Investigating the Influence of Personal Values on Requirements for Health Care Information Systems

Rumyana Proynova, Barbara Paech
Institute for Computer Science
University of Heidelberg
Im Neuenheimer Feld 326
D-69120 Heidelberg
{proynova, paech}@informatik.uni-heidelberg.de

Sven H. Koch, Andreas Wicht, Thomas Wetter
Section for Medical Informatics
University of Heidelberg
Im Neuenheimer Feld 305
D-69120 Heidelberg
sven.koch@med.uni-heidelberg.de,
andreas.wicht@med.uni-heidelberg.de,
thomas.wetter@urz.uni-hd.de

ABSTRACT
Stakeholder requirements for health care information systems cannot be defined purely objectively. Instead they are influenced by personal and social factors. In this paper, we present preliminary insights into one such factor, namely personal values. Based on work from psychology, we have developed first instruments to elicit personal values and their relationships to software requirements by interviewing nurses and physicians. We report on these instruments and the results of applying them in two small case studies.

Categories and Subject Descriptors
D.2.1 Requirements/Specifications – Elicitation methods

General Terms
Human factors

Keywords
Personal values, requirements engineering

1. Introduction
This paper represents one of the early results of our research project. In this section, we describe the motivation, goals and benefits of the research project and indicate the part that is covered in this paper.

1.1 The motivation of our research
A software project is only successful if the requirements of all stakeholders are met [1]. While software in the health care sector is written for the use of organizations, usual stakeholder models like the onion model [2] show that many important stakeholders are in fact individuals, including the users of the system. As individuals, they are motivated not only by the organizational goals, but also by individual factors like emotions, beliefs, values, attitudes [3].

We consider the health care domain a prime example for a situation where individual factors matter. While economic conduct is increasingly expected from health care professionals, their foremost goal remains improving the patients' state of health by providing a cure, supporting treatments, or palliative care. Consequently, clinical outcomes in terms of effective treatment are emotionally appreciated, while economic outcomes are perceived only marginally. Moreover, health care professionals are highly educated individuals, who do not adhere to predetermined work scenarios, but are expected and trusted to make decisions on /decide on individual situations, often under time pressure. Practice shows that market leaders for business information systems who adapt their business software for use in clinical environments typically suffer from low user acceptance of their products [4]. Therefore, we expect that the effect of individual factors on user expectations will be especially pronounced/clear in health care.

1.2 The scientific goals of our research project
Our research focuses on personal values as one important factor which influences the goals of individuals.. Our project aims at:

a) developing a method to elicite values and related information such as attitudes, needed to define value-based requirements;

b) developing a systematic method to derive requirements from the elicited information;

c) creating a reference model, which maps the personal values commonly found with health care specialists to typical value-based requirements for health care systems.

This paper reflects the current status of our research. We have already developed the elicitation method and an early version of the derivation method. The reference model is, at the moment, an early stage prototype. In this paper we describe the values and attitudes elicitation method and the requirements derivation method, as well as the case studies which were used to verify the existing version of the methods and to provide insights we used to refine the methods.

1.3 The practical benefits of our research
Our research provides methods suitable for practical use during requirements engineering (RE) in software projects. The information eliciting method can be used on its own to provide

SEHC'11, May 22-23, 2011, Waikiki, Honolulu, HI, USA

information about values and attitudes towards tasks. If a requirements analyst does not wish to commit resources to a systematic value-based requirements derivation, it is still possible to use this information in the requirements specification phase. The value-related information can be used to create a deeper intuitive understanding of the users and their needs, and it can be used to enhance classic techniques that rely on information about the users and their needs, like personas and goal models.

The requirements derivation method can be used to systematically generate new requirements or enhance existing requirements. This method is not meant as an alternative to traditional requirement engineering techniques, but rather as an additional approach, which is used together with other techniques to create a better, more complete requirements specification.

