Once	
3	Task four	Very Imp	Monthly Once	2

Figure 3.21: Analyst task list mockup

☆ Vision	Backlog						IM	PRESS 🕜
			TASKS FEATUR	ES USERS BEH	IAVIORAL VARIABLES			
								ADD TO FEATURE
Select	Id	Title	How often	How important	How important is improvement	Why important is improvement	Feature	Stakeholder
	1	Nitish task one	Daily once	Extremely important	Rather important	Saves time	1	1
	2	Nitish task two	Couple of many times	Extremely important	Extremely important	Saves extra efforts	1	1
	3	Mihir task one	Weekly once	Rather important	Less important	Saves time	1	2
	4	Mihir task two	Daily once	Extremely important	Extremely important	Saves extra efforts	1	2
Total tasks: 7							Limit: 4	Page: 1

Figure 3.22: Actual analyst task list screen

The feature list section lists all the features analysts have created. They can also quickly take a look at the tasks those belong to the same feature. This can be seen from following figures 3.23 and 3.24.

A Web Page					
Tasks	Featu	res	Behavior variables		
Creat	e new fe	eature	Add to exis	tingfeature	
	Select	Feature	Tasks inncluded 🗢	1	
	_	Feature one	1 & 2		
	۲	Feature two	3		
		Feature three	4		
Save					
				4	

Figure 3.23: Analyst feature list screen

♠ Vision Backlog							IMPRESS	0
		TASKS	FEATURES	USERS	BEHAVIORAL VARIABLES			
	1							
Choose one	Id			You	r comment	Tasks included		
	1			Sam	ple feature one	1, 2, 3, 4,		
						u	mit: 8 P	Page: 1

Figure 3.24: Analyst feature list screen actual

Profiles section lists all the stakeholder profiles. Analysts can take a look at the list of all the available profiles. This list is sortable according to the attributes chosen to reflect stakeholder's personal attributes i.e. questions 8 to 13 in figure 3.3. Each profile can be opened to explore all the details.

Behavior variables section displays a few pre-selected behavior variables. Following variables are considered:

Age: it will tell analysts how many stakeholders belong to specific age ranges, it is however not useful to derive any information about user types

Education: stakeholders will be plotted against the education level they have i.e. vocational, diploma etc. Higher education has positive impact on the user types i.e.

users with higher education tend to be innovators or early adopters [32].

Designation: stakeholders will be plotted against different designations like manager, assembler etc. Designation or higher social and economical status has positive impact on the user types i.e. users with higher designation tend to be innovators or early adopters [32].

Importance: for each feature, corresponding tasks are plotted against the scale of importance to see how many of them fall into different importance ranges i.e. Very important, less important etc. This will give hints about which tasks must deeply be investigated for their better realization in the proposed solution.

Frequency: for each feature, corresponding tasks are plotted against the scale of frequency of performance to see how many of them fall into different ranges i.e. Daily once, weekly once etc. This will give hints about which tasks must deeply be investigated for their better realization in the proposed solution.

Figures 3.25 and 3.26 show mockup and actual screenshot of how these behavior variables are displayed to the analysts. The horizontal line holds different values for a specific behavior variable i.e. for education it holds values like Diploma, Bachelor etc. For each such value, users which satisfy that value are plotted on top of the line. E.g. users with ids 1, 2, 3, 4 all fall into age range of 18-25 etc. One more analytic is provided which looks like *How many:* which tells how many users fulfill this criteria e.g. total 4 users fall into age range of 18-25.



Figure 3.25: Mockup behavior variables

🔒 Vision Backlog								IMPRESS	s 🕐
		TASKS	FEATURES	USERS	BEHAVIORAL VAR	RIABLES			
Age					Education				
How many: 5 User 1 User 2 User 3 User 4	How many: 0	How many: 0			How many: 0	How many: 0	How many: 2 User 2 User 5	How many: 2 User 1 User 3	How many: 0
User 5					Vocational	Diploma	Bachelor	Master	Doctor
18-29	30-54	55+							

Figure 3.26: Actual behavior variable screen

These variables are however just a starting point, analysts have to further identify other variables which are going to help them. In general, these analytics must indicate analysts which tasks have potential to improve considering their intended goals and considering the current way of performing them. It can also be brought to notice if any tasks which are not given importance in the current system, actually with suggested little improvements can make huge impact in terms of money or efficiency etc.

3.4 Summary

This chapter proposed a classification criterion observed from literature and helpful to Vision backlog. Techniques described in subsection 2.1 were further classified according to this criteria. A subset of the techniques those fulfill this criteria are further considered in section 3.3 to build the actual software.

This chapter also described all the important technical details of Vision backlog. At the beginning of the section, UI element choices are described which helped to obtain structured data. To model the behavior of the system behavioral UML diagram like use case diagram and structural UML diagram like class diagram are included. Enhanced Entity Relationship (ERR) diagram created through MySQL workbench is also included. The section also described the framework choices and the general architecture of Vision backlog. It gave details of both frontend and backend with appropriate code snippets from the actual code. It further explained how this application can be used, what all functionalities are available along with mockups and actual screenshots of the application.

Following chapter 4 explains how analysts can leverage from the analytics generated by Vision analytics, how the can understand stakeholder goals, see behavior patterns and build Persona.

Chapter 4

Usage Guidelines

4.1 Introduction

A software is not helpful if the user does not know how to use it. Vision backlog leverages from many concepts like SWOT analysis, Persona etc. It is important that the stakeholders are navigated effortlessly through the software without realizing which technique they are using. Stakeholders who are the end users of the system must not have to learn or understand these concepts as it is an overhead for them. All they need to understand is how to use the application. In-app help is provided to guide them through the entire process. Users of Vision analytics are analysts and they have to understand at least the interpretation of the answers and generated analytics. Those concepts are also described in the application, along with wiki on how to interpret the received answers, see application screenshots in appendix B.4.

Subsection 3.3.7 already explained in brief how the data created by stakeholders is going to help analysts. This section explains the generated analytics and how they can be used to generate Persona and to identify adopter category for the Persona. Section 4.2 lists the answers that stakeholders can give to questions and along with their meaning and interpretation. Section 4.4 explains how *behavior variables* those are provided will help to see behavior patterns. Further the section 4.3 explains how from the received answers in user profiles and from the behavior variables we have chosen, it is possible to guess the types of users. At last, section 4.5 explains how analysts can build Personas from the generated behavior patterns and other relevant data.

