A Collaborative Filtering Based Persona Identification in Requirements Elicitation

He Ye

University of Tampere
School of Information Sciences
Degree Progamme in Computer Sciences
M.Sc. Thesis
Supervisor: Zheying Zhang
November, 2016

University of Tampere
School of Information Sciences
Computer Science / Software Development
He Ye: A Collaborative Filtering Based Persona Identification in Requirements Elicitation
M.Sc. thesis, 58 pages and 4 appendix pages
November 2016

Abstract

Persona is a fictional character that archetypically represents a user group. Persona identification is an important step in requirements elicitation. A review of related literature has shown that the persona is identified using qualitative approaches such as ethnographic profiling, user observations and user interviews. These approaches classify users on the basis of demographics or behavioral patterns. The drawbacks for such qualitative approaches are: they focus on detailed information gathering rather than correctly identifying representative user of persona; identified personas are too subjective as different requirements analysts may create different personas; these approaches do not scale well for a large number user involvement due to the high computational complexity of processing unstructured data. This paper proposed the collaborative filtering based persona-scenario (CFPS) approach to identify persona by calculating the similarities between the representative user to other users, combining the collaborative filtering algorithm and the persona-scenario approach. The case study shows the proposed approach improves the efficiency and accuracy in persona identification and requirements elicitation.

Keywords: Persona, User Classification, Requirements Elicitation, Collaborative Filtering

Acknowledgements

I wish to express my sincere thanks for my supervisor Dr. Zheying Zhang for her patiently guidance, invaluable assistance and earnest comments. Her professional experience in requirements engineering and zero-tolerance attitude for literature review has led to innumerable improvements of this thesis. Thanks Dr. Timo Poranen for reviewing and providing valuable comments for this thesis.

I would like to express my big thanks to Jyrki Nummenmaa, Kirsi Tuominen and all professors and staffs in University of Tampere, for the kind, support, passion and care for all the foreign students.

In addition, I would also like to thank my parents and friends for the essential support, encouragement and company. My deepest thanks to a number of my friends who participated to the case study survey.

Thank you!

He Ye

Contents

1 Introduction

Requirements engineering is one of the most critical activities in software engineering which provides guidance to the design, implementation and testing processes [Zave, 1995]. Requirements elicitation forms a critical step of requirements engineering. Requirements elicitation is the process to gather stakeholders' needs and it occurs at an early stage of requirements development [Zhang, 2007]. However, the challenge of requirements elicitation is the involvement of heterogeneous of stakeholders, such as software sponsors, developers, managers and end users [Zhang, 2007]. They are from different backgrounds and with various individual or organizational goals, social status, and behavior patterns. These stakeholders together with physical environment, form sources of requirements.

There are many approaches to eliciting software requirements from stakeholders, such as interviews, surveys, brainstormings [Nunamaker, 1991], focus groups [Kitzinger, 1994], personas [Cooper, 1999] and scenarios [Scott, 1995], etc. Software requirements are divided into three subsets: business requirements, user requirements and functional requirements [Wiegers, 2003]. User requirements are widely believed as the most critical and difficult to be elicited among these three requirements by considering two reasons: users are all different and they have different preferences, backgrounds and ways to understand the software products; it is hard for users to clarify and express their needs regarding the problem domain [Wiegers, 2003; Zhang, 2007].

Persona, a user model that represents a specific user group, is one approach to eliciting user requirements [Tu et al., 2010]. Personas are created to describe the typical user groups. They represent the core users' goals, behaviors and motivations of individual user groups [Cooper, 1999]. For the past decades, the persona approach has gained attentions in both academic and practitioner community, and is considered as a promising approach to addressing the users diversity problems [Tu et al., 2010]. Many papers and studies have shown that personas identification is conducted using qualitative techniques, such as ethnographic studies, user observations and user interviews [Aoyama, 2005; Tu et al., 2010].

Grudin and Pruitt [2002] address that identifying the representative user is the key to personas identification, but there are few studies about the way to systematically identify personas. Aoyama [2005] proposes the Persona-Scenario (PS) approach to systematically eliciting user requirements. Personas identification in the PS approach

is to classify users into different user groups on the basis of demographic variables. The user group which has the most interest in the software product is identified as the primary persona. Similarly, the user group has the secondary interest in the software product is identified as the secondary persona. In the PS approach, users are simply grouped by the demographic variables, such as the age, gender, job, education background, etc. Users' preferences for the software product are not taken into account at the user grouping stage. One significant drawback of personas identification in the PS approach is that real target users might be distributed into different user groups. Thus, the primary persona contains users who might not be very interested in the software product.

Tu et al. [2010] proposes a quantitative method to cluster users by their goals and personalities. They cluster users by asking the questions about their goals and personality characters, such as "If you are a teacher, what course do you prefer to teach?" or "Are you a rational person? ". This approach clusters users on the basis of their goals, personalities and behavior patterns. However, not every user with the similar characteristics shares the similar opinions to a software product. They should focus more on users' opinions to the product services and requirements.

Researchers have summarized the drawbacks of current persona's identification approaches. Sinha [2003] points out that the current personas identification methods emphasize detailed information gathering of the persona, and ignore the accuracy of personas identification. In other words, the identified persona might not represent the typical users. Another significant drawback is that personas identification is a subjective process and different requirements analysts may identify different personas for the same project [Sinha, 2003; Tu et al., 2010]. In addition, current personas identification approaches are highly dependent on interviews, questionnaires and observations, they are tedious and time consuming processes to identify personas. It means the current persona approaches do not scale well for a large number user involvement. Today, software systems are growing in both complexity and scale. Large and complex projects have multiple stakeholder groups from different users and organizations [Lim et al., 2012]. Therefore, it is important to support massive users to be involved in requirements elicitation. Taken these factors into account, two research questions are proposed:

- How to efficiently and objectively identify personas?
- How to improve the PS approach to support a large number user involvement?

The first research question is to address the problems of tedious and subjective processes of personas identification. The second research question is to improve the scalability of the PS approach for massive user involvement. As mentioned, a persona represents a typical user group. Thus, before analyzing the personas, we firstly discuss what a user is and the classification of users. Further more, we discuss the definition of large-scale software projects and why user involvement is important for software projects. This thesis limits the scope to elicit user requirements. The goal of this thesis is to identify personas in an efficient manner, and scale well for massive user involvement.

In this thesis, we propose an approach named collaborative filtering based persona-scenario (CFPS), on the basis of the PS approach [Aoyama, 2005]. We adapt the collaborative filtering approach, specifically the Pearson correlation coefficient algorithm [Pearson, 1985], to calculating the similarities between the representative user and other users. In this way, we form the user groups on the basis of users similar preferences for the software product. In order to evaluate the CFPS approach, we design and implement a web-based survey tool to elicit user requirements for a travel website. The case study compares the result of the CFPS approach and the PS approach.

The thesis is composed of seven chapters. Chapter 2 introduces the basic concept of requirements engineering, user requirements and the existing requirements elicitation techniques. Chapter 3 introduces the PS approach, and the advantage and limitations of the approach. Chapter 4 presents collaborative filter algorithms and explains the reason of adapting the Pearson correlation coefficient algorithm to the PS approach. Chapter 5 proposes an approach named CFPS to combining the PS approach and the collaborative filtering algorithm. Chapter 6 applies both the CFPS approach and the PS approach to a case study for eliciting user requirements for online traveling website, and evaluates the CFPS approach according to the case study result. Finally, the thesis ends up with the conclusion and future work of the thesis in Chapter 7.

2 Requirements Engineering and Elicitation Techniques

2.1 Requirements

Software requirements are a set of statements of what the software system must implement, the qualities it must achieve, and the constraints that the system must satisfy [Zave, 1995]. These requirements are defined at an early stage of system development and reflect stakeholders' needs [Sommerville and Sawyer, 1997]. From the perspective of the abstraction level of a requirement, requirements are classified into three categories: business requirements, user requirements and functional requirements. Additionally, non-functional requirements are also commonly documented for each system [Wiegers, 2003].

Business requirements describe the purposes of an organization to implement the system. These requirements usually reflect the current business practices of the company or new practices to be adopted [Courage and Baxter, 2005]. Business requirements define high level objectives from the perspective of sponsors whom concern with business goals. Sources of business requirements are typically from the funding sponsors, the acquiring customers, the marketing departments, or a product visionary [Wiegers, 2003]. An example of a business requirement is "To develop a website to sale our products via internet".

User requirements describe services of a system that users expect to have. User requirements imply the goals from user's perspective which conform to business requirements [Courage and Baxter, 2005]. User requirements are commonly represented in the form of use cases or usage scenarios [Wiegers, 2003]. An example of a user requirement is "Order a take-away food" in a restaurant recommendation website.

Functional requirements (FRs) describe services of a system that the system needs to implement. They specify the software services that the system must provide to enable users to accomplish their tasks and achieve their goals [Wiegers, 2003]. They are typically documented in a software requirements specification, which describes the expected functions of the system as fully as necessary. They are in line with the business objectives and users' goals [Sommerville and Sawyer, 1997]. FRs are the traditional "shall" statements [Wiegers, 2003]. An example of a FR is "The system shall support apple pay". Non-functional requirements (NFRs) are the constraints upon the behaviors of a system. They include usability, portability, integrity, efficiency, robustness, etc. [Wiegers, 2003]. An example of a NFR is "The system

must safely support apple pay". This example contains an additional NFR comparing with the former FR example: it emphasizes the security of the system while using apple pay. Together with FRs, NFRs are also documented in requirements specification.

In this thesis, we focus on the user requirements elicitation, as the user requirements form the basis of specification of FRs and NFRs. In general, even users possess the knowledge of a problem domain, they find it hard to clarify and express their needs regarding the problem domain. Our goal is to understand users and their needs, further identify personas from the user groups.

2.2 Requirements Engineering

The term requirements engineering (RE) is generally used 1990s with the publication of an IEEE Computer Society tutorial [Thayer and Dorfman, 1997]. Zave [1995] addresses RE as a branch of software engineering regarding the objectives, behaviors, and constraints of the software systems and the relationship of these factors to precise specifications of software behaviors. RE is the process to identify and specify the needs for software systems. The success or failure of a software project is highly dependent on the successfulness in specifying complete and correct requirements [Wiegers, 2003].

RE include activities such as requirements elicitation, requirements analysis, requirements specification, requirements validation and requirements management [Wiegers, 2003]. These activities are not in a linear and one-pass sequence. They are processed in iterative processes as illustrated in Figure 1. In the elicitation stage, requirements analysts gather initial user needs and delivery these needs to the analysis stage. In the analysis stage, requirements analysts process requirements to understand them, classify them in different categories, and relate the stakeholders' needs to possible software requirements. If the requirements are incorrect and ambiguous, analysts go back to the elicitation stage to clarify them. In the specification stage, requirements are structured and derived as written documents and diagrams. In the validation stage, requirements analysts confirm if the written sources are accurate and complete. Analysts go back to the analysis stage to re-evaluate the inaccurate requirements or go back to the specification stage to rewrite that requirement depending on the inaccuracy level [Wiegers, 2003].

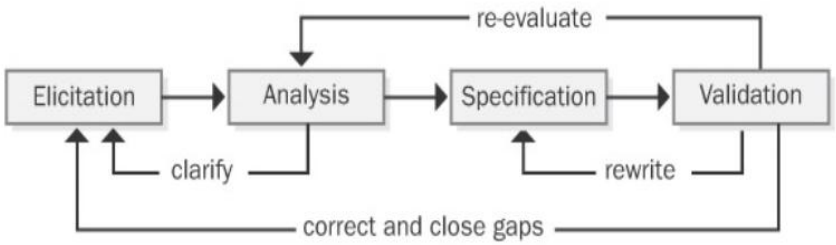

Figure 1: Requirements development is an iterative process [Wiegers, 2003].

Requirements elicitation

Requirements elicitation is a process of gathering the needs and constraints from various stakeholders and it occurs at an early stage of requirements development [Wiegers, 2003]. Zhang [2007] states requirements elicitation is a critical but error-prone stage in requirements development. Users' needs are initially summarized in the elicitation stage and these needs are the basis of follow up requirements analysis, specification and validation activities. Thus, without analysis and verification, the requirements elicited at this stage are error-prone. In addition, as requirements are elicited through a variety of various stakeholders (users, customers, developers, organizations or environments, etc.), it is hard to use a single technique to elicit all requirements in one single elicitation session. The best practice is to take a set of sessions in parallel or in sequence [Zhang, 2007]. Thus, selecting the appropriate elicitation techniques is critical in the elicitation process.

There are many requirements elicitation techniques available to elicit the requirements, such as interviews, surveys and questionnaires, brainstormings [Nunamaker, 1991], prototypings [Davis, 1992], focus groups [Kitzinger, 1994], personas [Cooper, 1999], use cases [Jacobson, 1991] and scenarios [Scott, 1995], etc. Details of these elicitation techniques are given in Section 2.5.

Requirements analysis

Requirements analysis is a process of refining the users' needs and constraints [Nuseibeh and Esterbrook, 2000]. This process deals with a large number of unstructured data about users' needs. Requirements analysts detect and resolve conflicts, identify missing requirements, and remove unnecessary ones. The goal of requirements analysis is to refine requirements with sufficient quality and details, discover problem as well as solve conflicts of elicited requirements. The requirements analysis includes decomposing high-level requirements into concretes, analyzing feasibility, building software prototypes, and setting priorities [Wiegers, 2003].

Requirements specification

Requirements specification constraints the output of requirements analysis clearly and precisely. Requirements specification ensures the requirements, both FRs and NFRs, are well understood and documented as detailed as possible in a consistent, accessible, and reviewable way [Wiegers, 2003]. Structured natural language, is a common way to document requirements specification. Besides, graphical notations and formal specifications are also used to provide complementing information on requirements written in a natural language.

Characteristics of good software requirements specification have been summarized as correct, unambiguous, complete, consistent, ranked for importance stability, verifiable, modifiable and traceable [IEEE Standard 29148, 2011]. Basically, there are five activities during requirements specification process: adopting a software requirements specification template, identifying sources of requirements, marking uniquely label each requirement, recording business rules and specifying quality attributes [Wiegers, 2003].