The requirements discovered by a personal values-based method only take into account the users' personal preferences,, but not the organizational objectives. Thus, after the values-based requirements have been derived, the existing business constraints and stakeholder goals (elicited by a traditional RE method) have to be considered. If a value-based requirement is in conflict with these, the conflict should be resolved using standard techniques, such as described in [2]. A low-effort option would be to decide that value-based requirements always have a lower priority than requirements derived from business goals and to drop value-based requirements in the final specification if they are in conflict with business-goals-derived requirements.

The requirements created by the derivation method are intended to be value-specific, as in "Users with [value] would like the system to support [feature]". We are aware of the fact that it is economically impossible to create different systems for users exhibiting different personal values. Other research suggests however, that users of a specific system share common values: Eliason at al. [5] report a profession-specific distribution of values while Mumford et al. [6] show that employees of a given organization (in which a software system is in use) have similar values. This enables a requirements engineer to prioritize requirements so that requirements important for the values predominant in a given profession and organization get a higher priority. Requirements that are important to rarer values and thus to a small proportion of users, can then be implemented as configurable options or left out, depending on the resources available.

1.4 Structure of the paper

In the following section, we give some background information, including the definitions for personal values and attitudes as well as their influence on a person's decisions. It also contains an overview of related work.

In Section 3 we describe a personal values elicitation method together with an early version of a method for requirements derivation. This includes two interchangeable versions, one of them based on personal values and another one based on attitudes. In Section 4 we present the case studies we conducted and explain how we gathered the data and how we evaluated it.

Section 5 contains a discussion of our results. We conclude the paper with an outlook describing the future direction of our research.

2. Background and related work

In this section we give some background information on the psychological research we use in our method (Section 2.1) and discuss related work dealing with the users' personal qualities (Section 2.2).

2.1 Personal values and attitudes

Personal values are a common object of study in psychology. In our research we are using the personal values theory as developed by Shalom Schwartz [7]. He used wide reaching empirical data to determine the personal values exhibited by individuals from different cultures around the world and to create an instrument suitable for capturing these values.

Table 1 lists the values determined by Schwartz with short explanations. Schwartz found that values expressed by individuals are constant over time and are present in individuals of different races, nationalities and social or cultural backgrounds. Tasks individuals perform are normally aimed at achieving these values if no other behavioral determinants, such as biological needs or ideological demands, are predominant.

Table 1. The personal values as defined by Schwartz [7]

Value: Description
Achievement: Personal success through demonstrating competence according to social standards.
Benevolence: Preservation and enhancement of the welfare of people with whom one is in frequent personal contact.
Conformity: Restriction of actions, inclinations and impulses likely to accept or harm others and violate social norms or standards.
Hedonism: Pleasure and sensuous gratification to oneself.
Power: Social status and prestige, control and dominance over people and resources.
Security: Safety, harmony and stability of society, of relationship, and of self.
Self-direction: Independent thought and action-choosing, creating, exploring.
Stimulation: Excitement, novelty and challenge in life.
Tradition: Respect, commitment and acceptance of the customs and ideas that traditional culture or religion provide the self.
Universalism: Understanding, appreciation, tolerance and protection for the welfare of all people and for nature.

Although Schwartz defines values as "desirable, trans-situational goals" [7], this connotation of "goal" is different from the one typically used in RE literature. While goals in RE are limited to a single purpose, Schwartz' personal values function on a much higher level and include aspects such as social recognition and freedom of choice.

Values influence behavior not directly, but through attitudes. An attitude is the sentiment "I like [object]" or "I do not like [object]", where the object is virtually anything which fits in this form of expression. When making decisions, people tend to choose options they like and the like/dislike attitude towards the options is based on values among other factors [8].

Schwartz also made the important observation that all people have all ten values – a person who values benevolence very high might also think that the goals associated with achievement, like a high social status, are desirable. But given a situation where he or she must decide between an alternative promoting achievement and another alternative promoting benevolence, he or she is likely to prefer the alternative promoting his or her higher-ranked value, in this example benevolence.

2.2 Related work

Studies of user acceptance of software systems often find that approval depends strongly, among other factors, on the users' personality [9]. However, a study of existing RE literature did not reveal systematic advice as to how to integrate knowledge about user personality into the software engineering process.