4.2 Interpretation of the received answers

For each question, different stakeholders can give different answers. Some questions expect descriptive answers whereas there are many questions which expect objective answers having multiple choices to choose from. Following figures 4.1 until 4.3 consolidates all possible answers to each question and explain their meaning. Each answer signifies a value and conveys a specif information. Answers for the questions starting from 1 until 16 provide us information about the possible user type or adopter category user belongs. Answers to the questions starting from 17 until

34 provide us information about the tasks stakeholders perform, their motivations behind it, and about surrounding context.

	Nr	Question	Answer	Inference
	1	My name	-	-
	2	My age	-	-
	3	What education I have	One of the following: 1. Vocational 2. Diploma 3. Bachelor 4. Master 5. Doctoral	Users with higher education tend to be early adopters
	4	My short story	-	-
	5	I work in this company as	One of the following: 1. Assembler 2. Supervisor 3. Manager	Users with higher education tend to be Innovators and early adopters
-	6	My interests	-	-
	7	My technical skills	One of the following: can use- 1. Computer 2. Smartphon e 3. Touch devices 4. Apps	Innovators and Early adopters have more exposure to mass media and communication channels
	8	l am a rational thinker	One of the following: 1. True 2. False	Innovators and Early adopters tend to be rational thinkers
-	9	I like to welcome good new changes	One of the following: 1. True 2. False	Innovators and Early adopters tend to appreciate innovation and changes
	10	I like to embrace science	One of the following: 1. True 2. False	Innovators and Early adopters have more favorable attitude towards science

Figure 4.1: Questions and interpretations - part 1

Nr	Question	Answer	Inference
11	I am social	One of the following: 1. True 2. False	Innovators and Early adopters have more social participation and are highly interconnected in social systems
12	I am cosmopolitan	One of the following: 1. True 2. False	Innovators and Early adopters tend to be more cosmopolitan
13	I seek latest information	One of the following: 1. True 2. False	Innovators and Early adopters tend to seek information about innovations more actively
14	This is what is exciting about my job	-	Innovators and Early adopters have strong opinion leadership
15	This is what I don't like about my job		
16	How I see myself in 5 years	-	Innovators and Early adopters have greater aspirations than others
17	What task I perform	-	-
18	How often do I perform it	 One of the following: 1. Couple of times a day 2. Daily once 3. Weekly once 4. Monthly once 5. Less frequently 	When combined together it can give hints about the relationship between the importance and frequency of performance of specific tasks. There could be certain tasks which are performed repeatedly but are of
19	How important it is	One of the following: 1. Extremely 2. Rather 3. Less	optimized and vice versa
20	What role I am playing while performing this task	-	-
21	What steps I perform while doing this task	_	Looking at the list of steps and the underlying goal, it could be possible to get rid of certain steps for efficiency
22	Why do I perform this task	-	This directly gives the motivation or goal behind performing the task

Figure 4.2: Questions and interpretations - part 2

Nr	Question	Answer	Inference
23	Why is it important to perform this task	-	This also gives underlying goal and the business driver behind performing the task
24	What tools you need to perform this task	-	Looking at the list of tools, it could be decided if they should be investigated to figure out their competency taking into consideration the task and the underlying goal
25	Do you need any special training to perform this task	One of following: 1. Yes 2. No	It gives additional contextual information
26	What kind of training you need to perform this task		
27	Can something go wrong while performing this task		
28	What are the effects of performing this task		
29	Do you use any special vocabulary related to this task		
30	Who else is involved in performing this task		
31	What should happen before and after performing this task		
32	Can you think of any improvements to this task		Valuable suggestions suggested by experienced employees which should be compared with the probable user type the user fits in. If he is a laggard and still suggesting improvements, then perhaps they could be worth investigating
33	How important you think is this improvement	One of the following: 1. Very imp 2. Rather imp 3. Less imp	Improvements marked very important could be worth investigating, if reporting user is laggard then even more
34	Main benefit of this improvement	One of the following: 1. Saves money 2. Saves time 3. Saves efforts	-

Figure 4.3: Questions and interpretations - part 3 $\,$

4.3 How to identify user types

The adopter categories as mentioned in section 2.2 can be identified with a set of characteristics; few important of which are considered and used as questions from 8 until 13 of figure 3.3. Apart from these 6 characteristics there are many others which Rogers suggests, but these 6 have been observed to make favorable distinction with a large margin (more than 50% of the observed studies support generalization that these attributes contribute positively to someone being Innovator or Laggard [49], and that's why they are chosen to be considered in Vision backlog). E.g. 74 percent of the research studies suggested that Education has favorable impact on someone being Innovator. Following table arranges the characteristics along with percentage of research studies supporting them.

Question	Percentage in support
Achievement motivations	61
I work in this company as/role	68
Mass media exposure	69
I am social	73
What education I have	74
I like to embrace science	74
I like to welcome good new changes	75
I am cosmopolitan	76
I am a rational thinker	79
I seek latest information	86

Table 4.1: User attributes and supporting percentage

Looking at the Table 4.1 analysts can roughly say that if a user says *I am a rational thinker*, then it boosts the chances that he could be an *Innovator*. Along with these 8 characteristics, analysts must also look the the technical skills a user has; Innovators and Early adopters tend to use social and digital medium enthusiastically than others. Also, if a user seems to have greater aspirations at the workplace, it also makes him a potential candidate to be an Innovator or Early adopter.

User type or Adopter category is an independent concept from Persona and they can be used independently to make different types of decisions. However they can also be combined if the analysts wish. If a user appears to be a Laggard and he suggests some improvement in the existing workflow or the way a certain task is performed, then it must really be worth investigating. Similarly, users who are potentially Innovators would be having creative suggestions for improvements.

4.4 How behavior variables can help

As explained earlier in subsection 3.3.7, it is possible to identify and create as many behavior variables as analysts can think of. We have presented a few as examples in my solution. It is a general practice to identify around 15 to 20 such variables [11]

so that clear behavior patterns can be seen. As examples we have presented Age, *Designation*, *Education*, *Importance* and *Frequency* as behavior variables.

- 1. Age: Age do not have any effect on user's innovativeness [32], this variable however shall give us the age distribution of our employees.
- 2. **Designation**: Innovators and early adopters tend to have higher social and economical status [32]. This variable works in conjunction with education. For example, if a particular user appears on the higher side of designation, education and if he is a rational thinker, open to change and science, he is likely to be a innovator or early adopter. If a specific user has less education and is at the bottom of designation hierarchy, is not equipped with latest communication channels or not social, he is likely to be a Laggard. If a Laggard suggests certain improvement to any current task with high importance, it could be worth investigating by analysts.
- 3. Education: Education has similar impact on user types as designation. People with higher education tend to be innovators or early adopters. This works in conjunction with designation and other same user attributes as mentioned in designation.
- 4. **Importance**: Analysts can choose a feature from a list. Tasks belonging to a specific feature are plotted against their reported importance. If for a specific feature more tasks are plotted on the side of very important, then analysts can roughly assume that this feature is important and must be investigated.
- 5. **Frequency**: Frequency also works like Importance, here the tasks in a specific feature are plotted against their frequency of performance. There could be tasks with similar goal but performed by different stakeholders, analysts can study their frequency of performance by different stakeholders to improve existing workflow.