Requirements analysts generally define a standard template for documenting software requirements or adopt an existing specification template to fit the project. IEEE Standard 29148 [2011] provides one of the most popular template to be referenced. Identifying sources of requirements is to ensure each requirement is able to be traced back to its origin in case of requirement's change. The purpose of a uniquely label is to trace requirements and record their changes. In terms of business rules, they include company policies, government regulations, and computational algorithms, etc. Business rules may lead to or enforce requirements. For example, a military information website does not allow the requirement of "online order a gun" in guns forbidden countries. Documenting quality requirements helps analysts and developers to make appropriate decisions in the future. Requirements quality attributes include the information regarding performance, efficiency, reliability, usability, etc. [Wiegers, 2003].

Requirements validation

Requirements validation is a process of reviewing software requirements to detect and fix errors. This process is to ensure the software requirements specification correctly describes the functions and constrains that will satisfy the various stakeholders' needs. Besides, requirements validation activities ensure requirements are complete with good quality, and the requirements are consistent with each other [Moore et al., 2001; Wiegers, 2003].

Requirements validation is not a single discrete process that is performed after the requirements specification. It is throughout the iterative elicitation, analysis, and specification activities, such as incremental reviews of the growing software requirements specification [Wiegers, 2003].

Requirements management

Requirements management is essential to achieve expected objectives of productivity and quality during software development. Changes arise naturally during the life cycle of project, such as new business priorities or existing bugs fix priorities. Therefore, requirements management keeps track of changes and ensure requirements are modified, updated and maintained under a control [Santillan and Käkölä, 2016].

Requirements management activities include defining the requirements baseline, reviewing requirements changes, evaluating the impact of the changes, negotiating new commitments of the software changes, tracing requirements to their corresponding sources, tracking requirements status and so on [Wiegers, 2003].

2.3 Stakeholders and Users

Stakeholders are individuals, groups or physical environments that can affect the realization of an organization's goals [Freeman, 1984]. Carroll [1991] summaries five major stakeholder groups that are recognized by most companies: owners, employees, customers, local communities and society. Wiegers [2003] defines stakeholders representing individuals or groups who are interested in a specific project, such as analysts who analyze the project requirements, users who will use the system and provide analysts with their needs, sponsors who support the project by financially, developers who build and maintain the project, etc. In addition, stakeholders also refer to the physical, organizational, or legislation environment where the desired system is used [Zhang, 2007]. From these descriptions, two common agreements of stakeholders are: stakeholders are key factors to affect the success or failure of a system; stakeholders are composed of complex components, and customers form a subset of stakeholders.

Users can be generalized to stakeholders, however, not all stakeholders are users. Users, often refer to end users, are individuals who use the product to accomplish their tasks directly or indirectly. Users form a subset of stakeholders. Specifically, users are a subset of customers [Wiegers, 2003]. They are from different backgrounds with various goals, behavior patterns, and personalities [Zhang, 2007]. Users have different ways to understand, communicate and express their opinions to software

products. They often have different opinions to the most important features for a system. A specific user, for example, might be more interested in discount information provided by the system while another user might be more interested in fashion and entertainment information. Basically, differences between users could be: the frequency they interact with the product, their application domain experiences, the features they use, the tasks they accomplish, the qualities they expect, and their access privilege or security levels [Wiegers, 2003].

Wiegers [2003] groups users into some distinct user classes: favored user classes, disfavored user classes, ignored user classes and other user classes. Each user class is a subset of users, which is a subset of the customers and a subset of the stakeholders. Figure 2 illustrates the hierarchy relationships of the stakeholders, customers and users. Before understanding user classes, it is necessary to understand stakeholders and their components.

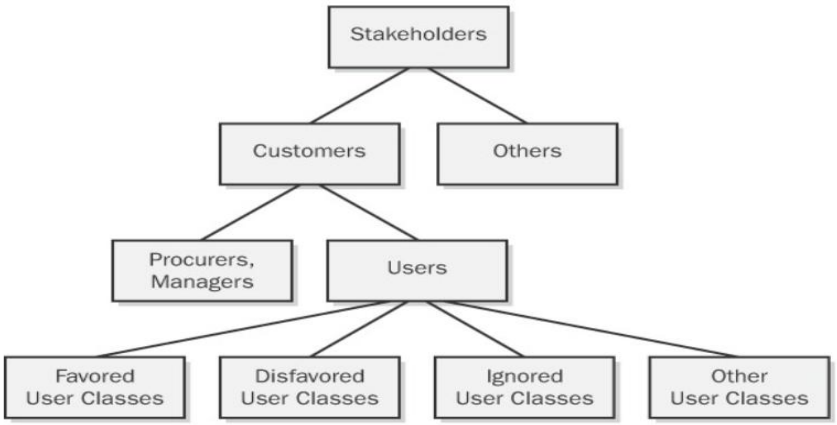

Figure 2: A hierarchy of stakeholders, customers and users [Wiegers, 2003].

Customers

A customer is an individual or a group who derives either direct or indirect benefit from a software product. Software customers include stakeholders who raise the ideas of the software product, who support the product in finance, who use the product or who receive the output generated by the software product [Wiegers, 2003]. In general, customers refer to producers, sponsors and users [Sommerville and Sawyer, 1997]. Users form a subclass of customers who use the software product directly or indirectly.

Users and users classes

Users, despite of the direct end users, other individuals are considered as users as well, such as the managers of the direct users, the system administrators of a system, people who receive information from the system or people who are considering whether they will use the system in the future [Courage and Baxter, 2005]. There are no standards to separate users, and users are grouped based on their differences [Wiegers, 2003]. The user groups form the basis of persona and it is important to understand these users categories. Courage and Baxter [2005] suggest to create at least one persona per user group. We introduce some user classification methods and discuss how they affect personas identification.

According to users usage patterns, direct or indirect, users are classified into three types: primary, secondary and tertiary. The primary users refer to the individuals who interact regularly or directly with the product. For example, users browse the traveling information more than once a week are considered as the primary users of a travel website. The secondary users refer to individuals who use the product infrequently or through an intermediary. For example, users browse the traveling information in the website less than once a month are likely to be considered as the secondary users. The tertiary users refer to individuals who are affected by the system or the purchasing decision makers of the system [Courage and Baxter, 2005]. Based on this classification method, analysts are able to easily identify the primary persona, the secondary persona and the tertiary persona. However, personas identified by this method still contain a variety of unknown users, and it is hard to conclude a user model according to their backgrounds, personalities and goals. We consider this user classification method is too coarse-grained. For example, the primary users of a travel website might contains users from different ages, jobs, hobbies and goals.

One widely used method is to classify users into different groups based on users' demographic variables, such as the kind of company, the kind of major, the kind of age or gender [Aoyama, 2005; Tu et al., 2010]. For example, the case study presented by the PS approach classifies students according to the genders and their study majors. There are four groups: female student in engineering, male student in engineering, female student in policy and male student in policy [Aoyama, 2005]. This kind of method is needed of gathering massive user information through surveys, interviews or observations before classifying users. This method is commonly adopted in personas identification as it is well understood. However, one significant drawback of this method is that users who are interested in the product might be distributed into

different groups. It groups users by focusing on users natural characters, rather than their interests and goals for the system.

As shown in Figure 2, based on users' preferences for the product, Wiegers [2003] classifies users into favored user classes, disfavored user classes, ignored user classes and other user classes. Favored user classes are those individuals or groups who accept and use the system. Favored user classes receive high priority treatment when analysts make decisions or resolve conflicts comparing with other user classes. Disfavored user classes refer to those individuals or groups who hardly use the product for personal, security, legal or safety reasons [Wiegers, 2003; Gause and Lawrence, 1999]. However, identifying personas according to their preferences is still in the theory discussion stage, and there are few approaches to calculating similarity between users' preferences [Tu et al., 2010].

Tu et al. [2010] use cluster analysis (CA) approach to classifying users. The CA approach classifies users on the basis of users' goals, behavior patterns and their personalities. Users are asked to fill out the online questionnaires regarding users' goals, behaviors and feelings. The CA approach uses the Euclidean Distance algorithm to cluster users based on their similar attitudes towards the questions. However, there is no definitive conclusion regarding how many user clusters should be identified and it is dependent to requirements analysts to determine the number of user clusters.

In this section, considering user groups form the basis of the personas, we discuss users, user classification methods and how these classification methods affect the further personas identification. Detailed personas identification approaches are given in the following section.

2.4 Personas Identification

Users often have different opinions on the software products. It is hard for requirements analysts to analyze which user's preference is the most important and which user's preference is less important for the product. In the user centered design approach, persona is first proposed by Cooper [1999]. Persona is a fictional character created to describe the typical users. The purpose of the persona is to understand the target users from various unknown users, and keep all analysts focused on the same target [Cooper, 1999]. In other words, a persona as a model of a group of typical users, is a promising solution to address user diversity problem.

A persona is often specified in these six components: identity, status, goals and tasks, skill set, requirements and relationships [Courage and Baxter, 2005]. Identity gives the persona a name, gender, age and other demographic information as well as a photo of the user. Status identifies the persona as a primary, secondary or tertiary one. Goals describe the motivations and expectations that the persona wants to achieve through the software product. Skill set means the background and expertise of a user, including education, training and specific skills. Requirements describe the needs from the persona to the product. These components are an idealized information to create a persona, however, we may not have all the information in reality. Creating a persona is also an iterative process, which means more and more users information become clarified and concrete with conducting user requirements activities. Below Figure 3 is an example of a persona profile for a travel agent.

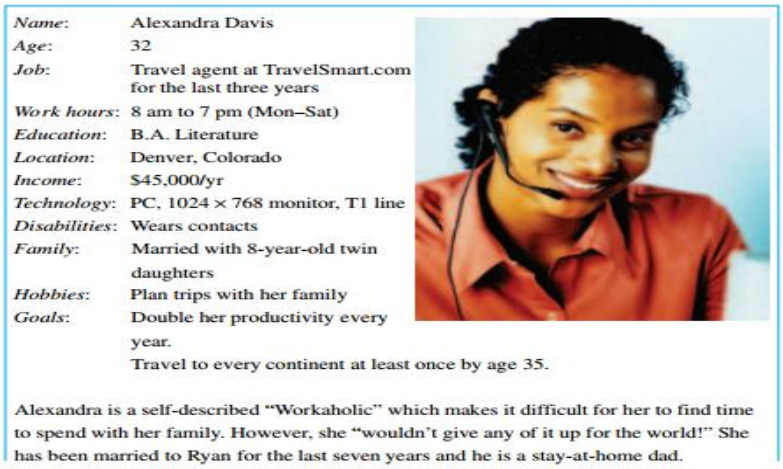

Figure 3: An example of a persona profile for a travel agent [Courage and Baxter, 2005].

There are many benefits to use personas. Since we are not able to speak for each end user, we use persona as a model to represent the end users. Personas help to limit the open space of possible personalities and attitudes among various users. They help all analysts focus on the same target user group, instead of his or her own vision of users [Courage and Baxter, 2005; Shahri et al., 2016]. Besides, personas represent the specific target user group from unknown and various users, they provide accuracy and valuable opinions on specific part of the product. For example, if a persona represents users from engineering backgrounds, and their opinions on software safety and security might be more valuable than other personas. In addition, personas as the user models, can be used in cognitive discussions and other requirements activities, such as interviews, surveys, scenarios, brainstormings [Courage and Baxter, 2005].

A persona is identified on the basis of different user groups. In the previous section, we have discussed some user classification methods and how these methods affect personas identification. In this section, we introduce the cluster analysis approach proposed by Tu et al. [2010] and the PS approach. The cluster analysis approach is a quantitative approach in personas identification, while the PS is a typical qualitative approach to identifying personas.

Cluster analysis approach

Cluster analysis (CA) approach, proposed by Tu et al. [2010], is the first quantitative approach to identifying the cluster of users. Before Tu's approach, qualitative approaches usually adapt requirements elicitation techniques such as interviews, observations and surveys to identify personas. However, these qualitative methods have met some criticisms. For example, the created personas can not accurately represent the target users. The personas identification processes are too subjective and different creators may identify different personas. To address the above problems, Tu et al. propose the CA approach. It uses the Euclidean Distance (ED) algorithm to calculate the linkage distance between two users.

In the CA approach, users are asked to provide ratings to the questions on the scale of 1 to 7, with 1 being the most dissimilar and 7 being the most similar. The questions are very abstract, which are about users' goals, personalities, behavior patterns. For example, one of the question regarding the participants habit is "Are you emotional?". In the CA approach, the personal demographic information such as age, gender, job, etc. Is not used. Once the users finish answering these questions, the ED algorithm is used in the following steps where t represents the index of the iterative process and its initial value is 1:

Step 0: Each participant is first treated as a separate cluster.
Step 1: Calculate the smallest distance between any two clusters. Mark these closest clusters as C_i and C_j.
Step 2: Merge clusters C_i and C_j to form a new cluster C_n+t (C_n represents the new cluster consists of C_i and C_j).
Step 3: Calculate the distance between the new cluster C_n+t and all remaining clusters C_k as follows: $dC\_nC\_k = \min\{dC\_iC\_k, dC\_jC\_k\}$.
Step 4: Store the cluster C_n+t as a new cluster and remove clusters C_i and C_j. Let $t = t+1$.
Step 5: Return and continue to Step 1 until the analysts consider they have the best clusters.

According to the analysis, the CA approach obtains 2 clusters with 24 volunteer users participated in the case study. These 2 user clusters form the basis of two typical personas, and analysts create two persona profiles for them. Once the personas are identified, the volunteer users are requested to participate in further elicitation activities such as interviews, focus groups, scenarios, etc.

The CA approach, a quantitative approach for personas identification, is one of the earliest approach that analyses users' goals, behaviors and personalities, instead of demographic information. To some extent, it addresses the subjective problem in the traditional personas identification approaches. Even different analysts with the same questions will have the same personas. However, there are still some drawbacks in the CA approach. Firstly, the survey about users' information is only regarding users' goals, personalities and behaviors. These questions are too abstract to group the users. Users with the similar goals might not have similar preference for the software product. One possible solution is to combine users' goals and personalities information with users' preferences for the software product. Secondly, the ED algorithm is applied to find the smallest distance between any two clusters, which means we need to calculate C_n^2 times where the users size is n even in the first round. The calculation continues until analysts obtain their ideal personas number. The time complexity problem is significant in this approach.