Among the more prominent studies in this area are the publications of Isabel Ramos et al. [10], [11]. They focus mainly on emotions, but consider values and beliefs of individual users to be also an important basis for requirements elicitation. They describe a constructionist RE process, where the knowledge about personal issues is elicited, managed and used alongside with other, more traditional parts of RE knowledge. They also propose that the individual stakeholder representatives should be selected ensuring an accurate representation of all work aspects of the organization, including aspects which depend on the personality of the individuals.

Ramos et al. provide a valuable framework which shows the role of knowledge of the users' personality within the requirements engineering process. However, it does not attempt to propose specific methods for eliciting, documenting and using this kind of knowledge.

Sarah Thew et al. [12] also research the role of "soft issues" in the requirements process. They give a list of possible soft issues and their impact on the elicitation process itself, should the stakeholders exhibit these soft issues. The emphasis of their publications is on the need to shape the communication with stakeholders according to their values, emotions and beliefs. They do not directly address the possibility of requirements being different between stakeholders with different values.

In the field of Human Computer Interactions there is also the concept of value sensitive design (VSD). This concept does not use the psychological definition of personal values, instead it sees human values as an element of moral and it promotes systems design which encourages responsible computer use [13].

The term value-based software engineering has also been used before in a different connotation [14]. In this context, "value" does not denote the personal values of the users, but mostly the monetary value of the software product.

In an earlier publication [15] we proposed a link between personality, especially personal values, and software requirements. This was a vision paper, based on literature studies. The present article builds on this first formulation of our theory and uses empirical data from two case studies to uncover links between specific attitudes, values and requirements.

3. Deriving software requirements from personal values and attitudes

In order to create a method for requirements elicitation, we started with a literature study to determine the information we will need to gather from the users. Then we conducted two case studies, which served a double purpose. First, they were used to validate the information gathering step of our method. Second, the information collected was used to substantiate the method itself.

The method we propose is intended to be based on information on personal values. However, some users regard such information as sensitive and may refuse to participate on privacy grounds. As attitudes are strongly influenced by values, the method can be based on attitudes instead. We describe both variants in the following two sections.

Figure 1 displays the information which is used in our method. Each rectangle represents an information type. The arrows between the rectangles represent the relationship between the information types. The comments indicate which information type is discovered in which step of our method.

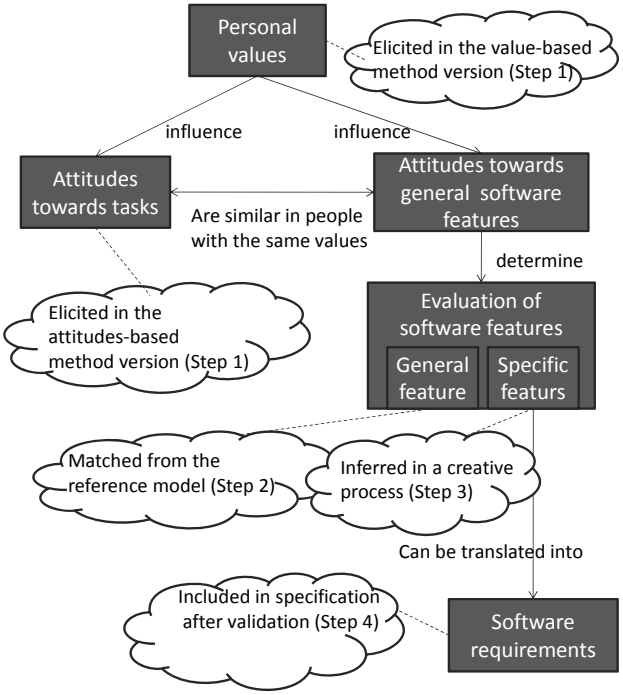

Figure 1. The information used for the value-based method

3.1 The personal values-based method

In this method version we rely on the assumed causative relationship between personal values and the users' evaluation of software features. We gather the information on personal values and then, in a reference model, look up the general features that a user with values likes. We then derive project-specific features from the proposed general ones. From this we create project-specific requirements. After a validation step based on feedback from the appropriate stakeholders, we include these requirements in the requirements specification.