Consider following figure 4.4 which shows sample behavior variables and users are plotted against different values, arrangement of elements in the figure is already explained in subsection 3.3.7:



Figure 4.4: Example behavior variables

From above figure, analysts can roughly guess that User 1 could be an **Innovator** as he has higher education and is working at a higher designation.

4.5 How to build Personas

Personas are built from the observed behavior patterns and associated goals. Behavior patterns can be observed from the set of behavior variables identified and by plotting our users against them. From the data we have collected, user goals can be synthesized by looking at goal oriented questions. We now have user goals and significant behavior patterns, we are just required to associate them to form a Persona. Below, we have outlined a sample Persona which will serve as a reference.

To build this Persona we have assumed that total 10 users- user 1, user 2,..., user 10 used *Vision backlog* and they entered following dummy data as shown in table 4.2. First column shows the attribute, second column shows a specific criteria chosen and the last column shows which users fulfill that criteria:

Attribute	Criteria	Which users
Education	Higher or equal to Bachelor	2, 3, 7, 9
Designation	Higher or equal to Manager	2, 3, 5, 9
Technical competency	Those are able to use Smartphones and Internet efficiently	1,3,4,7,9
Rational thinkers and latest information	Those who reacted with positive answers	2, 3, 4, 7, 9
Social and Cosmopolitan	Those who reacted with positive answers	1, 2, 4, 6, 7, 8
Aspirations	Those who were ambitious	3, 4, 6, 7, 8, 9

Table 4.2: Behavior pattern

From this table Users 2, 3, 4 and 9 appear to fulfill multiple criterion and that suggests a pattern. These criterion makes them potential candidates to be Innovators or Early adopters as well. From this data we could build a Persona as shown in following figure 4.5:

	Age	35
Monica Williams	Education	Masters
Monica is a CEO of a software company. She finished her masters in business	Role	Manager
her own company. She had been innovative since childhood and took interests in science and designs. She loved to solve problems as kid. She was	Main Responsibilities	Operseeing operations
always attracted to new devices- mobiles were in their early days and laptops yet scarce. However, when her parents got ber one for her, soon she	Expert in	Business communication, business negotiations
was able to write her first computer program. She grew up in a cosmopolitan part of the city which exposed her to different types of people. She is very	Technical competency	Can use smartphone, touch devices, Internet with ease
active on Internet and social media and is passionate about growing her business even larger.	Probable user type	Innovator

Figure 4.5: Example Persona with filled in details

This Persona represents a group of people who have similar needs, goals and motivations. Analysts could designate this Persona as their *Primary Persona* if the
market research suggests that this Persona resembles the target user group. This Persona would then serve as a communication tool to stay within certain boundaries when it comes to take a certain design decision.

4.6 Summary

This chapter explained how analysts are going to help from the analytics generated in Vision analytics. It explained what can they interpret from the stakeholder answers. It also explained how they can identify probable adopter categories of the stakeholders. Further it described how they can identify behavior patterns from provided behavior variables. The last part demonstrated how they can build a Persona from the data entered by stakeholders.

Following chapter 5 explains how Vision backlog is evaluated by stakeholders and analysts, and includes evaluation results with its interpretation.

Chapter 5

Evaluation

5.1 Introduction

To make sure that the concept of Vision-backlog is a viable one, that it meets its stated purpose and that it really helps to solve the identified problems, it is important to take a survey. The survey must tell us that the software solution that we have built is a usable one by stakeholders and that the content that is generated and the way it is presented is helpful to the analysts.

Total 6 people were asked to use Vision backlog and to enter the data. They were then asked to take a usability survey as described in following section 5.2 whose results are included in section 5.4 bellow. Total 4 participants were asked to use Vision analytics and they subsequently took the content quality survey as described in the same subsection.

Section 5.2 explains what type of surveys are taken and their purpose in evaluating Vision backlog. Section 5.3 lists sample questions those were asked as a part of the surveys. Section 5.4 discusses the evaluation results.

5.2 Evaluation setup

As a part of the evaluation, total two surveys were taken:

1. On the usability of the software solution:

For this **AttrakDiff** [3]- an online survey tool built on scientific findings [26] was used. AttrakDiff is a tool that measures the attractiveness of an interactive system [3]. The set of questions asked are listed in subsection 5.3 and the result of this evaluation is included in section 5.4. This survey was taken for frontend application- Vision backlog.

AttrakDiff provides total three types of surveys.

(a) Single evaluation: A single application is evaluated by participants against it pragmatic qualities and hedonic qualities (see bsection 5.4 for details about it). It tells us if the product is desired by the users or there is still room for optimization.

- (b) Comparison A-B: It is similar to single evaluation except for two products are evaluated at the same time and then results are compared.
- (c) *Before-After*: In this type of survey, the same product is evaluated twice and then results are compared to get a quick feedback on implemented improvements

For Vision backlog *Single evaluation* is used as there is only one product and we need a feedback on its usability. It was an online survey, participants were invited via email to participate. The duration was one month so that participants can evaluate it as per their convenience. None of the participants were requirements analysts.

2. On the quality of the content:

This survey is taken for frontend application- Vision analytics. For this a short survey using Google Forms was created. Results of this evaluation is included in subsection 5.4. This survey was also taken online and participants were invited via email. This survey was also active for a duration of a month. The participants were computer scientists only, who had some experience with requirements elicitation.

5.3 Questionnaire

For the first survey, the default questionnaire which AttrakDiff provides was used. For second survey a set of questions were designed independently.

Questionnaire for the feedback on usability

Few of the questions those were asked are shown in the following figure 5.1, entire set of questions are included in Appendix.

human*	\bigcirc	technical						
isolating*								connective
pleasant*	\bigcirc	unpleasant						
inventive*								conventional
simple*	\bigcirc	complicated						
professional*								unprofessional
ugly*	\bigcirc	attractive						
practical*								impractical
likeable*	\bigcirc	disagreeable						
cumbersome*								straightforward

Figure 5.1: Set of question - 1

Questionnaire for feedback on content quality

Few of the questions those were asked are shown in the following figure 5.2, entire set of questions are included in Appendix.