The PS approach

As mentioned, the PS approach is a systematic approach proposed by Aoyama [2005] to identifying personas and eliciting user requirements. The user grouping method in the PS approach is on the basis of the traditional demographic differences. The detailed discussion of the PS approach is given in Chapter 3, and we explain the reasons of choosing the PS approach to be improved in this thesis instead of the CA approach in that chapter.

2.5 Requirements Elicitation Techniques

Based on the user differences we discussed above, one single elicitation technique is unlikely enough to elicit user requirements from various stakeholders because the situational context changes during the elicitation process and the requirements analysts' experience varies [Zhang, 2007]. Eliciting requirements from various sources is the one of most challenging activities, however, persona is widely believed as a promising approach that addresses diversity problem of users [Courage and Baxter, 2005]. The goal of persona is to identify a group of primary target users from diversity users [Aoyama, 2005]. In the personas identification process, some

elicitation techniques are widely used together, such as interviews, surveys, brainstormings [Nunamaker, 1991], use cases [Jacobson, 1991] and scenarios [Scott, 1995]. Users are commonly invited to participated in these eliciting activities during requirements elicitation processes, and these techniques commonly used together with personas identification. In this section, we introduce these elicitation techniques.

Interviews

Interviews are traditional sources of requirements input. An interview is a guided conversation that one person seeks information from other individuals or groups [Courage and Baxter, 2005]. They are one of the most frequently used techniques for requirements gathering, which include structured interviews, non-structured interviews, oral interviews, written interviews, one-to-one interviews and group interviews [Katrina et al., 2004].

Interviews are good for collecting rich and detailed information. They provide good opportunities for requirements analysts to understand and explore a domain and users' usage in depth. However, interviews are not good for rapidly collecting information, and not appropriate for gathering information from a large number of users because they take significant time to conduct the requirements activities [Lauesen, 2002; Courage and Baxter, 2005]. In personas identification process, interviews are widely used to gather users information, such as users' education background, goals and motivations. However, interviews are only useful when users number is small and the purpose is to gather detailed information in depth.

Surveys

Surveys are written lists of questions given to stakeholders to obtain information about system requirements, which are collected and compiled. In contrast to interviews, surveys are useful to collect information from massive users in a short period. They allow requirements analysts to ask each user same questions in a structured manner.

Surveys are good for identifying potential user population, finding out users' preferences for the current product and collecting information from massive users. In contrast, the detailed information is not possible to be captured in a survey [Zowghi and Coulin, 2005; Courage and Baxter, 2005]. One significant problem is that a valid and reliable survey is not easy to design. A poorly designed survey provides meaningless and inaccurate information. A well designed survey can provide analysts with great data, but analysts must pay attention to survey creation, collection, and

analysis. It it good to start with brainstorming sessions to gather and discuss all the potential questions [Courage and Baxter, 2005].

Brainstorming

Brainstorming is to gather stakeholders together and explore as many solutions or ideas to a given problem or question as possible [Kunifuji et al., 2007]. The purpose of brainstorming is to generate a number of new ideas, and to derive from them for further analysis. In the brainstorming session, all ideas are need to be well recorded so that they are not lost. Each brainstorming session needs three types of people to get involved: participants, moderator and the scribe. It is recommended to recruit 8-12 participants per session to participate the brainstorming activities. A moderator is needed per session to organize the brainstorming session and a scribe is needed to write down what the moderator paraphrases and the numerous new ideas [Courage and Baxter, 2005].

Brainstorming is ideal when analysts are trying to scope the features or information of a software product [Herrmann and Nolte, 2010]. Similarly with interviews, a brainstorming session is not appropriate for collecting data from a large number of users but it is helpful to collect information in detailed and in depth.

Use case and scenario

A use case describes a sequence of interactions between an external actor and a system where an actor refers to a person, a software or a hardware device that interacts with the system to achieve a specific usage objective [Wiegers, 2003; Cockburn, 2001]. The use case approach describes all tasks that users are need to interact with the system [Wiegers, 2003]. Use case diagrams present a high-level visual of the user requirements. An example of a user case is “A customer orders coffee” with a cafeteria ordering system, as shown in Figure 4. A customer is an external actor of the coffee ordering system, and the “order coffee” action is the interaction with the system. There is no visibility into the system internal.

Wiegers [2003] states a scenario is a specific instance of a use case which often presented as a story. By that means, general statements of user goals that users need to perform are use cases, while a specific description of a use case is a usage scenario. Scenario describes the ideal way that the persona accomplish a given task or problems the persona might encounter during the process [Alexander and Maiden, 2004; Courage and Baxter, 2005].

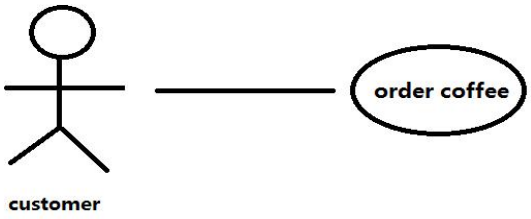

Figure 4: An example use case of a customer orders coffee through cafeteria ordering system.

An example scenario is that "A customer orders coffee with a cafeteria ordering system by clicking the start order button, then the system asks the customer to choose the preferred coffee type (latte or espresso). After the customer chose coffee and insert coins, the system prints the receipt of order.". We can see this scenario is one specific situation of "order coffee" use case. Another possibility of this scenario might be that no coffee is available and the customer is suggested to exit the system. Scenarios are widely used to analyze, filter or prioritize user requirements, and making the system more robust and realistic. Scenarios are commonly used with personas approach together, such as the PS approach. Once the personas are identified, the personas are asked to conduct the scenarios activities for further requirements analysis.

2.6 User Involvement in Large-Scale Project

Research has shown the importance of user involvement. User involvement is a major success factor for requirements elicitation. End user involvement is an effective way to avoid an expectation gap between the customers' expectation and the real system. Courage and Baxter [2005] states the importance of knowing users as "the single most critical activity to developing a quality product is understanding who your users are and what they need, and documenting what you have learned". Pagano and Bruegge [2013] have found that user involvement provides valuable information to improve software quality and to discover missing requirements. They consider a large number user involvement in requirements eliciting process helps to improve the accuracy of requirements [Pagano and Bruegge, 2013]. Bano and Zowghi [2015] have identified about 290 publications and agree with the positive impact of user involvement. They describe software systems with more end users involved in requirements elicitation processes are more satisfied and accepted by the market.

Today, software systems are growing in both complexity and scale. Large and complex projects basically have multiple stakeholder groups from different

individuals, companies, and organizations. For example, the requirement elicitation in FBI Virtual Case File project include 12,400 users and more than 50 stakeholder groups [Lim et al., 2012]. There are mainly two ways to measure the size of projects: from the perspective of development effort and from the perspective of requirements elicitation effort [Burstin and Ben-Bassat, 1984].

From the perspective of development effort, many measures to size a project and define the scale of the projects. Some widely used measures of project size include lines of code (LOC) [Albrecht, 1979], function points (FPs) [Low and Jeffery, 1990], number of developers and man-hours [Brooks, 1995]. These measurement methods have been used to indicate the relative size of projects, as shown Table 1. Software projects with more than 500,000 LOC, or 5,000 FPs or with over 50 developers are considered as large scale systems [McConnell, 2004].

Project Size	Measure		
	Lines of Code	Function Points	Number of Developers
Small	< 2,000	< 100	< 5
Large	> 500,000	> 5,000	> 50
Ultra-large	1,000,000,000	> 100,000	> 1,000

Table 1: Measures of the size of projects [Lim et al., 2012].

From the perspective of requirements elicitation effort, the number of users and stakeholders of the project is the key factors to measure the project size. Burstin and Ben-Bassat [1984] define a large-scale software project has a large and diversity users involved, and it entails a variety of human, organizational, environment and automated activities. Cleland-Huang and Mobasher [2008] describe a large-scale project to have more than thousands or hundreds of thousands of users. In this measure method, software products with a large number of users are considered as large-scale projects.

With the development of internet, massive social websites, applications, games or online services are provided and these software have gained a huge number of users, such as Facebook, Amazon.com, Clash of Clan, etc. As the users volume and software scale are growing, it is important for requirements elicitation approaches to support massive user involvement in requirements eliciting processes. In this thesis, we focus on software projects which provide the services to the public and gaining a huge

number of active users. In Chapter 6, we will choose a typical large scale project as a case study to evaluate our proposed approach.

3 The Persona-Scenario Approach

3.1 Introduction of the PS Approach

The Persona-Scenario (PS) approach is proposed by Aoyama [2005] to eliciting user requirements for software embedded in digital products. In the PS approach, the persona that represents the target users or who use core features of the product is called the primary persona. The secondary persona refers to those use the product infrequently or less interested in the product comparing with the primary persona. For example, students and teachers are primary persona for an online examination system. School administrators might be the secondary persona. They might not directly use the system, instead accessing its data through other application or study reports.

The PS approach identifies personas based on the conjoint analysis. The conjoint analysis is widely used in marketing, retailing and behavioral sciences [Hauser and Rao, 2005]. It is to measure the value of each feature for a product and predicts the value of any combination of features [Halme and Kallio, 2014]. In other words, the conjoint analysis is a statistical technique to determine how people value different attributes of a product (e.g. feature, quality, function) [Luce and Tukey, 1964]. An example of conjoint analysis is that male teachers, female teachers, male students and female students provide ratings to different services of the online learning system according to their preferences. In this way, the useful and popular services (the service with higher rating value) could be identified according to the user groups' total rating value.

In the PS approach, users are classified into different user groups according to their demographic variables, such as their age, gender, education background and major, etc. These user groups are asked to provide ratings to each requirements of software which is prepared by stakeholders. The ratings scale is from 1 to 5 according to each user's usage frequency and user's preference. The user group with the highest total rating value in total is identified as the primary persona. Similarly, the user group with second highest total rating value is identified as the the secondary persona. Once the personas are identified, the PS approach identifies the hot spots of requirements by evaluating the interactions between personas and requirements through the usage of scenarios. Requirements hot spots refer to those requirements received varying ratings across different user groups, which are characterized by high standard deviation value.

The PS approach classifies the requirements into three categories: variant requirements, common requirements and marginal requirements. Variant requirements represent the requirements varying the ratings across different user groups, which means users have different opinions on these requirements. Basically, variant requirements are characterized by high standard deviation value. Common requirements refer to the requirements widely used by most user groups and most of users give high rating scores to these requirements. They are characterized by high average value and low standard deviation value. Marginal Requirements refer to those requirements which are significantly less frequently used by most user groups and they are characterized by both low average value and low standard deviation value.

3.2 Steps in the PS Approach

The process of PS approach is illustrated in Figure 5. There are mainly three steps: simultaneous conjoint analysis between users and requirements; identifying primary persona and requirements hot spots as well as analyzing requirements with personas and scenarios.

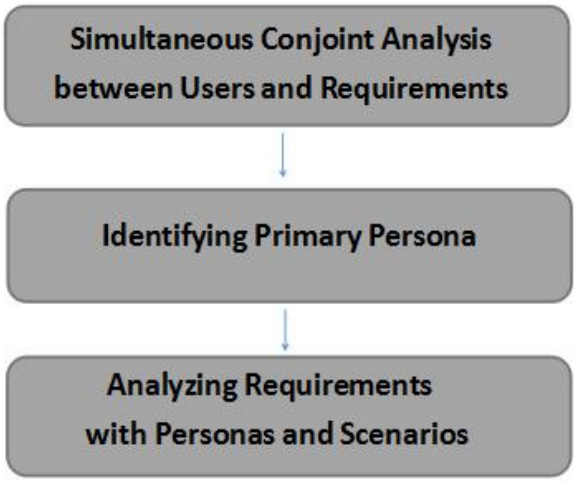

Figure 5: Process of the Persona-Scenario approach [Aoyama, 2005].

Step 1: Simultaneous Conjoint Analysis between Users and Requirements

This step is intended to classify users and requirements with conjoint analysis. Conjoint analysis starts with simultaneously decomposing the users (U) and requirements (R) into some disjoint groups respectively. As discussed in the previous chapter, there are many methods to classify users according to user differences. In the PS approach, users are decomposed into different groups according to demographic variables, such as age, major, gender, etc. The PS approach decomposes user space

U into a list of user groups where the number of user groups is n on the basis of the demographic variables:

$$U = \sum u_i \quad \text{for i = i..n} \qquad (1)$$

Similarly, the requirements space R is decomposed into a list of m requirements. Each requirement is an independent unit:

$$R = \sum r_j \quad \text{for j = j..m} \qquad (2)$$

The PS approach defines C_i to represent user's preference value for each requirement r_j provided by user u_i, where V represents the variant and common requirements:

$$C_i = \sum V_{ij}$$

$$\text{For} \quad j = \{j : r_j \cap \{Variant \,/\, Common\text{Requirements}\} \neq \Phi\} \qquad (3)$$

Once users and requirements are decomposed, the PS approach starts conjoint analysis by inviting users to rate these requirements according to their preferences and usage frequency. Users are asked to provide rating from 1 to 5 for each requirement. In this way, a preference matrix of user groups and requirements is generated. Figure 6 is an example presented in the PS approach of illustrating a preference matrix of the user groups and requirements. It is the field study figure of mobile phone products conducted in Aoyama's paper.

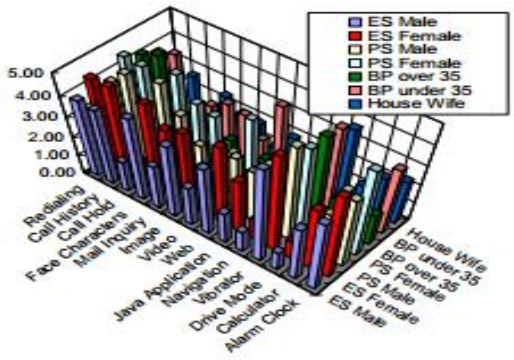

Figure 6: Example preference matrix of user groups and requirements [Aoyama, 2005].