3.1.1 Step 1: Collecting information

This step consists of administering a standard questionnaire called the Schwartz Value Survey [7]. This survey is easily conducted with a large number of users, as the time effort for the individual user is limited to about 15 minutes and the data can be processed automatically. It consists of 57 items, for which the study subject is asked to indicate how much they conform to his or her views using a 9-point Likert scale.

To calculate a value ranking for an individual, we average the score on the Likert scale for the items correlated with each of the ten personal values. We then rank these average scores.

As the higher ranked values are the ones most likely to influence a decision, we do not use all ten values in the ranking (in this case, all users would be the same), but only the two highest ranked values of each user. For the remainder of this paper, we do not write "This user ranks [value] at the first or second position in the ranked value list", but instead use the shorter formulation "This user has [value].

3.1.2 Step 2: Matching against a reference model

In this step a reference model is needed. This is a domain specific model which maps personal values to possible software features, ranked by the users' evaluation of these features. The reference model contains information of the type "Users with [value] think that [feature] is [type]", where [type] is an evaluation of the feature describing its desirability. So the requirements engineer can use this model to select features according to the values elicited in the first step.

A reference model as described here needs to be based on extensive empirical data. We are currently gathering this data and will then publish the model itself. An early prototype, described in Section 4.4, was created in the case studies.

3.1.3 Step 3: Deriving requirements

The features identified in Step 2 need to be converted to requirements. The features included in the reference model are too abstract, as they need to apply to a wide range of systems. Thus, the requirements engineer has to infer a requirement suited to the specific project. If, for example, in Step 1 a prevalence of the value *self direction* was discovered among the users, and the reference model states that "Self direction users think that additional information about treatments is a must-be feature", the requirements engineer (alone or with the help of domain experts) has to find out where in the system additional information about treatments can be included, and has to formulate a requirement like "On the prescription screen, there should be a link to external information sources about the prescribed treatment". This does not mean that the requirements engineer is supposed to create new requirements from scratch; as our method is used as an addition to traditional methods, the requirements engineer can use the information from the reference model in conjunction with information gathered with other RE techniques to arrive at new requirements.

3.1.4 Step 4: Validating requirements

The requirements derived in Step 3 are a combination of general domain information reflected in the reference model and a creative effort of the requirements engineer. As they were not based on information from stakeholders other than the personal values of users, they should only be included in the requirements specification after they have been approved by the relevant stakeholders. They should be validated like any other requirement.

3.2 Alternative version of Step 1 and 2: The attitude-based method

In our research we sometimes met users wary to provide information about their personal values. While this is not a problem if only a few users hold this sentiment, this might also be an issue for a union of work force advocates and thus hinder the gathering of personal values information for RE purposes. In such situations our method can be used with information about attitudes towards work tasks instead.

This is made possible by the fact that the users' evaluation of features is closely connected to their attitudes to a feature. Theory predicts that personal values (and other factors) will have similar influence on attitudes to different objects, so the users' attitudes towards software features should exhibit a correlation to their attitudes towards the tasks supported by the software [8].

The value-based method should still be preferred, because it requires less time and effort. Also, we expect it to provide better results, as the connection between personal values and feature evaluation is less likely to be skewed by additional factors than the task attitudes-feature connection.

In the following we only describe the first two steps of the method, as Steps 3 and 4 are equivalent to the corresponding steps in the value-based version.

3.2.1 Step 1: Collecting information

The goal in this step is to gather information about the users' attitudes towards their tasks. This is done by a questionnaire consisting of a list of the tasks which the system will support. An example for a task would be "Document a patient's illness history". The tasks can be identified using standard RE techniques, as described in textbooks [2], [16], [17]. Participants are then asked to indicate for each task how much they like doing it, using a 9-point Likert scale ranging from "I dislike performing this task" to "This task is among my favorites". There is a separate item on the scale marked as X, which is labeled as "This task does not apply to my work". The questionnaire data analysis consists of creating three categories of tasks – the tasks the user likes, the tasks he or she dislikes, and the neutral ones for each participant. To evaluate the study, we just divided the scale in three equal intervals.