The concepts Vision-analytics uses i.e. Persona, behavior variables are explained properly



Figure 5.2: Set of question for obtaining content quality - 1

5.4 Evaluation results

Results for both surveys were obtained independent of each other. They give us information about different characteristics of the application.

Subsection 5.4 explains the terms those are used in the evaluation result with AttrakDiff. It further explains the final results. Subsection 5.4 explains the results of the evaluation taken to measure the quality of the content generated i.e. of Vision analytics.

Results of usability evaluation

For usability evaluation total 6 participants were invited out of which 5 actually participated. The results of the evaluation are shown in diagrams 5.3, 5.4 and 5.5.

The meaning of the terms used in the diagram [3]:

- *Pragmatic Quality (PQ)* indicates how successful users are in achieving their goals with the product
- *Hedonic Quality- stimulation (HQ-S)* indicates to what extent the product supports human needs of developing something new in terms of novel, interesting and stimulating functions, contents, interaction and presentation styles
- *Hedonic Quality- identity (HQ-I)* indicates to what extent the product allows user to identify with it
- *Attractiveness (ATT)* is the global value of the product to which Hedonic and pragmatic qualities contribute equally

The values of hedonic quality are represented on the vertical axis (bottom = low value). The horizontal axis represents the values of the pragmatic quality (left = low value). The product lies within one or more of the character-regions i.e. neutral, self-centered etc. More towards the left and top your product lies, more it is desired. Bigger the confidence rectangle, less sure one can be about evaluation. Figures 5.3, 5.4 and 5.5 depict the results of the evaluation users did.

Figure 5.3 indicates that in general Vision backlog is rated on the positive side. However, the confidence rectangle surrounding it is relatively big and spans across regions on both the axis. This happened because the participants evaluated the application differently. It means we can not totally be sure to which region our product belongs. The product i.e. Vision backlog appears more self centered than desired and there is still a room for optimization.



Portfolio-presentation

Figure 5.3: Usability evaluation- Portfolio of results

In figure 5.4 the average values of above mentioned dimensions for Vision backlog are plotted. In this representation, hedonic quality distinguishes between the aspects of simulation and identity. Furthermore the rating of attractiveness is presented [3].

It indicates that the overall impression of the product is very attractive, it also succeeds to stimulates and bind your users closely to the product. However, there is still room for improvement while users are not completely sure if their goals are achieved.



Diagram of average values

Figure 5.4: Usability evaluation- Diagram of average values

Figure 5.5 represents the mean values of the word-pairs. Left hand side word pairs show the scale of choices participants had to choose from. They are divided into categories as mentioned earlier. The scale bellow displays the number of options participants had i.e. total 7. The extreme values should be of interest as they show which characteristics are particularly critical or well-resolved [3]. It shows that on an average the users have rated the product on the positive side for almost all the qualities.



Description of word - pairs

Figure 5.5: Usability evaluation- Description of word pairs

Results of usefulness evaluation

Total 4 participants were invited for this evaluation out of which three participated. Following figures 5.6, 5.7, 5.8, 5.9, 5.10 show different questions asked and answers received for each of them. Each question had four options to choose from i.e. strongly disagree to strongly agree which are numbered from 1 until 4 respectively. Figures 5.6, 5.7, 5.8, 5.9, 5.10 are pie charts. The corresponding question is mentioned at the top of the figures. Colored regions indicate the number of the choice and percentage of users who chose that option.

Figure 5.6 indicates that, 66.7% participants agree that concepts like Persona are explained properly in Vision analytics. 33.3% participants strongly agree on it.



The concepts Vision-analytics uses i.e. Persona, behavior

Figure 5.6: Usefulness evaluation-1

Figure 5.7 indicates that, 66.7% participants strongly agree that Vision analytics also explains how concepts like Persona can be used to elicit requirements.



The application explains how the concepts used can be used for eliciting requirements

Figure 5.7: Usefulness evaluation-2

Figure 5.8 indicates that, 66.7% participants strongly agree that Vision analytics appropriately aggregates data. Which means they found it easier to find stakeholders tasks, stakeholders profiles or behavior variables in the application so that they can process it i.e. sort it, merge it etc.



The semantic segregation of the data is helpful

Figure 5.8: Usefulness evaluation-3

Figure 5.9 indicates that, 100% of the participants strongly agree that the analytics provided in Vision analytics are helpful to their elicitation activities.



The analytics provided are helpful

Figure 5.9: Usefulness evaluation-4

Figure 5.10 indicates that, only 33.3% of the participants strongly agree that using list to represent analytics i.e. to show stakeholder tasks, profiles was a good decision. Others slightly disagree with it.



The structures used i.e. lists etc. to represent the analytics are helpful

Figure 5.10: Usefulness evaluation-5

On an average participants evaluated and rated the product i.e. Vision analytics positively indicating that the analytics application could be a helpful tool for them.

5.5 Summary

This section described how Vision backlog is evaluated to ensure its viability and usability. It explained what type of surveys were taken and how they were taken. It described their significance in evaluation. It included sample questionnaires of the surveys. It further described the results of both the surveys with the help of graphs and figures. These graphs and figures are also explained i.e. meaning of terms used etc.

Chapter 6

Conclusion

As correct, complete and well understood requirements play a vital role in building a successful software product, it is important that we do the elicitation correctly, effectively and efficiently. For elicitation to be effective; understanding end user goals, or in case of enterprise application understanding stakeholder goals along with surrounding context is important. Experienced analysts use a set of techniques to extract such goals and to understand underlying context. However, there is no standardized set of techniques or a procedure that can be followed by all the practicing analysts. Vision backlog suggested that various concepts like *Goal-directed design*, *Adopter categories*, *Persona* and techniques like *SWOT analysis*, *Theory of change* from different disciplines can be combined with traditional elicitation techniques to give better results as compared to using conventional elicitation methods and approaches alone.

Vision backlog also takes into consideration current challenges in requirements elicitation. Rigid time constraints for stakeholder meetings and psychological factors like people's moods limit effectiveness of employed elicitation process and used tools or methods. To tackle this problem, Vision backlog proposes to build a web application that can be used by stakeholders anytime and from anywhere to do the elicitation themselves. Vision backlog leverages from above mentioned concepts and combines them with popular elicitation techniques in such a way that stakeholders can share the responsibility of eliciting requirements effortlessly and effectively with analysts. Vision backlog also helps analysts by generating analytics for them where they can explore all the stakeholders goals and behavior patterns to be able to build Persona.