Step 2: Identifying the primary persona and requirements hot spots

As the matrix of use groups and requirements is generated in Step 1, the average value of frequency usage (Ave.) of requirements and the standard deviation (SD) value can be calculated from the usage statistics. The PS approach identifies the user group with the highest total rating value of requirements usage as the primary persona. For example, Table 2 is a usage analysis for a mobile phone in PS approach. Users are classified into four groups: ES F (female students in engineering), ES M (male students in engineering), PS F (female students in policy) and PS M (male students in policy). As illustrated in Table 2, the total usage value of ES F, 64.0, significantly outperforms than other user groups. Thus the female students in engineering are identified as the primary persona.

Services	ES F	ES M	PS F	PS M	Ave.	SD
Redial	4.41	3.57	3.53	3.93	3.86	0.41
Calculator	3.35	2.43	3.33	2.71	2.96	0.46
Vibrator	4.94	4.93	4.87	4.79	4.88	0.07
Personal Calling Tone	4.06	2.14	1.53	1.29	2.26	1.26
User Directory	3.59	2.71	2.60	2.71	2.90	0.46
Directory Group	4.53	4.14	3.67	3.86	4.05	0.37
Alarm Clock	4.82	3.93	3.87	3.93	4.14	0.46
Scheduler	1.94	3.36	1.53	2.93	2.44	0.85
Secret Memory	3.24	1.29	1.27	1.00	1.70	1.04
E-mail Attachment	4.53	2.21	2.67	2.50	2.98	1.05
E-mail Editor	4.82	2.50	3.20	3.14	3.42	0.99
User Symbols	3.82	3.14	4.93	3.71	3.90	0.75
Face Character	3.59	4.07	4.87	3.71	4.06	0.58
Receive Folder	1.47	4.71	3.40	4.00	3.40	1.39
Image	3.18	2.50	2.60	2.43	2.68	0.34
Video	1.88	1.29	1.47	1.00	1.41	0.37
Web	1.24	2.93	3.53	3.07	2.69	1.00
Java App	3.24	2.36	1.67	1.64	2.23	0.75
Navigation	4.53	1.29	1.27	1.00	2.02	1.68
Total=Coverage	**64.0**	55.5	55.8	53.4	58.0	-

Table 2: Usage analysis of mobile phone services in PS approach [Aoyama, 2005].

Since the primary persona is identified, the next step is to identify hot spots of requirements. As mentioned, the PS approach defines three types of requirements: common requirements, variant requirements and marginal requirements. Both common requirements and variant requirements are our focuses, as they form the basis of the final user requirements. Common requirements are the most important component as they represent most users' preferences. Besides, the PS approach analyzes the variant requirements as hot spots in requirements which are attractive to

specific user groups. In addition, scenarios are used as well to analyze the requirements hot spots through the interactions with the primary persona.

Step 3: Analyzing requirements with personas and scenarios

Once the primary persona and requirements hot spots are identified, the PS approach continues to analyze the interactions between the primary persona and requirements hot spots regarding persona's usage of scenarios. As mentioned in the last chapter, scenarios analysis is a process of providing detailed descriptions of a set of use cases to individual users. Users are asked to participate in the usage scenarios activities and provide feedback from their point of view. Scenarios help to analyze users' preference patterns of the system, and to refine the initial elicited requirements.

Service transition diagram (STD), a simplified state transition diagram, is used in the PS approach to record and analyze the scenarios' results. With STD, the PS approach traces the interactions of the primary persona with scenarios by interviewing the users from the primary persona. Figure 7 is an example of STD for a coffee ordering system. Figure 7 presents the usage ratio elicited from the primary persona. When a customer orders coffee, he/she is allowed to select different sizes of the coffee mug (e.g. tall, grande or venti), different types of coffee (e.g. espresso, latte or cappuccino) and with sugar or not. We find out most of the users prefer one pattern to order coffee which is indicated by bold line in Figure 7. This pattern suggests that most users prefer to select the coffee type first, then to select size and with sugar or not.

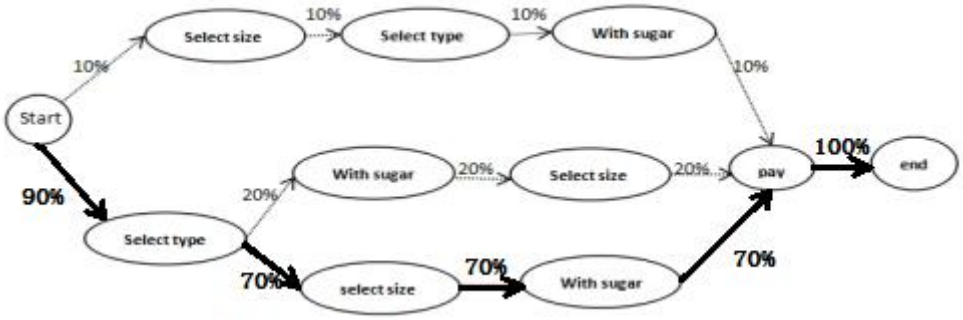

Figure 7: An example of service transition diagram of coffee ordering system.

3.3 Discussion of the PS Approach

Persona is a requirements elicitation technique that first proposed by Cooper [1999], however, there is no concrete approaches for personas identification before the PS approach [Aoyama, 2005]. The PS approach provides a systematic guidance to identify persona, analyze scenarios and elicit user requirements. The personas

identification in the PS approach is on the basis of user-requirement rating matrix and it is easily to be understood and adopted in the practical projects.

However, there are some limitations of the PS approach. The PS approach attempts to classify users into different groups based on demographic factors, such as the information of gender and major, without considering the interactions between the users and the system. One drawback of this classification method is that users with most interest to the software product are distributed into different groups at this stage. The primary persona identified on the basis of this user classification method might not represent the real target users. For example, a female politic student has a passion to a software system, but she is not identified as the primary persona as the male students in engineering represent the primary persona. In addition, personas identification in the PS approach is too subjective. The personas identification process is highly dependent on analysts' experiences and preferences. Different analysts might use different demographic variables to classify users and they might identify different personas. In addition, a persona represents a group of users' goals, behaviors and motivations [Tu et al., 2010], but the PS approach only focuses on users' preferences for the software requirements and there is no discussion about users' goals and their motivations. However, the CA approach only contains these abstract information. Therefore, it would be a good way to combine this information in the survey. With the information of users' goals and motivations, we will have a better understanding about the target users.

Comparing with the CA approach, the PS approach is a more systematic approach for personas identification and it provides guidance on how to further elicit user requirements with the identified primary persona. The PS approach has more discussions about how to make use of the personas in requirements elicitation process. Considering we will elicit user requirements with identified personas in our case study, we believe the PS approach is more appropriate to be adapted.

In the CA approach, there is no definitive conclusion regarding how many user clusters should be chosen and it is totally up to the analysts to determine the number of user clusters. It is more difficult to be adapted due to its complexity problem in calculation which we have mentioned in the previous chapter. Besides, our goal is to improve personas identification and the scalability to support a large number user involvement. Due to the complexity problem in calculation, it is hard to adapt the CA approach to support massive user involvement. Therefore, taken into account of above reasons, we decide to improve the PS approach in this thesis.

In the following chapters, we propose an approach to addressing the inefficiency and subjective problems in the PS approach. Considering the collaborative filtering algorithm is on the basis of the user-item matrix which is similar with the user-requirement matrix used in the PS approach, thus we propose to adapt collaborative filtering algorithm to the PS approach aiming to classify user groups automatically according to similarity coefficient value between users. Collaborative filtering algorithms are widely used in the recommender systems. Basically, they classify users by analyzing users' historical interests and calculating the similarity coefficient value between users, then recommending items for the specific user. We try to adapt collaborative filtering algorithms to the PS approach to identify the primary persona on the basis of users' similarities, rather than the demographic factors.

4 Collaborative Filtering Approach

4.1 Recommender Systems

The recommender system is a subset of information filtering systems which aim to present items that are likely to attract users. The objective of the recommender system is to address the overloaded information to filter the useful information for users when they have no idea about what they are looking for [Ricci et al., 2011]. The recommender systems have become extremely popular in decades, and they have been applied in many software products. For example, Amazon.com uses user's purchase records to recommend items; Youtube recommends videos on the basis of a user's historical viewing patterns and ratings, and Facebook recommends friends according to a user's personal information and friend's groups.

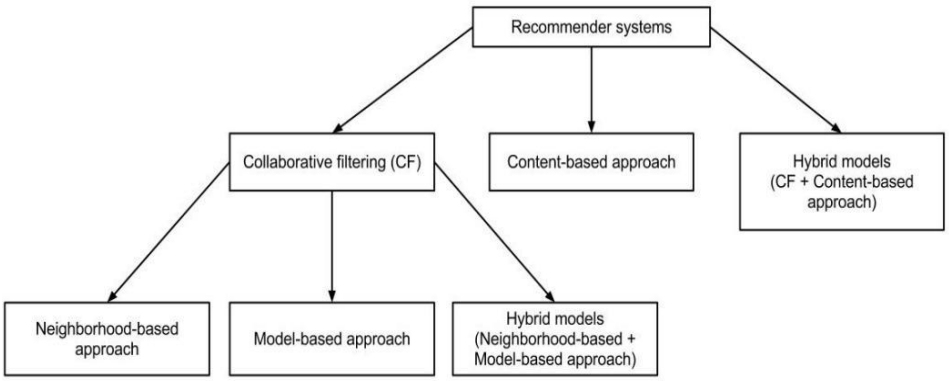

Figure 8: Filtering approaches in recommender systems [Ricci et al., 2011].

As shown in Figure 8, the recommender systems are typically implemented in three approaches: the collaborative filtering (CF) approaches, the content-based approaches and the hybrid models approaches. Further, the CF approaches are classified into three main subsets: the neighborhood-based CF approaches, the model-based CF approaches and the hybrid models approaches. The neighborhood-based CF approaches are also named memory-based CF approaches. The neighborhood-based CF approaches are to identify a group of similar-minded users called neighbors and they are further divided into the user-based collaborative filtering (UCF) approaches and the item-based collaborative filtering (ICF) approaches [Sarwar et al., 2001]. The UCF approaches make predictions by using users' historical ratings to calculate similarity coefficient value between users while the ICF approaches make predictions by calculating similarity coefficient value between items. The model-based CF approaches, instead of manipulating the ratings directly, train a predefined compact model based on observed user-item rating matrix [Yang et al., 2015]. The hybrid CF

models combine the neighborhood-based CF approaches and the model-based CF approaches, and they are only used in the complex recommender systems. Among them, the neighborhood-based CF approaches are the most popular algorithms as they are well understood and easily applied in the practical projects [Goldberg et al., 1992]. The model-based CF approaches are able to accurately predict the items, however, the prerequisite of the model-based CF approaches is to train the model which much effort should be paid in advanced.

The content-based approaches are mainly dependent on items' descriptions to make personalized recommendations. They produce recommendations on the basis of keywords matches between the descriptions of an item and the user profile [Yuan et al., 2014]. In other words, they recommend items similar to the ones that the user preferred in the past. For instance, the content-based approach will likely recommend a user some basketball shoes if a user have bought some basketball clothes before. Since the similarity of items is based on the content, the content-based approaches might make less accurate recommendations if the content features are not sufficient [Yuan et al., 2014].

The hybrid models approaches combine the CF approaches and the content-based approaches to alleviate their drawbacks and to enhance the accuracy of the recommendations. Hybrid approaches can be implemented in several ways: by using the CF approaches and the content-based approaches to make recommendations separately and then combining them; by adding the content-based approaches capabilities to the CF approaches; or by unifying these two approaches into one model [Adomavicius and Tuzhilin, 2005].

Among these three approaches in the recommender systems, we notice that the CF approaches are relying on rating profiles of different users. The CF approaches use a historical ratings of similar users to generate recommendations. However, the content-based approaches generate a classifier for the user’s like and dislike and make recommendations according to the user’s previous behavior patterns. There is no need of other users involved in content-based approach while the CF approaches are based on rating profiles from other users. In addition, hybrid models approaches are basically used to address the large-scale and complicated issues in the recommender systems. In the meanwhile hybrid models approaches increase the complexity of the system [Ricci et al., 2011].

4.2 Collaborative Filtering Approach

The term collaborative filtering (CF) is first proposed by Goldberg et al. [1992] in a paper called “Using collaborative filtering to weave an information tapestry”. The CF approaches are popular recommendation algorithms that based on users’ historical ratings or behavior patterns [Goldberg et al., 1992]. The CF approaches are on the basis of these three assumptions: people have similar preferences and interests; their preferences and interests are stable; we can predict the items they are needed according to their past preferences [Zhao and Shang, 2010].

The NCF approaches are effective recommender algorithms and have been widely used in a broad range of personalized recommender systems, such as Amazon, Youtube, Netflix, etc. The NCF approaches are used to identify like-minded users as the neighbors. They make recommendations through analyzing these neighbors preferences and behavior patterns. As mentioned in the previous section, the NCF approaches can be further divided into the user-based collaborative filtering (UCF) approaches and the item-based collaborative filtering (ICF) approaches [Sarwar et al., 2001].

The UCF approaches first identify a small group of users that have similar preferences and then recommend the items that the group commonly shared or used [Karypis, 2001]. The UCF approaches make recommendations on the basis of other users’ ratings or behavior patterns. However, the ICF approaches analyze the historical items to identify relationships between different items, and then use these relationships to generate recommendations for the specific user [Sarwar et al., 2001]. In other words, the UCF approaches make recommendations by calculating similarity coefficient value between users while the ICF approaches make recommendations by calculating similarity coefficient value between items. In addition, one drawback of the UCF approaches is that it is highly dependent on the previous observed ratings from users. Before using the UCF approach, efforts are spent to obtain the user-item matrix information.

The thesis focuses on understanding user, user grouping and how to elicit requirements from users’ feedback. Considering we are aiming to identify a group of similar users as the project’s primary persona, therefore, the UCF approaches are more appropriate to be adapted into the PS approach. The detailed discussion of the UCF approaches is given in following sections.

4.3 User-based Collaborative Filtering Approach

The UCF approaches recommend interesting items to a user based on similar-minded users called neighbors. They predict the relevance of an item for the target user by a linear combination of his/her neighbors' ratings [Herlocker et al., 2002]. The item could consist of products, musics, services and anything ratings could be provided. The similarities among users form the basis of the UCF approaches. Basically, the UCF approaches include three steps: obtaining user-item rating matrix; calculating similarity coefficient value between users and recommending items [Herlocker et al., 2002].