3.2.2 Step 2a): Assigning tasks to categories

A reference model is needed again for this step, this time mapping attitudes towards task categories to features. Given the high number of possible tasks, creating a reference model containing correlations for all possible tasks is not practical. Instead, we include categories of tasks in the reference model, e. g. "tasks involving direct communication with the patient". So before the requirements engineer can start matching the users' attitudes towards tasks to the reference model, he or she needs to assign each task to one or more categories included in the reference model. For each category, the prevalent attitude is determined by the attitude towards the majority of tasks in that group. If there is no clear prevalence of one attitude within a category, the matching in step 2b) should be done for both the "like" and "dislike" users for this category.

3.2.3 Step 2b): Matching against a reference model

This step is similar to Step 2 of the value-based version. The requirements engineer uses an attitude-based reference model provided by our research which contains information like "Users who [like/dislike] [task category] think that [feature] is [type]", where [type] is an evaluation category for software features. Depending on the attitudes exhibited by the users in his or her project, he or she then chooses the features likely to be preferred by the users. Then he or she continues with Steps 3 and 4 as described in Section 3.1.

The task questionnaire is not a validated instrument for eliciting personal values. With this method, there is no guarantee that the attitudes towards the tasks in our list cover all ten values. However, personal values and attitudes are only a small subset of the reasons which compel a user to want a given feature. Thus, neither approach allows us to build an exhaustive catalogue of liked and disliked features. While the attitudes-based method will probably not uncover some of the feature evaluations which would have been discoverable with the value-based method, this is only a drawback that has to be weighed against its advantages and does not invalidate the use of attitudes per se.

4. Case studies

We developed first versions of the methods described above based on literature. To validate the methods' feasibility and gauge their performance and the usefulness of the resulting requirements, we

applied the methods in two small case studies, one with physicians and one with nurses.

In these studies we adhered to the methods as far as possible. We could not conduct the matching against the reference model, because we did not have one. Instead, we used the data gathered in the first steps, together with information on the users' Kano evaluation of software features, which we collected specifically for the case study, to create an early prototype of the reference model.

In the next section we give some background information on the participants in our study. After that, we describe the case study conduction step by step, together with any deviations from the method described in the previous section and list the data gathered in this step.

4.1 Participants of the study

The physician case study included three physicians from the University Hospital in Heidelberg. They varied in seniority from resident to a member of clinic management. The application on which we focused was a drug database for managing medication prescriptions. The application was developed by a third party as an off-the-shelf solution for clinics.

The nurse case study was conducted in a clinic in the University of Freiburg. The interviews were conducted with six nurses from two wards, and four of them completed the questionnaire, resulting in a 67% response rate. The responding nurses ranged in seniority from registered nurse to nurse manager. We focused on their use of an application for patient management and documentation, developed in-house.

4.2 Step 1: Information collection

In practice, we expect a requirements engineer to use either the value-based or the attitudes-based version of our method. We needed to assess both collection instruments, so we collected data on both personal values and attitudes.

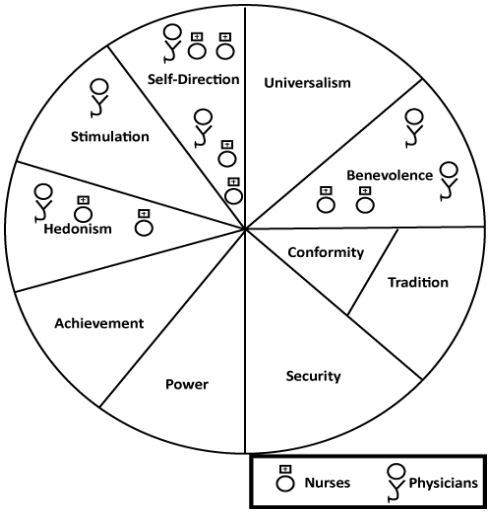

Figure 2: Values exhibited by the case study participants

Figure 2 visualizes the values and the distribution of the two highest values of the participants. Our small sample does not cover all values, but we have enough difference between participants to ensure realistic results from the application of the method. The findings are consistent with those of a study with several hundred respondents, which found that benevolence and self-direction are the most common values among physicians [5].