To achieve this, popular requirements elicitation techniques employed by practicing requirements analysts were studied. Motivation behind studying those techniques was to classify them in such a way that a subset can be chosen to be used in Vision backlog. These techniques help to locate and scope a problem by exploring surrounding context. Vision backlog proposes a comparison criteria that helps to qualify a technique to be used in Vision backlog.

A set of shortlisted techniques based on the proposed comparison criteria are used in a web application for stakeholders to enter their goals and surrounding information. For that, a set of questions is proposed. Each question corresponds to a specific elicitation technique and extract a specific information. To obtain structured answers to be able to process further, appropriate UI elements are also suggested. Questions are segregated in a logical way so that a smooth sequence can be followed by stakeholders while answering them.

The evaluation results give us confidence that the concept of *Vision backlog* solves the problem effectively. The software built to tackle the identified problem lets stakeholders state their goals and needs effortlessly. It guides them through various techniques without having them to learn any of them. Web application provides help sections wherever stakeholders might find themselves clueless about current or next steps. The web application maintains a flow that is easier to follow. It also helps analysts to understand end user, stakeholder and business goals and provides important contextual information for tasks that stakeholder performs to achieve a specific goal. The concepts used in Vision analytics are explained properly and are helpful. Both the applications have clear navigation structures that lets their respective users to use smoothly. Vision analytics helps requirements analysts to locate and scope problems during initial stages of software development.

Vision backlog web application is a usable one and is rated to be almost desirable by participating stakeholders. The overall impression of this web application is attractive, however it needs improvements regarding both hedonic and pragmatic qualities. For Vision analytics, participants reported that it is a useful application, however the UI elements used to display analytics should be improved.

Chapter 7

Future work

Vision backlog concentrated on requirements elicitation in general without focusing on any specific software development methodology. As the concept Vision backlog itself is agile in nature i.e. it encourages stakeholders to iterate over their own vision repeatedly to refine it, it would be interesting to figure out and demonstrate how such an approach can be extended and integrated into an Agile methodology (e.g. into Scrum). By doing this, stakeholder needs can be managed and monitored since their inception until they are developed, typically in a span of a sprint. This would help to measure stakeholder involvement into elicitation activities and can give feedback on how much rework is avoided in the presence of Vision backlog. Same approach can also be used for reducing technical debt. The general idea is depicted in following diagram:



Figure 7.1: Future work

Vision backlog would serve as a starting point which will be used to build a broader product vision and through which potential tasks concerning a specific product feature can be extracted. Such tasks could potentially correspond to tasks in tools like Jira. This should form initial product or even a smaller sprint backlog. Once the development work is over for a sprint, a retrospection should provide feedback to Vision backlog so that stakeholders or end users can refine their vision if the developed solution do not fit closely enough to their ideas. Ideally, rework costs while practicing scrum without Vision backlog should be higher than when Vision backlog is integrated into scrum.

Glossary

- **repository pattern** The repository pattern is intended to create an abstraction layer between the data access layer and the business logic layer of an application. It is a data access pattern that prompts a more loosely coupled approach to data access. We create the data access logic in a separate class, or set of classes, called a repository with the responsibility of persisting the application's business model [48]. 52
- requirement engineer The requirements engineer is in charge of working with the project stakeholders and end users to elicit, understand, analyze, and document the requirements for a system in order to solve a given business problem. Other common titles for this role are: Requirements Analyst, Business Systems Analyst, Business Analyst (generic term), etc. [50]. 13
- stakeholders According to the Project Management Institute (PMI), the term project stakeholder refers to, 'an individual, group, or organization, who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project' (Project Management Institute, 2013). ISO 21500 uses a similar definition [24]. In this thesis however this term to refer to all the members of the project excluding requirements engineer/analyst. 14

Appendix A

Interview questions

A.1 Interview questions

A.1.1 Usability survey

stylish*	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	tacky
predictable*								unpredictable
cheap*	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	premium
alienating*								integrating
brings me closer to	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	separates me from
people*								people
unpresentable*								presentable
rejecting*	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	inviting
unimaginative*								creative
good*	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	bad
		Figure	A.1: S	et of qu	estion -	2		
confusing*	\bigcirc	Figure	A.1: S	et of qu	estion -	2	0	clearly structured
confusing* repelling*	0	Figure	A.1: S	et of qu	estion -	- 2 0	0	clearly structured appealing
confusing* repelling* bold*	0	Figure	A.1: S	et of qu	estion -	· 2 ○ ○	0	clearly structured appealing cautious
confusing* repelling* bold* innovative*		Figure	A.1: S	et of qu	estion -	2 0 0		clearly structured appealing cautious conservative
confusing* repelling* bold* innovative* dull*		Figure	A.1: S	et of qu	estion -	2 0 0 0		clearly structured appealing cautious conservative captivating
confusing* repelling* bold* innovative* dull* undemanding*		Figure	A.1: S	et of qu	estion -	2 0 0 0 0		clearly structured appealing cautious conservative captivating challenging
confusing* repelling* bold* innovative* dull* undemanding* motivating*		Figure	A.1: S	et of qu	estion -	2 0 0 0 0 0 0 0 0 0 0 0		clearly structured appealing cautious conservative captivating challenging discouraging
confusing* repelling* bold* innovative* dull* undemanding* motivating* novel*		Figure	A.1: S	et of qu	estion -			clearly structured appealing cautious conservative captivating challenging discouraging ordinary

Figure A.2: Set of question - 3

A.1.2 Survey on content quality

The application explains how the concepts used can be used for eliciting requirements



Figure A.3: Set of question for obtaining content quality - 2 The semantic segregation of the data is helpful

	1	2	3	4	
Strongly agree	0	0	0	0	Strongly disagree

Figure A.4: Set of question for obtaining content quality - 3 The analytics provided are helpful

	1	2	3	4	
Strongly agree	0	0	0	0	Strongly disagree

Figure A.5: Set of question for obtaining content quality - 4

Appendix B

Actual application screenshots

B.1 Sample screenshots of the application

1	Vision Backlog				0	θ	€
	the second			Alexandre de la compañía de la compa			
1.14					×		
		1 That's me!	2 What am I like?	i am at my workplace!			
		The information that you enter in you can handle. How and why you This help us understand what kin For example, should the proposed	this and next two tabs tell us what is it like to be 'You'l What are your infl u are involved in this job, what you like about it, what you don't. We keep th d of user you are, which in turn makes easier for us to take certain decis 5 system use touch controls etc.	erests, what kind of latest technologies is information confidential.			
		l am	Niţish:ame? Dummy name is fine! 28 _/ age				
		l have	Vocational training Diploma Bachelor *	Masters Doctoral			