Step1: Obtaining user-item rating matrix

The first step is to obtain a user-item rating matrix *M*. The matrix M is obtained by mapping over each rating pairs of user and item. Each rating is within a numerical scale, e.g. from 1 to 5. In a typical collaborative filtering scenario, we have a list of users $U = \{u_1, u_2, ..., u_k\}$ and a list of items $I = \{i_1, i_2, ..., i_n\}$, where k and n represent the number of users and items respectively. M is a $n \times k$ matrix where $M_{k,n}$ represents the value of item i_n provided by user u_k. Below Equation 4 is an example of a matrix *M*.

$$M_{kn} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \dots & \dots & \dots & \dots \\ m_{k1} & m_{k2} & \dots & m_{kn} \end{pmatrix} \quad (4)$$

Step2: Calculating similarity coefficient value between users

Once the user-item rating matrix is obtained, the second step is to calculate the similarity coefficient value between the specific user and other users to identify that user's nearest neighbors. Similarity calculation between users is a critical step in the UCF approaches. There are many popular algorithms to calculate the similarity coefficient value between two users, such as the cosine-based algorithm and the Pearson correlation coefficient algorithm [Pearson, 1985].

Below Equation 5 is the cosine-based approach and Equation 6 is the Pearson correlation coefficient approach to calculate the similarities between user x and user y. I_{xy} represents the items rated by both user x and user y. The cosine similarity is a measure of similarity between two users that measures the cosine of the angle between them [Huang et al., 2015]. The cosine similarity is commonly used in positive space. The outcome of the cosine similarity algorithm is bounded in [0,1] [Huang et al., 2015]. However, cosine similarity has a problem that some users might

provide higher ratings as a preference, while other users might give lower ratings as a preference. To remove this drawback from vector-based similarity, the Pearson correlation coefficient approach [Pearson, 1985] subtracts average rating for each user from each user's rating for the pair of items. In this way, the Pearson correlation coefficient approach improve cosine-based approach by removing an average rating for each user. The outcome of Pearson approach is bounded in [-1, 1] where 1 represents positive correlation between users, 0 represents zero correlation and -1 represents negative correlation. Considering the users are unknown and their rating preferences are unknown, thus we decide to employ the Person algorithm to calculate similarity coefficient value between users.

$$simil(x,y) = \frac{\sum_{i \in Ixy} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in Ixy} r_{x,i}^{\ 2}} \sqrt{\sum_{i \in Ixy} r_{y,i}^{\ 2}}} \tag{5}$$

$$simil(x,y) = \frac{\sum_{i \in Ixy} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r}_y)^2}} \tag{6}$$

Step 3: Recommending items

In the UCF approaches, once the similarity coefficient values between users are calculated and the similar-minded user groups are identified, the third step is to generate recommendations for a specific user. However, no recommendations will be generated in this thesis. We only focus on how to adapt the similarity calculation algorithm to identify personas.

4.4 Discussion of the UCF Approach

The UCF approaches are widely used in many recommender systems. They are useful for information overloaded scenarios as they are helpful to filter the important information and easily prioritize them. The UCF approaches are not only used in the recommender systems, but also suitable for information filtering in other systems which a user-item matrix could be provided. Likewise, the first step of the PS approach is to generate a user-requirement matrix as well. Therefore, we propose to adapt the UCF approach to the PS approach, and to group users according to their similarity coefficient value. In this thesis, we only adapt the Step 1 (obtaining

user-item rating matrix) and the Step 2 (the Pearson correlation coefficient algorithm) of the UCF approach to the PS approach. There is no predictions and recommendations generated in this thesis.

5 Collaborative Filtering Based Persona-Scenario Approach

Based on the collaborative filtering algorithm and the persona-scenario approach, we propose collaborative filtering based persona-scenario (CFPS) approach to identifying personas. We adapt the Pearson correlation coefficient algorithm to identify the primary persona by calculating similarity coefficient value between users, where the similarities are based on users' preferences for software requirements and their goals. We adapt the CF algorithm to the PS approach by considering following reasons:

The basis of both the PS approach and the CF approach is to generate a user-item matrix. In the PS approach, the items refer to user requirements. As mentioned, the PS approach relies on demographic variables to classify the user groups and the primary persona is identified from those user groups. Adapting the CF approach is helpful to identify personas according to similarity coefficient value between users, which we assume it is a more appropriate way to identifying personas. In addition, the classification of users are very dependent on requirements analysts' experiences. Personas identification varies due to different requirements analysts have various opinions in users analysis. However, the rule to identify personas in CFPS approach is to group users according to the users' similar preferences to the product and their goals. In this way, the identification of personas through collaborative filtering algorithm is more objective and reliable.

We have discussed the importance of requirements elicitation approaches to be scalability to support large-scale projects in the previous chapter. The PS approach is not considered to scale well with a large number user involvement as a lot of interviews and face to face meetings are widely used. It is time consuming when the users number is large. Information are overloaded with a large number user involvement, therefore, it is more difficult for analysts to handle the eliciting processes. While in the CFPS approach, the Pearson correlation coefficient algorithm is used for efficiently identifying the primary person, and it also scales well for a large number user involvement. Further more, we assume the CFPS approach will improve the efficiency of the PS approach and accelerate the requirements elicitation processes.

5.1 The Process of the CFPS Approach

In our approach, we propose to combine the CF approach to the PS approach which aims to identify personas automatically through the Pearson correlation coefficient algorithm. Figure 9 illustrates the process of the CFPS approach. There are basically

four steps to elicit user requirements: preparing survey, obtaining user-requirement rating matrix, identifying the primary persona and eliciting requirements with primary persona and scenarios.

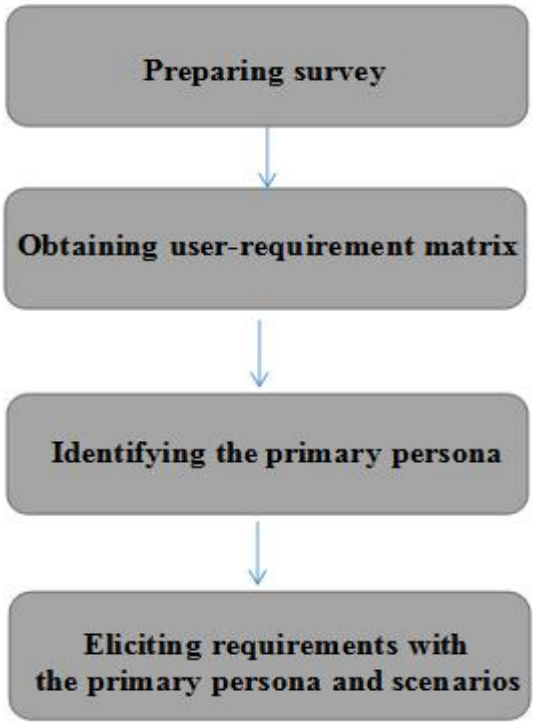

Figure 9: The process of the CFPS approach.

5.1.1 Preparing Survey

The first step in the CFPS approach is to acquire a list of initial requirements as well as users' goals and behavior patterns information. The questionnaires are generally discussed and proposed by stakeholders such as requirements analysts, sponsors, developers, or brainstorming sessions are arranged to gather ideas. The survey not only contains user requirements to the product, but also questions about users' goals, behavior patterns and personalities. Therefore, in this step we will obtain a set of questions of requirements and users' goals information. As the survey is mostly consist of descriptions of requirements, and to align with the PS approach, we use R to represent the collection of requirements from r_1 to r_n and their goals from g_1 to g_n. Therefore, we will obtain survey collection $R = \{r_1, r_2, ..., r_n, g_1, g_2, ..., g_n\}$. Once the survey is ready, users will be asked to provide ratings to each question according to their preferences and frequently usage of the requirements.

5.1.2 Obtaining User-Requirement Rating Matrix

In this step, users will be asked to fill out the survey designed in the first step. Users, might be volunteers or randomly collected from internet, in the group $U = \{u_1, u_2, ..., u_k\}$ provide ratings to each requirement to obtain user-requirement rating matrix, where k is the number of users. As shown in Equation 7, M represents the user-requirement rating matrix of where n refers to the number of requirements and k refers to the number of users.

$$M_{kn} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \dots & \dots & \dots & \dots \\ m_{k1} & m_{k2} & \dots & m_{kn} \end{pmatrix} \quad (7)$$

5.1.3 Identifying the Primary Persona

Identifying the primary persona is the core section of the CFPS approach. After the user-requirement matrix is generated, we use the Pearson correlation coefficient approach to calculate the similarity coefficient value between users. For each user in the user group $U = \{u_1, u_2, ..., u_k\}$, we calculate the similarities between the representative user and other users. The method we used to identify the representative users is based on Wiegers [2003] user classification: favored users, disfavored users and other users. We identify the user with the highest total rating value of all survey questions is the representative favored user. Similarly, the user with the lowest total rating value of all survey questions is the representative disfavored user. Table 3 is a sample data of user-requirement matrix display in the database. For example, as shown in Table 3, user B obviously is the representative favored user, while user C is the representative disfavored user.

User/Requirement	R1	R2	R3	G1	Total
User A	3	4	3	2	12
User B	4	3	5	5	**17**
User C	3	2	1	3	**9**
User D	2	4	4	3	13
User E	3	3	4	4	14

Table 3: Sample data of user-requirement rating matrix.

Once the representative users are identified, we continue to apply the Pearson correlation coefficient algorithm to calculate similarity coefficient value between the representative favored user and the rest of users and the representative disfavored user

and the rest of users respectively. Therefore, similarity coefficient value result, prioritized in descending order, will be generated like the example illustrated in Table 4. In this way, we are able to easily identify top K similar users for the representative favored user as favored persona, where K is the number of users that analysts want to choose as the persona. Likewise, we identify top K similar users for the representative disfavored user as disfavored persona. For the rest of users neither belong to favored persona nor disfavored persona are identified as the other persona. The reason we classify these three types of personas is to find out variant requirements (requirements hot spots) through comparing three personas' preference values. Requirements hot spots are basically characterized by high standard deviation value among different user groups. Identifying variant requirements is helpful to improve the existing software services and elicit new user requirements.

user/similarity	similarity coefficient value
User A	0.9
User B	0.8
User C	0.7
...	...

Table 4: Sample data of similarity coefficient value of favored user.

In this step, we identify three types of personas: favored persona, disfavored persona and other persona. Favored persona represents the target users of product who are interested in the software product and use it frequently. Disfavored persona are those users who are not supposed to use the product and have little interest to it. Other persona represents users that use the product in an average frequency, and they are neither favor nor disfavor the product. In the CFPS approach, the primary persona represents the core users or users who use the core features. Therefore, we identify favored persona as the primary persona.

5.1.4 Eliciting Requirements with Primary Persona and Scenarios

This step is similar to the PS approach. The primary persona is asked to participate in some eliciting activities, such as interviews, brainstorming sessions and scenarios activities. However, we mainly focus on the identification of personas in this thesis. Thus, this part is not our focus.

5.2 Discussion of the CFPS Approach

The CFPS approach combines collaborative filtering algorithm into the PS approach to eliciting user requirements in an efficient manner, specifically, the CFPS approach

adapts the Pearson correlation coefficient algorithm to identify the primary persona. In this way, there is no necessary to classify users into different user groups in advanced. Efforts of user classification are saved in the CFPS approach. Further, the CFPS approach reduce the dependency on analysts' personal experiences.

Comparing with the PS approach, the CFPS approach add an extra discussion regarding information of users' goals, behaviors and motivations. The personas identified in the CFPS approach is on the basis of both their preferences to the product and their goals and motivations. We believe this information is helpful to enrich and complete the personas. The CFPS approach is adapted in a case study in Chapter 6. In order to compare and evaluate the CFPS approach, we adapt the PS approach as well in the case study.

6 Case Study: Eliciting requirements for a Travel Website

The case study is designed to elicit user requirements for a travel website. With the development of internet, it is very common to search the travel information through travel websites [Lai et al., 2007]. Over decades of internet development, travel websites have been the effective information facilities for both individual and business travelers. According to China Tourism Academy, until 2010, Chinese online travel services had attracted over 190 million active users and the market transaction scale had been up to 200 billion RMB, nearly one quarter share of total tourism revenue in China [Zhang and Zhang, 2012].

It is important to be an outstanding travel website among the intense competitions. Researches have shown that improving services quality of a travel website is one successful strategy [Lai et al., 2007]. Therefore, correctly eliciting user requirements and providing satisfied services are critical for a travel website. Correct, unambiguous and complete user requirements are helpful to ensure that the services provided by the travel website are satisfied by most of users. The reason of choosing a travel website as a case study is considering that travel websites are used by a large number of various users and they are typical large-scale projects. Our goal for this case study is to identify primary persona for a travel website and elicit user requirements by employing the CFPS approach with as many users involvement as possible. We evaluate whether the CFPS approach is able to elicit user requirements in an objective manner and improve the efficiency of requirements eliciting process.

We develop a web survey tool named “cfps-survey” which is deployed to Google app engine: http://cfps-survey.appspot.com. One brainstorming session is arranged for discussing the design of survey among requirements analysts, users, travel website sponsors and developers. The result of the survey is imported directly into our database. In the cfps-survey, users are asked to fill in their basic information (e.g. name, age, gender, marriage status, job, etc.), provide ratings to requirement questions from 1 to 5 according to their usage frequency and preferences, and provide ratings to questions regarding their goals and behaviors. The design of the cfps-survey can be found in Appendix 1. The case study is applied in both the CFPS approach and the PS approach with 60 users. We invite 30 volunteers to participate in the survey and the rest users are unknown internet users who are invited by these volunteers via sharing the survey link in social network or by email. Section 6.1 introduces how to elicit user requirements in the CPFS approach while Section 6.2 is the process of the case study in the PS approach. Section 6.3 evaluates and compares these two approaches.