For a task list, we studied literature from the medical domain and performed preliminary observations at the clinics where the studies were conducted, arriving at 43 tasks for physicians and 45 tasks for nurses.

Figure 3 is a histogram of the relative frequency of attitudes towards tasks. For example, a value of 23% for physicians at point 7 means that the average physician participant rated 23% of the tasks as a 7 on the Likert scale.

The skewing to the right is typical for self-administered preference questionnaires [18]. However, even with the small participant number we can discern that the distribution is trimodal (has three local maxima). This means that the method is well suited for capturing the difference between tasks the user likes, tasks the user dislikes and tasks to which the user is indifferent.

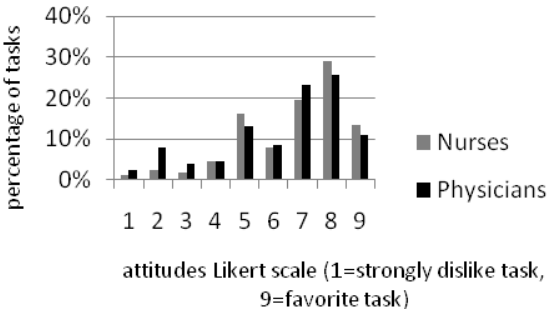

Figure 3 Distribution of the case study participants' attitudes towards tasks

4.3 Step 2a) Task categorization

We did not yet have defined task categories. Instead, for each profession (physicians and nurses) we made a collection of task category proposals, with each author contributing a list of proposals based on his or her knowledge of the medical domain and of information systems. This was performed separately for the physicians' and nurses' tasks. We then assigned each task to one or more categories.

In order to determine which of our category proposals are in fact connected to attitudes, we performed a binomial test on the attitudes data. For each study participant, we tested for the zero hypothesis that he or she likes (or, in a second test, dislikes) no more tasks from the proposed category as tasks not assigned to the proposed category, with hypothesis rejection. If we could reject the hypothesis, we accepted the proposal as a category of tasks towards which users have a clear-cut attitude.

To create a reference model prototype, we accepted a category proposal even if its existence was confirmed by the data of a single study participant. For the final version of the reference model, category proposals will only be included if they pass the binomial test for a sufficient number of users.

For example, the task category "Documentation" consisted of the four tasks "Create an epicrisis", "Enter diagnostic codes", "Enter patient's illness history into the system" and "Do special documentation, e.g. for a research study". Physician C disliked three of these four tasks. Out of the complete set of 43 tasks, he only disliked five. So we concluded that Physician C has a specifically strong dislike for documentation related tasks. The binomial test confirmed that finding at a 5% significance level. This result confirmed that "Documentation" is, indeed, a category of tasks towards which users have a consistent attitude, and as

such, it is included in the list of categories used to create the reference model prototype.

4.4 Step 2b) Creating a reference model prototype

We used these studies to construct the first prototype of the reference model. Therefore, matching against a reference model was not possible. Instead, we directly elicited feature evaluations from users and correlated these once to the personal values and once to the attitudes towards tasks.

We created a list of features present in the current software systems of our study participants. We used information from direct user observation, manufacturer provided demonstration materials, and also the use of a test account for the physicians' system and a personal conversation with a leading developer of the nurses' system. We then presented the study participants with the list of features (14 features for the physicians and 8 for the nurses) and asked them to evaluate each feature using the Kano model [19]. For this model, the user has to imagine the product in two versions, both with and without the feature in question, and then indicate how much he or she likes this product version on a scale of five predefined answers. Based on the answers for both versions, a feature is evaluated either *indifferent*, *must-be*, *performance*, *excitement* or *reverse* (reverse meaning that the customer does not want the feature to be included in the product).