Figure B.1: Stakeholder's view - Profile

ISION BACKIOG						0	0
							x
Quick Intro Give us few short answers		2 Think about it Why is it important	3	Supporting details Let's uncover some important details	Finishin Few more to voice y	g details e details our opinion	
The information that you e	nter in this and next two tabs giv	res us all the important details ab	bout the task that you perform. It hele	lps us understand why you need to	perform it, how oftenly you perform it a	and how	
mportant it is. Also, It give any other ways of perform hrough everyday experier	s us more detailed information lil ng this task which could be more ce on the floor.	ke- if there is any special traning efficient. The entire information v	required to perform this task or any s will help us improve the existing work	special vocabulary you use. Apart fr kflow taking into consideration your v	rom this, it gives you an opportunity to valuable suggestions- those you have a	suggest acquired	
important it is. Also, It give any other ways of perform through everyday experier The understanding we dev ocused that improves effic	s us more detailed information lin ng this task which could be more ce on the floor. elop from this information can gi iency, etc. Take your time but fill	ee- if there is any special traning efficient. The entire information v ve us hints like- whether any acti- this information in detail. You can	required to perform this task or any will help us improve the existing work ivities that you perform as a part of th a always add/edit this information afte	special vocabulary you use. Apart fr kflow taking into consideration your v his task can be dropped or redesigne erwards.	om this, it gives you an opportunity to valuable suggestions- those you have a ed or perhaps other important activitie:	suggest acquired s can be	
important it is. Also, it give any other ways of perform through everyday experier The understanding we dev focused that improves effic	s us more detailed information in ng this task which could be more ce on the floor. elop from this information can gr ency, etc. Take your time but fill	ke- if there is any special traning efficient. The entire information v ve us hints like- whether any acti- this information in detail. You can	required to perform this task or any will help us improve the existing work writes that you perform as a part of th always add/edit this information after	special vocabulary you use. Apart fr kflow taking into consideration your v his task can be dropped or redesign envards.	om this, it gives you an opportunity to valuable suggestions- those you have a ed or perhaps other important activities	suggest acquired s can be	
Important it is. Also, it give any other ways of perform through everyday experier The understanding we dev focused that improves effic what task I perform?	s us more detailed information in ng this task which could be more ce on the floor. elop from this information can ge iency, etc. Take your time but fill <u>Tell us what you do, t</u>	xe- if there is any special traning efficient. The entire information v we us hints like- whether any acti- this information in detail. You can this information in detail. You can y	required to perform this task or any ; will help us improve the existing work where that you perform as a part of th n always add/edit this information after fell us which main steps you perfor while doing this task?	special vocabulary you use. Apart fr ktlow taking into consideration your v his task can be dropped or redesign erwards. Consider a student department of the ur initiative where lect	om this, it gives you an opportunity to valuable suggestions- those you have a ed or perhaps other important activitier Your assistant who works part time in the media niversity. His department has taken res from different professors of	suggest acquired s can be	

Figure B.2: Stakeholder's view - Task creation

I VISION BACKING					0
			+	and the second	
al number of tasks are 2					
tish task one	1 Quick Intro Give us few short answers	2 Think about it Why is it important	3	Supporting details	Finishing details Few more details to voice your opinion
DIT.	The information that you enter in	this and followling two tabs gives us all the impo	rtant details about the task tha	it vou perform. It helps us underst	and why you need to perform it, how
	oftenly you perform it and how in use. Apart from this, it gives you	nportant it is. Also, It gives us more detailed infor	mation like- if there is any spec	cial traning required to perform th	s task or any special vocabulary you
tish task two	existing workflow taking into cons	sideration your valuable suggest any other ways of	nave acquired through everyda	and be more enclose. The entire	information will help us improve the
		,,,.,		,	
DIT	The understanding we develop f	rom this information can give us hints like- whe	her any activities that you per	form as a part of this task can b	e dropped or redesigned or perhaps
EDIT	The understanding we develop f other important activities can be to the state of th	rom this information can give us hints like- whe focused that improves efficiency, etc. Take your t	ther any activities that you per ime but fill this information in d	form as a part of this task can b etail. You can always add/edit this	e dropped or redesigned or perhaps information afterwards.
EDIT	The understanding we develop f other important activities can be t	tom this information can give us hints like- whe focused that improves efficiency, etc. Take your I	her any activities that you per me but fill this information in d	form as a part of this task can b etail. You can always add/edit this erform Y	e dropped or redesigned or perhaps information afterwards.
EDIT	The understanding we develop f other important activities can be What task I Nitis	trom this information can give us hints like- when focused that improves efficiency, etc. Take your t the task.oge u do, try to be precise with	her any activities that you per me but fill this information in d I us which main steps you pr lie doing this task?	form as a part of this task can b etail. You can always add/edit this erform Y Consider a student	e dropped or redesigned or perhaps i information afterwards. Dur assistant who works part time in the
EDT	The understanding we develop 1 other important activities can be what task 1 <u>Nitis</u> perform?	tom this information can give us hints like- whe focused that improves efficiency, etc. Take your t the task.copeu do, try to be precise with	her any activities that you per me but fill this information in d I us which main steps you pe lie doing this task?	form as a part of this task can b etail. You can always add/edit this erform Y Consider a student media department of has taken an initiat	e dropped or redesigned or perhaps information afterwards. Dur assistant who works part time in the if the university. His department ve where lectures from different

Figure B.3: Stakeholder's view - Task list

0

✿ Vision Backlog

Welcome to Vision-analytics!

This apps gives you analytics for the content that is created by project stakeholders. Click on 🏫 button to start working if you already have read this Help content.

Concepts used

Goal directed design process

Goal-directed design process tries to combine multiple techniques of elicitation which can be used to understand user goals. It gives solutions that meet the user goals or needs, and cater to the business and technical constraints at the same time. The process is roughly divided into six phases:

1. Research

• It is aimed at gathering and providing qualitative data about the potential and /or actual users of the product being built. It involves many techniques like observation, contextual interviews, competitive product audits etc. • As an outcome of employing the echniques we obtain Behavior patterns which tell us how the product will be used and what goals or motivation lie behind using the product. These Behavior patterns and associated goals with them give us Persona in the next phase.

With Theft give us performed in the first performance.
2. Modeling
• The data obtained in research phase is used to build domain and user models in modeling phase. Domain models plots insights about domain in simplified formates such as workflow or information flow diagrams.
• From the identified groupings of behaviors, attitudes, aptitudes or skills of the users Personas can be built. A specific Persona tells us what a specific group of users expect and how will they use the proposed product. By
prioritizing Personas we can very much prioritize what needs to be focused in the proposed product.