6.1 Case Study with the CFPS Approach

(1) Preparing Survey

The questionnaire is composed of three parts: basic information of a user, user requirements for the product, and users' goals, behaviors and personalities regarding the product under development. A user's basic information include name, age, gender, job, hobbies, etc. This information is not used to identify personas. However, they are helpful to understand users and create persona profile. Both user requirements and users' goals, behaviors and personalities are used to identify the personas.

R1	Providing travel package
R2	Providing special price flights
R3	Providing special price hotels
R4	Providing Visa application information
R5	Providing shopping discount information for all destinations
R6	Providing weather information all over the world
R7	Providing currency exchange information
R8	Supporting travel blogs post and share
R9	Supporting air plane booking
R10	Supporting hotel booking
R11	Supporting travel insurance services
R12	Supporting multiple payment methods
R13	Supporting search function by destinations, travelers...
R14	Supporting car rental services
R15	Supporting forum
R16	Recommending travel companions
R17	Recommending local tour guide
R18	Recommending popular travel blogs and destinations
Goals	Most agree to choose 5, most disagree to choose 1.
G1	Do you like travel?
G2	Do you browse the travel website frequently?
G3	Basically you use the website for booking the business trip?
G4	Basically you use the website for booking the personal trip?
G5	Do you have money but little time for traveling?
G6	Do you have time but little money for traveling?
G7	Do you plan a traveling more than twice a year?

Table 5: Survey for a travel website.

As shown in Table 5, there are 18 user requirements for the the travel website development and 7 questions to facilitate the analysis of goals, behaviors and personalities from stakeholders. Users are asked to rate each question from 1 to 5, with 1 being most infrequently used or most disagreed and 5 being most frequently used or most agreed.

(2) Obtaining User-Requirement Rating Matrix

We have 60 volunteers to participate in the survey. We invite 30 volunteers and another 30 volunteers are randomly invited from internet by sharing the survey through social network or email. Their information is summarized in Table 6. These users rate each question in the survey, and the rating result is imported in our database. In this step, we obtain a user-requirement matrix with 60 users and 25 survey questions (18 requirement questions and 7 users' goals questions).

Category	Male	Female	Total
age 0-17	4	5	9
age 18-28	11	14	25
age 29-39	8	9	17
age 40+	4	5	9
Total	27	33	60

Table 6: Participating users basic information.

(3) Identifying the Primary Persona

In our database, we store data of each user's ratings. By aggregating each user's the total rating value for all questions, we can identify two users among the 60 individuals who answered the questionnaires: with the highest total rating value and with the lowest total rating value. The user with the highest total rating value is identified as the representative favored user; likewise, the user with the lowest total rating value is identified as the representative disfavored user. The information of these two representative users is summarized in the Table 7. We can see the representative favored user is a single lady at age 29, and her job is a designer. She is an active user of the travel website and she plans to travel more than twice a year. While the most disfavored user is also a female at age 16, she infrequently uses the travel website even thought she likes traveling (she rates 4 to the question G7).

User	Age	Job	Gender	Single	Ratings for requirements	Ratings for goals	Total rating
Favored user	29	designer	female	yes	81	30	111
Disfavored user	16	student	female	yes	36	12	48

Table 7: Information of the representative users.

Once these two representative users are identified, we adapt the Pearson correlation coefficient algorithm (Equation 6) to calculate the similarity coefficient values between these two representative users and other users respectively. As mentioned in Subsection 5.1.3, analysts could choose top K similar users of the representative users to form the personas. Considering we have 60 users in total, thus we decide to choose top 19 similar users of the representative favored user and disfavored user so that we have equally 20 users for each persona. Thus, we choose top 19 similar users of the representative favored user and top 19 similar users of the representative disfavored user, together with the representative favored user and representative disfavored user to compose the favored persona and disfavored persona. For the rest 20 users are identified as other persona. Other persona in this thesis refers to those users are average interested in the travel website. They are neither too like nor dislike traveling. Other persona represents a group of potential users who might not be the customers at this moment, but they might be attracted through improving the services of travel website.

Through analyzing these three personas' preference values, it is helpful to identify the variant requirements which are characterized by high standard deviation value. Table 8 presents the statistics about 25 questions rated by these three personas. We notice the total rating value of requirements and goals questions provided by the primary persona (favored persona), 98.95, significantly outperforms than values rated by other persona and disfavored persona. The total value rated by other persona, 79.75, is in the average while the total value rated by disfavored persona is the lowest among these three personas.

Questions	Favored persona	Other persona	Disfavored persona	Ave.	SD
R1	4.15	3.8	3.1	3.683	0.388
R2	4.4	3.75	2.8	3.65	0.566
R3	4.35	3.7	2.8	3.616	0.544
R4	4.1	3.7	3.15	3.65	0.333
R5	4.35	3.75	3.3	3.8	0.366
R6	3.9	3.15	2.6	3.216	0.455
R7	4.1	3.2	2.3	3.2	0.6
R8	4.45	3.65	3.55	3.883	0.377
R9	4.15	3.55	2.2	3.3	0.733
R10	4.15	3.7	2.1	3.316	0.811
R11	3.2	2.6	1.55	2.45	0.6
R12	4.05	3.1	1.2	2.783	1.055
R13	4.55	4.1	3.1	3.72	0.556
R14	3.8	2.15	1.45	2.466	0.888
R15	3.65	2.4	1.7	2.583	0.711
R16	3.7	2.35	1.35	2.466	0.822
R17	4.1	2.45	1.3	2.616	0.988
R18	4.25	3.75	3.25	3.75	0.333
Total Req.	**73.4**	**58.25**	**42.8**	-	-
Ave.	4.07	3.27	2.37	3.24	**0.618**
G1	4.3	3.65	3.3	3.75	0.366
G2	4.45	3.5	2.4	3.45	0.7
G3	2.6	2.15	1.25	2	0.5
G4	4.2	3.4	3.15	3.583	0.411
G5	3.75	2.45	1.45	2.55	0.8
G6	2.3	3.35	3.25	2.966	0.444
G7	3.95	2.4	1.35	2.566	0.922
Total	**98.95**	**79.75**	**58.95**	-	-

Table 8: Preference value analysis of personas in the CFPS approach.

As discussed in the previous chapter, the favored persona is identified as the primary persona. Table 9 presents the statistics about 20 users in the primary persona. They are mostly from the age of 18 to 39, including single and married users from different genders and different backgrounds. We contacted them via email or face to face meetings to better understand them. Trough further interactions with the primary persona, we found out some common characters about the primary persona: (1) They have stable jobs and most of them work for big scale companies with regular annual leaves, most of them are IT engineers, accountants, photographers, designers, etc. (2) Most of them have good education backgrounds, they speak well English and prefer to make the travel plan themselves. (3) Most of them do not have children, and average they travel more than twice a year. (4) They prefer to browse the traveling blogs shared by other travelers before making their travel plans. Based on the primary persona we identified by the CFPS approach and interactions with them, we created the primary persona profile as shown in Figure 10.

	Age 18-28		Age 29-39		Age 40+	
	male	female	male	female	male	female
primary persona	5	6	3	4	1	1

Table 9: Information about the CFPS primary persona.

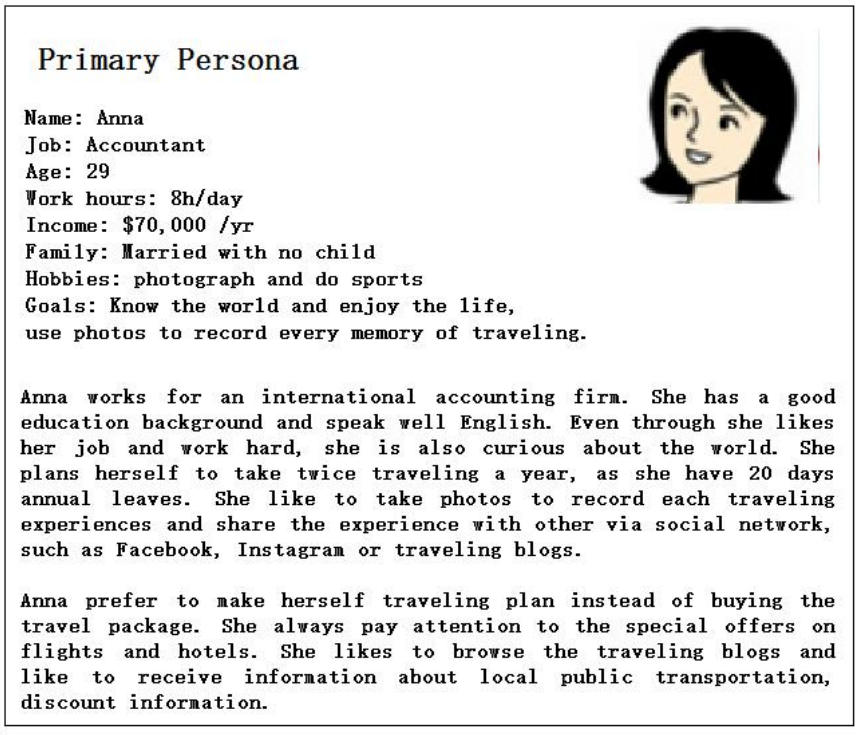

Figure 10: The primary persona identified by the CFPS approach.

(4) Eliciting Requirements by Primary Persona and Scenarios

Once the primary persona is identified, we analyze user requirements according to preference values provided from different personas. Our final requirements consist of variant requirements and common requirements. The variant requirements, characterized by high standard deviation values, are our requirements hot spots. The common requirements are characterized by high average value and low standard deviation value.

As shown in Table 8, we identify common requirements include R1, R2, R3, R4, R5, R6, R7, R8, R13, and R18. We identify the requirements hot spots include R9 (Supporting air plane booking), R10 (Supporting hotel booking), R11 (Supporting travel insurance services), R12 (Supporting multiple payment methods), R14 (Supporting car rental services), R16 (Recommending travel companions) and R17 (Recommending local tour guide). Through interactions with the primary persona on the scenarios activities, we further figure out the users' various opinions on these requirements hot spots. For example, users have different opinions to R10 (Supporting hotel booking) because favored persona rates 4.15 and disfavored persona rates only 2.1. Though interactions with the personas, we find out the disfavored persona mostly consists of the single young users and they prefer to solve the accommodation problems through couch surfing or Airbnb rather than booking a hotel. Another example, users have various opinions on the R11 (Supporting travel insurance services) because some of them have already bought complete insurances before; some users trust the professional insurance websites rather than the travel websites; and others consider there is no necessary to buy a travel insurance. To improve this requirement, we gather some valuable information and elicit new requirements: (1) The travel website should remind users of the importance of purchasing travel insurances. (2) The travel website should convince users that the insurance services provided in the website are trustable and reliable.

Through the interactions with the primary persona, we find out identifying the correct primary person is critical for eliciting correct user requirements, as following requirements elicitation activities are interacted with the primary persona and their opinions might lead to the different improvements of the travel website services.

6.2 Case Study with the PS Approach

In order to compare and evaluate the CFPS approach, we apply the primary persona identification steps in the PS approach to the case study as well. According to the demographic variables, we classify users by considering following factors: age and

marriage status. Therefore we category three user groups: group A represents single young users less than 28 years old; group B represents married users less than 28 years old; group C represents users over 29 years old. Considering 19 users are married and only 7 users are single in group C, thus we do not further classify marriage status in group C. Table 10 illustrates these three user groups preferences to the 18 requirements questions. The PS approach identifies the user group with the highest total rating value for all requirements as the primary persona. Therefore, group C, slightly outperforms than group B, is identified as the primary persona in the PS approach.

Requirements	Group A	Group B	Group C	Ave.	SD
R1	3.3	3.65	4.15	3.7	0.3
R2	3.4	3.8	3.85	3.68	0.189
R3	3.4	3.75	3.7	3.61	0.144
R4	3.35	3.8	3.8	3.65	0.2
R5	3.6	4.05	3.4	3.68	0.244
R6	2.95	3.1	3.55	3.2	0.233
R7	2.9	3.05	3.6	3.18	0.277
R8	3.85	3.7	4.1	3.88	0.144
R9	3.1	3.55	3.2	3.28	0.177
R10	2.45	3.7	3.85	3.33	0.588
R11	1.9	3.05	3.3	2.75	0.566
R12	2.1	3.3	3.25	2.88	0.522
R13	3.85	4.05	4.15	4.01	0.111
R14	3.65	2.2	2.15	2.67	0.655
R15	3.15	2.45	3.05	2.88	0.288
R16	3.6	1.8	2.0	2.47	0.75
R17	2.45	3.7	3.1	3.08	0.422
R18	3.7	4.1	3.65	3.82	0.188
Total	**56.7**	**60.8**	**61.85**	-	-
Ave.	3.15	3.37	3.43	3.31	**0.33**

Table 10: Preference value analysis in the PS approach.

We contacted the primary persona via email and interviews, and concluded some common characters of the primary persona: (1) Most of them have stable jobs and sound financial bases. They are able to afford traveling more than twice a year. (2) Most of them travel with their families and the major purpose is to enjoy the family time and to broaden their children's horizons through traveling. (3) They prefer the traveling packages rather than planning the trip by themselves. (4) They prefer safe travel destinations rather than adventures. (5) They value the quality of the traveling over the price. According to above common characters, we created the primary persona profile as shown in Figure 11.

From the statistics presented in Table 10, we are able to identify common requirements include R1, R2, R3, R4, R5, R6, R7, R8, R9, R13, R17, and R18. We notice the standard deviation value is not significant among these three user groups in the PS approach. We identify requirements with relatively high standard deviation value as the hot spots of requirements, including R10 (Supporting hotel booking), R11 (Supporting travel insurance services), R12 (Supporting multiple payment methods), R14 (Supporting car rental services) and R16 (Recommending travel companions).

Figure 11: The primary persona identified by the PS approach.

6.3 Evaluation and Discussion

This section evaluates the results of the case study. Firstly, we discuss the primary persona identified in both CFPS and PS approaches. Secondly, we analyze the requirements hot spots identified in both approaches. Lastly, we evaluate the CFPS approach as well as its achievements and limitations.

6.3.1 The Primary Persona Analysis

In the previous section, we apply both the CFPS approach and the PS approach to the case study, a travel website, to identify the primary persona and elicit requirements hot spots through interacting with the primary persona. Thus, we have two primary personas identified by these two approaches.