Table 2 lists the evaluation the three physicians gave to the software features of the medication database. It refers to each Kano type using its first letter, so a line containing the description of a feature followed by the letters p, i and m means that the feature was evaluated as performance by physician A, indifferent by physician B and must-be by physician C. The letters "r" and "e" indicate reverse and excitement features, respectively.

Table 2 Kano evaluation of software features from the physician study

Features ↓ Physicians →	A	B	C
Displaying current prescription	m	m	m
Displaying potential interactions	p	p	p
Medication search	m	m	m
Displaying detailed information about a drug	p	m	m
Adding a medication to the patient's prescription	m	m	m
Loading patient data from administrative system	m	m	m
Dosage choice, including the creation of an individual dosage schedule	m	m	m
Possibility to document the rationale behind the prescription	p	p	r
Possibility to add arbitrary text to the prescription document, e.g. notes	p	i	m
Print a prescription	m	p	m
Integration of information from external systems, e.g. a catalogue of diagnostic codes	i	i	i
Possibility to open official guidelines from within the system	i	r	i
Additional information about dosage adjustment for patients with kidney/liver malfunction	p	p	p
Access to a clinic-specific knowledge base for medication administering considerations	i	i	i

For most features, there is no difference between the participants; they are always assigned to the same Kano type. But there are still some features which participants with different values assign to different categories. We used this subset of value-differing features to compile prototypes of the reference model.

To create a reference model prototype for values, we correlated the personal values to each feature evaluation category. For the prototype for a reference model for attitudes, we correlated the attitudes towards the task categories to the software features. These correlations were present only for a part of the data. For example, the feature "print a prescription" is described as "performance" by physician B and "must-be" by physicians A and C. Physician B was the only participant exhibiting the value hedonism at the first or second position in the value ranking, so the fact that hedonism users evaluate printing a prescription as a performance feature was added to the reference model prototype. As physicians A and C share the value self direction, the correlation between self direction and evaluating printing as a must-be feature is also entered in the reference model.

Table 3 and

Table 4 contain all the correlations we found in our case studies. An example for reading the first row of the physicians table would be "A physician who has the value benevolence thinks that displaying detailed information about a drug is a must-be feature". The abbreviations used to indicate Kano types are the same as in the previous table.

Table 3 Reference model prototype for physicians

Physicians – personal value correlated to features	
Value	Feature (The system supports…)
Benevolence	Displaying detailed information about a drug (m)
Stimulation	Displaying detailed information about a drug (p), Possibility to add arbitrary text to the prescription document (p)
Hedonism	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)
Self direction	Print a prescription (m), Possibility to open official guidelines from within the system (i)
Physicians – attitudes towards tasks correlated to features	
Task group – attitude	Feature (The system supports…)
Information search – liked	Displaying detailed information about a drug (p), Possibility to add arbitrary text to the prescription document (p)
Communication with coworkers – liked	Displaying detailed information about a drug (p), Possibility to add arbitrary text to the prescription document (p)
Patient contact – liked	Displaying detailed information about a drug (m)
Documentation – disliked	Displaying detailed information about a drug (m)
Administration – disliked	Possibility to document the rationale behind the prescription (r), Possibility to add arbitrary text to the prescription document (m)
Treating a patient – disliked	Possibility to document the rationale behind the prescription (r), Possibility to add arbitrary text to

	the prescription document (m)
Treating a patient – liked	Possibility to add arbitrary text to the prescription document (i)
Involving one's hands - liked	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)
Delegation – liked	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)
Organize patient treatment – liked	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)
Documenting patient state – disliked	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)
Exchange information – disliked	Possibility to add arbitrary text to the prescription document (i), Print a prescription (p), Possibility to open official guidelines from within the system (r)