Figure B.4: Analyst's view - Help

🔒 Vision	Backlog						IMPR	ESS 🕜
			TASKS FEATUR	RES USERS BEH	HAVIORAL VARIABLES			
Name	Designation	Rational thinker	Attitude towards change	Attitude towards science	Is social	Is cosmopolite	Is information seeker	Stakeholder
Nitish	Supervisor	true	true	true	true	true	true	1
Mihir	Intern	true	true	true	true	true	true	2
Marcel	Manager	true	true	true	true	true	true	3
Neha		true	true	true	true	true	true	4
wolle	Intern	true	true	true	true	true	true	5
							Limit: 5	Page: 1

Figure B.5: Analyst's view - Profiles

Bibliography

- 10 Card Sorting Tools for Surveying Information Architecture (IA). https:// dynomapper.com/blog/19-ux/428-card-sorting-tools. [Online; accessed 20-Oct-2017].
- [2] Angular. https://angular.io/. [Online; accessed 26-Oct-2017].
- [3] Attrakdiff demo project. http://www.attrakdiff.de/files/demoproject_ results.pdf. Accessed: 2017-16-10.
- [4] N. L. Atukorala, C. K. Chang, and K. Oyama. "Situation-Oriented Requirements Elicitation". In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). Vol. 1. This paper states communication skills are important in requirements elicitation, verbal communication is not always enough, use of traditional approaches in RE is difficult, human perspective in elicitation is important, and discusses different approaches for elicitation. June 2016, pp. 233–238. DOI: 10.1109/COMPSAC.2016.191.
- [5] A. Batool et al. "Comparative study of traditional requirement engineering and Agile requirement engineering". In: 2013 15th International Conference on Advanced Communications Technology (ICACT). Jan. 2013, pp. 1006–1014.
- [6] Donald Bell. "UML Basics: The class diagram". In: IBM. [Online] IBM 15.09 (2004).
- [7] Brainstorming made simple. https://bubbl.us/. [Online; accessed 20-Oct-2017].
- [8] Glenn J Browne and Michael B Rogich. "An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques". In: Journal of Management Information Systems 17.4 (2001), pp. 223–249.
- [9] Pär Carlshamre and Joachim Karlsson. "A usability-oriented approach to requirements engineering". In: Requirements Engineering, 1996., Proceedings of the Second International Conference on. IEEE. 1996, pp. 145–152.
- [10] Nancy J Cooke. "Varieties of knowledge elicitation techniques". In: International Journal of Human-Computer Studies 41.6 (1994), pp. 801–849.
- [11] Alan Cooper, Robert Reimann, and David Cronin. About face 3: the essentials of interaction design. This book explains goal based process and personas. John Wiley & Sons, 2007.
- [12] Alan M. Davis. "Elicitation Technique Selection: How Do the Experts Do It". In: Great Software Debates. Wiley-IEEE Press, 2004, pp. 225-. ISBN: 9781119134657. DOI: 10.1002/9781119134657.ch26.URL: http://ieeexplore. ieee.org/xpl/articleDetails.jsp?arnumber=7304181.

- [13] A. Davis et al. "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review". In: 14th IEEE International Requirements Engineering Conference (RE'06). Sept. 2006, pp. 179–188. DOI: 10.1109/RE.2006.17.
- [14] Do it yourself. http://diytoolkit.org/tools/. [Online; accessed 19-April-2017].
- [15] Ramez Elmasri and Shamkant B Navathe. *Fundamentals of database systems*. Pearson, 2015.
- [16] Entity Framework Core. https://docs.microsoft.com/en-us/ef/core/. [Online; accessed 24-Oct-2017].
- [17] William Foddy. Constructing questions for interviews and questionnaires: Theory and practice in social research. Cambridge university press, 1994.
- [18] A. Jaramillo Franco. "Requirements elicitation approaches: A systematic review". In: 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS). This paper researches different approaches for requirements elicitation from past 25 years and reports the finding. Particularly interesting part is changing trend in sources of requirements and methods used for elicitation in past few years. May 2015, pp. 520–521. DOI: 10.1109/RCIS.2015.7128917.
- [19] Fay Fransella, Richard Bell, and Don Bannister. A manual for repertory grid technique. John Wiley & Sons, 2004.
- [20] Joseph Frantiska. "Use Case Diagrams". In: Visualization Tools for Learning Environment Development. Springer, 2018, pp. 1–8.
- [21] Rubén Fuentes-Fernández, Jorge J. Gómez-Sanz, and Juan Pavón. "Understanding the Human Context in Requirements Elicitation". In: *Requir. Eng.* 15.3 (Sept. 2010), pp. 267–283. ISSN: 0947-3602. DOI: 10.1007/s00766-009-0087-7. URL: http://dx.doi.org/10.1007/s00766-009-0087-7.
- [22] Joseph A Goguen and Charlotte Linde. "Techniques for requirements elicitation". In: Requirements Engineering, 1993., Proceedings of IEEE International Symposium on. IEEE. 1993, pp. 152–164.
- [23] Sue Greener. Business research methods. BookBoon, 2008.
- [24] Guidance on project management. https://www.iso.org/obp/ui/#iso: std:iso:21500:ed-1:v1:en. [Online; accessed 19-April-2017].
- [25] Jonathan Gutman. "A means-end chain model based on consumer categorization processes". In: *The Journal of Marketing* (1982), pp. 60–72.
- [26] Marc Hassenzahl, Michael Burmester, and Franz Koller. "AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität". In: Mensch & Computer 2003. Springer, 2003, pp. 187–196.
- [27] Ann M Hickey and Alan M Davis. "A unified model of requirements elicitation". In: Journal of Management Information Systems 20.4 (2004), pp. 65– 84.