The CFPS primary persona is identified by their similar preferences to the travel website as well as their goals and behaviors. Firstly the representative favored user is identified, then the rest 19 similar users are calculated and together with the representative favored user are identified as the primary persona. As shown in Table 11, the CFPS persona consists of 20 users. They are from different ages and most of them are from 18 to 39 years old. Through interviewing with the CFPS persona, we discover the common characteristics of them which have mentioned in the previous section. They are from different ages, but they have good education backgrounds, stable jobs and they are interested in traveling. Their motivations are to take a break from regular work and enjoy the life with families or themselves. They prefer to planning their trip by themselves rather than purchasing a travel package.

The PS primary persona is a user group that represents users over 29 years old (17 users are from age 29-39 and 9 users are over 40 years old)，as shown in Table 11. As this user group's total rating value for all requirements is slightly higher than other groups, it is identified as the primary persona. Through interviewing with the PS primary persona, we discover that most of them are married and with children. Their traveling motivations are related to their families. Some users plan a traveling for family as the make up of busy work and they want to enjoy the time with family; other users plan the family trip for children to broaden children's horizons. They prefer to purchasing the travel packages rather than planning their trip by themselves, and they value the quality and safety of the traveling over than the price.

It is obvious that different approaches identify different primary personas and they have different opinions and comments that could affect on the requirements analysis for the product. For example, according to the PS primary persona, the travel website

might provide more safe and good quality travel packages to attract their target users. However, according to the CFPS primary persona, the travel website might provide more special low price offers for the flights and hotels booking. Therefore, it is very critical to accurately identify the primary persona for the product.

	Age 18-28		Age 29-39		Age 40+		Total
	male	female	male	female	male	female	
CFPS persona	5	6	3	4	1	1	20
PS persona	0	0	8	9	4	5	26

Table 11: Statistics of primary personas in CFPS approach and PS approach.

The CFPS primary persona contains users from different ages while the PS primary persona represents opinions from the perspective of the relative older users. The CFPS primary persona consists of each single individual who is interested in traveling and frequently use the travel website. The PS primary persona represents that the older user group are likely be the core target users than other user groups. The main reason that the older users are identified as the primary persona in the PS approach is because of their stable economic bases. We consider the CFPS approach identifies the primary persona in the user level while the PS approach identifies the primary persona in a more general user group level.

To analyze the primary persona that accurately represent the core target users of the travel website, we compare their preferences statistics of the travel website as shown in Figure 12. The preference statistics of the CFPS primary persona is slightly over than the preference statistics of the PS primary persona. To some extent, we consider the CFPS primary persona are more interested in the travel website and they more frequently use the travel website than the PS primary persona. As illustrated in Table 8, the CFPS primary persona's average rating value to the question G1 (Do you like traveling?) is 4.3 and to the question G7 (Do you plan a traveling more than twice a year?) is 3.95. To prove this conclusion, likewise, we request the PS primary persona to rate these two questions in the interview sessions, their ratings are 3.85 and 2.7 on average. In this way, we consider the CFPS primary persona is better to represent the target users of the travel website.

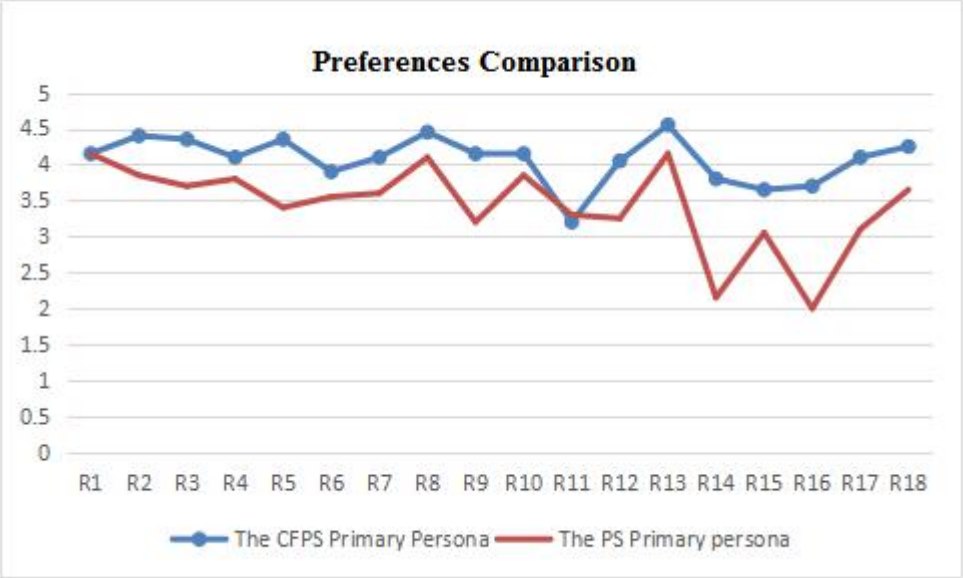

Figure 12: Preferences comparison of requirements.

6.3.2 The Requirements Hot Spots Analysis

We find it easier to identify the requirements hot spots with the CFPS approach than the PS approach. Both these two approaches identify R10, R11, R12, R14 and R16 are the requirements hot spots. Additionally, the CFPS approach identify R9 (Supporting air plane booking) and R17 (Recommending local tour guide) as the requirements hot spots. The reason of the high stand deviation values for question R9 and R17 in the CFPS approach is that the favored persona provides good ratings (4.15 and 4.1 respectively) while the disfavored persona provides very low ratings to these two requirements (2.2 and 1.3 respectively).

Regarding the requirement R9 (Supporting air plane booking), the favored persona uses the service frequently while the disfavored persona hardly uses it. We notice that disfavored persona mainly consists of young students or users with unstable jobs. Considering their economic status, they prefer short distance trips and they prefer traveling by bus, train or even by bicycle. Regarding the requirement R17 (Recommending local tour guide), the CFPS favored persona likes local tour guide service as sometimes they travel alone and they expect to have a better and deeper experience of the local culture while the disfavored persona barely uses the local tour guide service under the consideration of cost saving. However, the PS primary persona has an average attitude to the local tour guide service as they usually purchase the travel package which a tour guide is included in the package.

According to the statistics illustrated in Table 8 and Table 10, the average standard deviation values in the CFPS approach and PS approach are 0.618 and 0.33 respectively. The user classification in the CFPS approach is on the basis of similarity coefficient value between users, in this way, users with relative higher rating values are grouped together and users with relative lower rating values are grouped together. While in the PS approach, users are grouped on the basis of age and marriage status. Therefore, the standard deviation values calculated in the CFPS approach are significant than the PS approach. Regarding requirements hot spots identification, we consider the CFPS approach performs better than the PS approach.

6.3.3 Pearson Correlation Coefficient Analysis

In the CFPS, we adapt the Pearson correlation coefficient algorithm to calculate similarity coefficient values between the representative user and other users according to their ratings to the requirements and product goals. Appendix 2 presents the statistics of the CFPS primary persona's ratings to each question. In the case study, the average similarity coefficient value among the primary persona is 0.46. Figure 13 illustrates the relationship between the CFPS primary persona's total rating value and similarity coefficient value. We notice that users with higher rating value are likely to have a higher similarity coefficient value of the representative favored user, but it is not absolutely. Users with total rating value about 100 have various similarity coefficient values from 0.2 to 0.7. The Pearson correlation coefficient algorithm focuses more on users' opinions on each single question rather than their total rating values. However, the PS approach focuses on the total rating values provided by user groups. In this way, we consider the Pearson approach is a more accurate manner to group users as it is able to identify users from a single question level rather than from a high level.

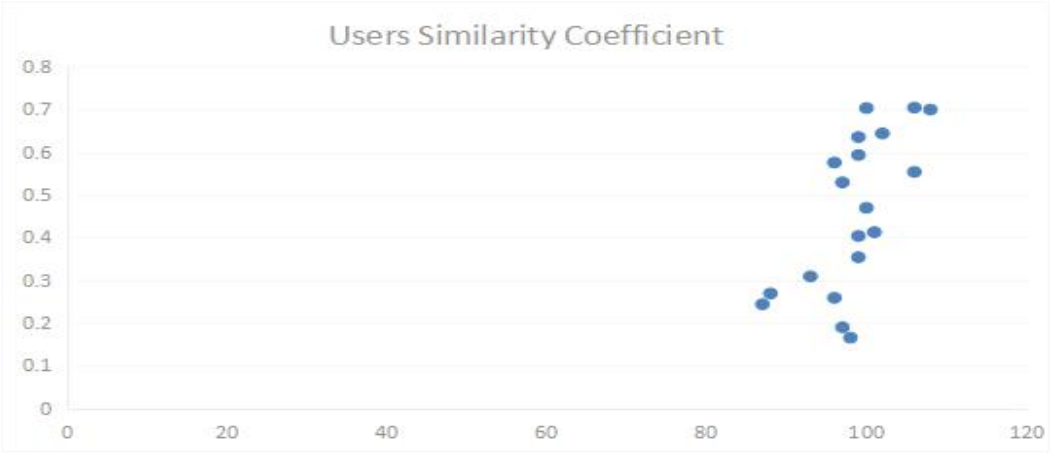

Figure 13: The similarity coefficient values of the CFPS primary persona.

In addition, we find out some users passively participated in the survey as they rate 3 and 4 to all questions. But Pearson correlation coefficient algorithm is helpful to identify this useless data. When user provides the same rating score to all questions, the result of the numerator in Pearson equation is zero. It means the user is zero correlation to the representative user and that user's rating data could be easily identified as the useless subject. In this way, we avoid to use this passively data in personas identification and further ensure the accuracy of identification of primary persona.

6.3.4 The CFPS Approach Analysis

The CFPS approach combines the collaborative filtering algorithm to the PS approach for personas identification. Through successfully applying the CFPS approach to the travel website to identify personas and elicit user requirements, the CFPS is considered as a useful approach to identifying personas. We consider there are mainly three achievements of proposing the CFPS approach as following:

The CFPS approach adapts the collaborative filtering algorithm for personas identification according to users similar preferences to the software product as well as their goals and behavior patterns regarding the product. In the PS approach, users are firstly classified into different user groups according to demographic variables, such as age, gender, job, marriage status, etc. In this way, different requirements analysts will identify different personas as they do not have a standard to identify personas. Thus, personas identification is basically with a problem of being too subjective. With the CFPS approach, users are grouped on the basis of their similarities. We first identify the representative favored user and use the Pearson correlation coefficient algorithm to calculate similar users. In this way, even different analysts could identify the same personas as long as the number of users of the persona is predefined. We consider the CFPS approach, to some extent, is a solution for the problem of personas are being subjectively identified.

Comparing with the PS approach, we add discussion of users' goals, behaviors, personalities regarding the product and this information also affect the identification of personas. As mentioned, a persona is created by analyzing the real users' goals, behaviors and motivations. Gathering this information has two benefits. Firstly, they help to group users on the basis of both requirements of the product and similar goals of the product. Requirements analysts usually create a persona profile once the personas are identified. With understanding the users' goals, behaviors and motivations, it is helpful to understand users and create persona profile. Secondly,

user information regarding their goals and motivations are important information that analysts should be aware in advanced before conducting requirements eliciting activities, as they are the underlying reasons to explain users' behaviors.

Comparing with the CA approach, we add discussion regarding users' preferences to the software requirements as considering the information about users' goals and behavior patterns is too abstract to classify users. The CA approach does not provide a definitive answer for how many user groups should be clustered and it is up to decisions of analysts. In addition, the CA approach has a significant time complexity problem in computation which is hard to apply in practical projects. In the CFPS approach, we adapt the collaborative filtering algorithm to classify users into three personas in an automatic and efficient manner. We consider that the primary persona identified in the CFPS approach is more accurately to represent target users. The CFPS approach is easier applied in the practical projects than the CA approach.

The CFPS approach improves the efficiency of personas identification comparing to the PS approach through adapting the collaborative filtering algorithm. The representative favored user, disfavored user and their similar users are automatically calculated. In the PS approach, requirements analysts are needed to manually classify users into different user groups. It is a tedious process to handle when the users volume is large. With the CFPS approach, even if thousands of users are involved, it is efficient to identify the personas. In this way, we consider the CFPS is an efficient approach which scales well with a large number user involvement.

6.3.5 The Limitations

Regarding the design of user survey, it would be better to include more questions about the travel website requirements, users' goals and behaviors. Considering the users are volunteers and most of them are invited via internet, they might give the survey up if we put too many questions on the survey. Therefore, the survey is very limited to gather information in this case study. In addition, as we were adapting the Pearson correlation coefficient algorithm in similarity calculation, it is better to provide a broader rating scale when designing the survey, for example the rating scale from 1 to 7 or from 1 to 10. A broader rating scale is helpful for accurately calculating similarity coefficient value between users and increasing the accuracy of the personas identification.

In addition, some users passively participated in the survey. For example, one user rated 3 for all questions. Some email addresses are not correct and we were not able to

contact those users for further interviews. In the future, we are looking for using other interesting approaches to conduct the survey and attract users' interests. For example, we could design a gamification survey to allow volunteers proactively participate in the requirements elicitation processes.

In this case study, data collected from 60 users is rather limited for real statistics analysis. Our goal is to support a large number user involvement in the personas identification. However, only 60 users were invited successfully to participate in the case study. We are looking for opportunities to use a larger data set to evaluate the CFPS approach in the future.

7 Conclusion

In this thesis, the CFPS approach is proposed to identify personas in a more efficient and objective manner and supports a large number user involvement. The CFPS approach adapts the collaborative filtering algorithm to the PS approach to identify personas by calculating similarity coefficient value between users.

The contribution in this research, as mentioned in Chapter 6, can be divided into three parts. Firstly, the CFPS approach adapts the collaborative filtering algorithm to improve the objectiveness of the personas identification. Secondly, discussion regarding users' goals, motivations and behaviors are helpful to understand persona and persona profile creation. Lastly, the CFPS approach improves the efficiency and accuracy of personas identification which supports a large number user involvement in the requirements eliciting processes.

According to discussion in the Subsection 6.3.5, there are still some limitations in this thesis. The further work could focus on how to improve the survey design and allow users proactively involved in the requirements eliciting processes. In the future, we are looking forward to find out some better algorithms to identify personas and improve user requirements elicitation processes.