- [28] Dennis Neil Hinkle. "The change of personal constructs from the viewpoint of a theory of construct implications". PhD thesis. The Ohio State University, 1965.
- [29] Robert R Hoffman. "The problem of extracting the knowledge of experts from the perspective of experimental psychology". In: *AI magazine* 8.2 (1987), p. 53.
- [30] M. Hosseini et al. "Configuring crowdsourcing for requirements elicitation". In: 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS). This paper talks about how crowd sourcing can be helpful in requirements elicitation and how it leads to a successful software system. May 2015, pp. 133–138. DOI: 10.1109/RCIS.2015.7128873.
- [31] JSON Web Token. https://jwt.io/. [Online; accessed 24-Oct-2017].
- [32] Veronika Karnowski and Anna Sophie Kümpel. "Diffusion of Innovations". In: Schlüsselwerke der Medienwirkungsforschung. Ed. by Matthias Potthoff. This book explains different types of users and their attributes. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, pp. 97–107. ISBN: 978-3-658-09923-7. DOI: 10.1007/978-3-658-09923-7_9. URL: http://dx.doi.org/10.1007/ 978-3-658-09923-7_9.
- [33] George Kelly. The psychology of personal constructs: Volume two: Clinical diagnosis and psychotherapy. Routledge, 2003.
- [34] Lucy Kimbell and Joe Julier. "The social design methods menu". In: London, United Kingdom: Fieldstudio Ltd (2012).
- [35] A. Knauss. "On the usage of context for requirements elicitation: End-user involvement in IT ecosystems". In: 2012 20th IEEE International Requirements Engineering Conference (RE). Sept. 2012, pp. 345–348. DOI: 10.1109/RE. 2012.6345835.
- [36] Walid Maalej and Anil Kumar Thurimella. *Managing requirements knowledge*. Springer, 2013.
- [37] N. Maiden et al. "Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding". In: 2010 18th IEEE International Requirements Engineering Conference. This paper talks about RE as creative problem solving, how emergence of new technologies has to be considered in RE, about Agile and RE together. Sept. 2010, pp. 57–66. DOI: 10.1109/RE.2010.16.
- [38] Material Design Lite. https://getmdl.io/. [Online; accessed 26-Oct-2017].
- [39] John McManus. "A stakeholder perspective within software engineering projects". In: Engineering Management Conference, 2004. Proceedings. 2004 IEEE International. Vol. 2. IEEE. 2004, pp. 880–884.
- [40] *mecanalyst.* http://skymax-dg.com/mecanalyst/. [Online; accessed 20-Oct-2017].
- [41] P. Nemutanzhela and T. Iyamu. "Theory of diffusion of innovation for analysis in information systems studies". In: Science and Information Conference (SAI), 2015. This paper talks about Diffusion of innovation in information systems. July 2015, pp. 603–608. DOI: 10.1109/SAI.2015.7237205.

- [42] Donald A. Norman. "Emotional Design". In: Ubiquity 2004. January (Jan. 2004), pp. 1–1. ISSN: 1530-2180. DOI: 10.1145/966012.966013. URL: http://doi.acm.org/10.1145/966012.966013.
- [43] OpenRepGrid. http://openrepgrid.org/. [Online; accessed 20-Oct-2017].
- [44] Alex F. (Alex Faickney) Osborn and 1923-1979 Bristol Lee Hastings. Applied imagination : principles and procedures of creative problem-solving. English.
 3rd. rev. ed. Includes index. New York : Scribners, 1979. ISBN: 0023895209.
- [45] David W Pickton and Sheila Wright. "What's swot in strategic analysis?" In: Strategic change 7.2 (1998), pp. 101–109.
- [46] POCO classes. http://www.c-sharpcorner.com/UploadFile/5d065a/ poco-classes-in-entity-framework. [Online; accessed 20-Oct-2017].
- [47] Klaus Pohl. Requirements Engineering Fundamentals, Principles, and Techniques. Springer, 2010. ISBN: 978-3-642-12577-5. URL: http://www.springer.com/computer/swe/book/978-3-642-12577-5?changeHeader.
- [48] Repository Pattern In ASP.NET Core. https://social.technet.microsoft. com/wiki/contents/articles/36287.repository-pattern-in-asp-netcore.aspx. [Online; accessed 20-Oct-2017].
- [49] Everett M Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.
- [50] Role: Requirements Engineer. http://www.modernanalyst.com/TheProfession/ Roles/RequirementsEngineer/tabid/188/Default.aspx. [Online; accessed 19-April-2017].
- [51] Bryce Ryan and Neal C Gross. "The diffusion of hybrid seed corn in two Iowa communities." In: *Rural sociology* 8.1 (1943), p. 15.
- [52] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.
- [53] A. Sutcliffe and P. Sawyer. "Requirements elicitation: Towards the unknown unknowns". In: 2013 21st IEEE International Requirements Engineering Conference (RE). This paper talks about the problems faced by poor RE, different RE techniques, also suggests combining approached with traditional interviews like prototyping. July 2013, pp. 92–104. DOI: 10.1109/RE.2013.6636709.
- [54] Swagger. https://swagger.io/. [Online; accessed 26-Oct-2017].
- [55] Dana H Taplin and Hèlene Clark. "Theory of change basics: A primer on theory of change". In: New York: Actknowledge (2012).
- [56] The repository pattern. https://social.technet.microsoft.com/wiki/ contents/articles/36287.repository-pattern-in-asp-net-core.aspx. [Online; accessed 20-Oct-2017].
- [57] The Standish Group- "The chaos report". https://www.standishgroup. com/sample_research_files/chaos_report_1994.pdf. [Online; accessed 20-Oct-2017].
- [58] Top 10 Software Failures Of 2011. https://www.businesscomputingworld. co.uk/top-10-software-failures-of-2011/. [Online; accessed 28-Oct-2017].

- [59] Vero Vanden Abeele and Bieke Zaman. "Laddering the user experience!" In: User Experience Evaluation Methods in Product Development (UXEM'09)-Workshop. 2009.
- [60] Web design and development. https://www.slideshare.net/auroracslk/ web-design-and-development-at-aurora. [Online; accessed 20-Oct-2017].
- [61] Michael Wilson et al. "Task models for knowledge elicitation". In: *Knowledge Elicitation: principles, techniques and applications* (1989), pp. 197–220.
- [62] Jane Wood and Denise Silver. Joint application development. John Wiley & Sons, Inc., 1995.
- [63] K. Yozgyur. "A proposal for a requirements elicitation tool to increase stakeholder involvement". In: Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on. This paper proposes a tool which encourages multiple stakeholders to participate in the requirements elicitation process and facilitates their direct and increased input, provides a common ground for RE specialists and other stakeholders. June 2014, pp. 145–148. DOI: 10.1109/ICSESS.2014.6933532.
- [64] Zheying Zhang. "Effective requirements development-A comparison of requirements elicitation techniques". In: Software Quality Management XV: Software Quality in the Knowledge Society, E. Berki, J. Nummenmaa, I. Sunley, M. Ross and G. Staples (Ed.) British Computer Society (2007), pp. 225–240.
- [65] Didar Zowghi and Chad Coulin. "Requirements elicitation: A survey of techniques, approaches, and tools". In: *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.