Reference

[Adomavicius and Tuzhilin, 2005] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* **17** (6), 734-749, 2005.

[Albrecht, 1979] A. J. Albrecht, Measuring application development productivity. In: *Proc. of the Joint Share/ Guide Symposium,* 83–92, 1979.

[Alexander and Maiden, 2004] I. F. Alexander and N. Maiden, Scenarios, Stories, Use Cases, through the Systems Development Life Cycle, *John Wiley & Sons,* 2004.

[Aoyama, 2005] M. Aoyama, Persona-and-scenario based requirements engineering for software embedded in digital consumer products. In: *Proc. of the 13th IEEE International Conference on Requirements Engineering*, 2005.

[Bano and Zowghi, 2015] M. Bano and D. Zowghi, A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, **58**, 148-169, 2015.

[Brooks, 1995] F. P. Brooks, *The Mythical Man-Month.* Addison-Wesley, 1995.

[Burstin and Ben-Bassat, 1984] M. Burstin and M. Ben-Bassat, A user's approach to requirements analysis of a large software system. In: *Proc. of the 1984 Annual Conference of the ACM on the Fifth Generation Challenge,* 133-145, 1984.

[Carroll, 1991] A. B. Carroll, The pyramid of corporate social responsibility: toward the moral management of organizational stakeholders. *Business Horizons*, **34**, 39-48, 1991.

[Cleland-Huang and Mobasher, 2008] J. Cleland-Huang and B. Mobasher, Using data mining and recommender systems to scale up the requirements process. In: *Proc. of the 2nd International Workshop on Ultra-Large-Scale Software-Intensive Systems,* 3-6, 2008.

[Cockburn, 2001] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley, 2001.

[Cooper, 1999] A. Cooper, *The Inmates Are Running the Asylum.* SAMS, 1999.

[Courage and Baxter, 2005] C. Courage and K. Baxter, *Understanding Your Users :A Practical Guide to User Requirements Methods, Tools, and Techniques*. Elsevier Inc, 2005.

[Davis, 1992] A. M. Davis, Operational prototyping: a new development approach. *IEEE Software* **9**(5), 70-78, 1992.

[Freeman, 1984] R. E. Freeman, *Strategic Management: a Stakeholder Approach.* Pitman Press, 1984.

[Gause and Lawrence, 1999] D. Gause and B. Lawrence, User-Driven Design. *Software Testing & Quality Engineering* **1**(1), 22–28, 1999.

[Goldberg et al., 1992] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, Using collaborative filtering to weave an information tapestry. *Communications of the ACM* **35**(12), 61-70, 1992.

[Grudin and Pruitt, 2002] J. Grudin and J. Pruitt, Personas, participatory design and product development: an infrastructure for engagement. In: *Proc. of the Participartory Design Conference*, 144-161, 2002.

[Halme and Kallio, 2014] M. Halme and M. Kallio, Likelihood estimation of consumer preferences in choice-based conjoint analysis. *European Journal of Operational Research* **239** (2), 556–564, 2014.

[Hauser and Rao, 2005] J. R. Hauser and V. R. Rao, Conjoint analysis, related modeling, and applications. In: Y. Wind and P. E. Green, *International Series in Quantitative Marketing* **14**. Springer, 141-168, 2005.

[Herlocker et al., 2002] J. Herlocker, J. A. Konstan and J. Reidl, An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval Journal* **5**(4), 287-310, 2002.

[Herrmann and Nolte, 2010] T. Herrmann and A. Nolte, The integration of collaborative process modeling and electronic brainstorming in co-located meetings. In: *Proc. of the* C*ollaboration and Technology,* 145-160, 2010.

[Huang et al., 2015] S. Huang, S. Wang, T. Liu, J. Ma and J. Veijalainen, Listwise collaborative filtering. In: *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval,* 343-352, 2015.

[IEEE Standard 29148, 2011] Systems and software engineering - Life cycle processes - Requirements engineering. ISO/IEC/IEEE, 2011. Available as https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:29148:ed-1:v1:en.

[Karypis, 2001] G. Karypis, Evaluation of item-based top N recommendation algorithms. In: *Proc. of the* 10*th International Conference on Information and Knowledge Management* **22**(1), 247-254, 2001.

[Kitzinger, 1994] J. Kitzinger, The methodology of focus groups: the importance of interaction between research participants. *Sociology of Health & Illness* **16** (1)*,* 103-121, 1994.

[Kunifuji et al., 2007] S. Kunifuji, N. Kato and A. P. Wierzbicki, Creativity support in brainstorming. *Creative Environments Studies in Computational Intelligence*, 93-126, 2007.

[Lai et al., 2007] C. Lai, C. Chen and P. Lin. The Effects of Service Quality on Customer Relational Benefits in Travel Website. In: *Proc. of the 2007 Portland*

International Conference on Management of Engineering & Technology, 93-126, 2007.

[Lim et al., 2012] S. L. Lim, D. Quercia and A. Finkelstein, StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation. *IEEE Transactions on Software Engineering* **38** (3), 707-735, 2012.

[Low and Jeffery, 1990] G. C. Low and D. R. Jeffery, Function points in the estimation and evaluation of the software process. *IEEE Transactions on Software Engineering* **16** (1), 64-71, 1990.

[Luce and Tukey, 1964] R. D. Luce and J. W. Tukey, Simultaneous conjoint measurement: a new type of fundamental measurement. *Journal of Mathematical Psychology* **1** (1), 1-27, 1964.

[McConnell, 2004] S. McConnell, *Code Complete (2nd ed.).* Microsoft Press, 2004.

[Moore et al., 2001] J. W. Moore, B. Pierre, D. Robert, A. Alain and T. Leonard, G*uide to the Software Engineering Body of Knowledge.* IEEE Computer Society Press, 2011.

[Nunamaker, 1991] J. Nunamaker, A. Dennis, V. Joseph, V. Doug and G. Joey, Electronic meeting systems to support group work. *Communications of the ACM* **34** (7), 40–61, 1991.

[Nuseibeh and Esterbrook, 2000] B. Nuseibeh and S. Easterbrook, Requirements engineering: a roadmap. In: *Proc. of the* C*onference on the Future of Software Engineering,* 35–46, 2000.

[Pearson, 1985] K. Pearson, Note on regression and inheritance in the case of two parents. In: *Proc. of the Royal Society of London* **58**, 347-352, 1895.

[Ricci et al., 2011] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, Recommender Systems Handbook. *Springer-Verlag New York, Inc.*, 2011.

[Santillan and Käkölä, 2016] M. F. Santillan and T. Käkölä, Evaluation framework for analyzing the applicability of criteria lists for the selection of requirements management tools supporting distributed collaboration and software product line requirements management. In: *Proc. of the 49th Hawaii International Conference on System Sciences*, 5783-5792, 2016.

[Sarwar et al., 2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Item-based collaborative filtering recommendation algorithms. In: *Proc. of the 10th International Conference on World Wide Web*, ACM New York, 285-295, 2001.

[Scott, 1995] A. Scott, Reduce development costs with use-case scenario testing. *Software Development* **3** (7), Miller Freeman Inc., 53–61, 1995.

[Sinha, 2003] R. Sinha. Persona development for information-rich domains. In: *Proc. of the 2003 Extended Abstracts on Human Factors in Computing Systems*, ACM New York, 830-831, 2003.

[Sommerville and Sawyer, 1997] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide.* Wiley, 1997.

[Thayer and Dorfman, 1997] R. H. Thayer and M. Dorfman, *Software Requirements Engineering (2nd ed.).* IEEE Computer Society Press, 1997.

[Tu et al., 2010] N. Tu, X. Dong, P. P. Rau and T. Zhang, Using cluster analysis in persona development. In: *Proc. of the 8th International Conference on Supply Chain Management and Information Systems*, 1-5, 2010.

[Wiegers, 2003] C. Wiegers, *Software Requirements*. Microsoft Press, 2003.

[Yang et al., 2015] H. Yang, G. Ling, Y. Su, M. R. Lyu and I. King, Boosting response aware model-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* **27** (8), 2064-2077, 2015.

[Yuan et al., 2014] Q. Yuan, G. Cong and C. Y. Lin, Com: a generative model for group recommendation. In: *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 163-172, 2014.

[Zave, 1995] P. Zave, Classification of research efforts in requirements engineering. In: *Proc. of the 2nd IEEE International Symposium on Requirements Engineering*, 1995.

[Zhang and Zhang, 2012] Y. Zhang and L. Zhang, The effects of ICT innovation and industry regulation on Chinese travel website's marketing logic. In: *Proc. of the 2012 IEEE 14th International Conference on Commerce and Enterprise Computing*, 86-93, 2012.

[Zhang, 2007] Z. Zhang, Effective requirements development - a comparison of requirements elicitation techniques. In: *Proc. of the SQM2007 Conference*, Tampere, 2007.

[Zhao and Shang, 2010] Z. Zhao and M. Shang, User-based collaborative-filtering recommendation algorithms on Hadoop. In: *Proc. of the 2010 Third International Conference on Knowledge Discovery and Data Mining*, IEEE Computer Society Washington, 478-481, 2010.

[Zowghi and Coulin, 2005] D. Zowghi and C. Coulin, Requirements elicitation: a survey of techniques, approaches, and tools. In: E. Kamsties, *Engineering and Managing Software Requirements.* Springer, 19-46, 2005.

Appendix 1: The CFPS Survey

Please input your personal information before fill out the survey.
Name:
Age:
Gender:
Job:
Email:
Single or Married:
Education Background:
Hobbies:

Please rate below requirements from 1 to 5 for a travel website according to your frequently usage and your preferences, with 1 being no interest at all and 5 being most interested and most frequently used.

R1	Providing travel package
R2	Providing special price flights
R3	Providing special price hotels
R4	Providing Visa application information
R5	Providing shopping discount information for all destinations
R6	Providing weather information all over the world
R7	Providing currency exchange information
R8	Supporting travel blogs post and share
R9	Supporting air plane booking
R10	Supporting hotel booking
R11	Supporting travel insurance services
R12	Supporting multiple payment methods
R13	Supporting search function by destinations, travelers...
R14	Supporting car rental services
R15	Supporting forum
R16	Recommending travel companions
R17	Recommending local tour guide
R18	Recommending popular travel blogs and destinations

Please rate below questions about yourself from 1 to 5, with 1 being most disagree and 5 being most agree.

Do you like travel?
Do you browse the travel website frequently?
Basically you use the website for booking the business trip?
Basically you use the website for booking the personal trip?
Do you have money but little time for traveling?
Do you have time but little money for traveling?
Do you plan a traveling more than twice a year?

Appendix 2: Rating Statistics From CFPS Primary Persona

	Favored User	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9
R1	4	4	4	5	4	4	3	4	5
R2	5	5	5	4	4	5	5	4	4
R3	5	5	4	5	5	4	5	4	4
R4	5	5	4	5	4	5	5	4	4
R5	5	5	5	5	5	4	4	4	4
R6	4	3	4	3	3	4	4	4	4
R7	4	3	4	3	3	4	4	4	5
R8	5	5	4	5	5	5	5	5	5
R9	5	4	4	5	5	4	4	5	4
R10	5	4	4	4	4	4	4	4	5
R11	4	3	4	3	3	3	4	4	4
R12	3	4	3	4	4	4	4	4	4
R13	5	4	5	4	4	5	4	4	5
R14	4	5	4	5	5	3	3	3	4
R15	5	5	4	5	3	4	3	3	5
R16	4	4	4	4	4	4	4	3	4
R17	4	4	4	4	4	4	4	4	4
R18	5	5	4	5	5	4	5	4	5
Tota Req.	81	77	74	78	74	74	74	71	79
G1	5	5	4	5	5	4	4	4	5
G2	5	5	4	5	5	4	4	4	5
G3	3	3	3	3	3	3	2	2	3
G4	5	4	4	5	5	4	4	4	5
G5	4	4	4	4	4	3	4	3	3
G6	3	3	3	3	2	2	3	3	2
G7	5	5	4	5	4	5	4	5	4
Total	111	106	100	108	102	99	99	96	106
Similarity Coefficient	-	0.7034	0.7024	0.6988	0.643	0.6346	0.5926	0.5751	0.5531

	User 10	User 11	User12	User 13	User 14	User 15	User16	User 17	User 18	User19	User 20
R1	5	5	4	3	5	5	3	4	5	4	3
R2	5	5	4	4	4	3	5	4	4	5	4
R3	5	4	4	5	4	3	5	4	4	4	4
R4	4	5	3	4	4	3	3	4	4	4	3
R5	4	5	4	4	4	4	5	4	4	5	3
R6	5	4	4	4	5	3	3	4	3	5	4
R7	5	5	4	4	5	5	4	4	3	5	4
R8	4	5	3	4	4	4	3	5	3	5	5
R9	4	4	5	4	4	3	3	4	3	4	5
R10	4	5	5	4	5	3	4	4	3	4	4
R11	3	2	4	3	4	3	3	3	3	2	4
R12	3	4	4	5	4	3	4	5	4	5	5
R13	4	5	5	5	4	5	4	5	4	5	4
R14	4	4	4	4	4	4	3	3	3	3	4
R15	4	2	5	3	3	3	3	3	3	3	4
R16	3	3	3	4	4	4	3	3	4	4	4
R17	3	5	5	4	4	4	3	4	4	5	4
R18	4	4	4	4	4	4	4	4	4	3	5
Tota Req.	73	76	74	72	75	66	65	71	65	75	73
G1	5	4	4	5	4	4	4	4	4	3	4
G2	5	5	4	5	4	5	3	4	4	5	4
G3	2	2	2	3	2	3	3	3	3	2	3
G4	3	4	5	5	4	5	3	4	3	4	4
G5	4	4	4	4	4	4	4	4	3	3	4
G6	2	2	3	2	2	2	3	3	2	2	3
G7	3	3	5	3	4	4	3	3	3	3	3
Total	97	100	101	99	99	93	88	96	87	97	98
Similarity Coefficient	0.5286	0.4692	0.4122	0.4033	0.3537	0.3089	0.2689	0.2587	0.2438	0.1899	0.1656