Table 4 Reference model prototype for nurses

Nurses – personal value correlated to features	
Value	Feature (The system supports…)
Benevolence	Printing sticky labels with patient ID (m), Sending a request to the administrative department (m), Showing available forms (p), Entering nursing care level (m), Managing patient release preparation (m)
Hedonism	Printing sticky labels with patient ID (p), Sending a request to the administrative department (i), Showing available forms(p)
Self direction	
Nurses – attitudes towards tasks correlated to features	
Task group - attitude	Feature (The system supports…)
Providing information to patient – liked	Listing all documents relating to a patient (i), Entering nursing care level (i), Managing patient release preparation (p)
Logistics – disliked	Listing all documents relating to a patient (i), Entering nursing care level (i), Managing patient release preparation (p)
Working with other nurses – liked	Listing all documents relating to a patient (r)

None of these results have a statistical significance due to the small sample size. They are included here for illustrative purposes only.

As the studies were not a part of a software project, we could not conduct steps 3 and 4.

5. Discussion

The two case studies showed that our elicitation instruments work well and that relating future attitudes to values or task attitudes respectively is possible. In the following we comment on the results and ideas for improvement.

Our method starts with the gathering of data on either personal values or task attitudes of users. From a theoretical point of view, the use of personal values is the superior solution. However, if users feel that providing their personal values in a RE process infringes on their privacy, the use of task attitudes is an acceptable substitute. Both kinds of information can be gathered accurately and with low effort using standard psychological instruments.

The attitude-based method version needs a description of the user tasks the software will support. While the identification and correct representation of tasks is a costly, time-consuming activity, it is a standard part of the RE process, so it does not require any additional resources.

We noticed a curious phenomenon in the distribution of attitudes towards tasks as gathered in the case studies. While the cumulated distribution over all participants is clearly trimodal, one participant exhibited a monotonously increasing distribution without maxima, meaning that he liked almost all tasks. If such a distribution should appear often in the wider study, we intend to revise the limits we use to classify attitudes towards tasks.

Assigning the tasks to predefined categories is needed for the attitudes version of our method. This step is not a usual part of the RE process. It is especially error-prone, because it relies on the requirement engineers' subjective assessment of the matches between tasks and categories. We intend to keep the risk of mis-categorization minimal by 1) creating categories agreed on by multiple experts and 2) providing unambiguous category descriptions.

The matching of the personal values or the task group attitudes to the features is easily done, but its quality depends on the quality of the reference model available. We intend to provide a reference model based entirely on empirical studies with actual software users, with enough users to ensure statistical significance of the matches included in the model.

The usefulness of the end results of our method is determined by the type of information it produces. In the current version of the method, the results of the method are of the type "Users with [value] think that [software feature] is of [Kano type]". The Kano scale is a popular method to evaluate requirements for software features and it can be used as a basis to decide which possible features to include in a software product. For example, if the reference model predicts that "phonetic search" is perceived as a "must-be" feature by Benevolence users and the questionnaire yields a prevalence of Benevolence users in the customer organization, the requirements engineer should propose to the stakeholders to include "phonetic search" into the requirements specification, because users usually do not accept products which do not meet the "must-be" features. However, we noticed some problems when applying the Kano questions in our study, including confusion over their meaning and a tendency to answer "must-be" for every feature, so a change of the evaluation scale might lead to better results.

Another issue with the usefulness of the method results is the granularity of the features included in the reference model. The feature of a reference model only makes sense in a specific context – an information of the type "Users with [value] think that an alert shown when the patient's vital parameters exceed predefined limits is a must-be feature" is only useful when writing requirements for a medical information system. But even within a domain such as medicine, different reference models for different system types are conceivable. Our goal is to provide a reference model which is broad enough to work with a wide range of systems within a domain. On the other hand, making the features too general reduces the usefulness of the information gained. In the case studies we used a granularity roughly corresponding to single use case steps, like "should the system support placing of

medications on the prescription?". This lead to results that did not discriminate well between values. For the future, we intend to change the granularity asking not about which use case step should be supported, but how it should be supported, e. g. "When you are creating a prescription, should the system offer a search for alternatives to a medication already on the prescription?" while still maintaining a detail level well above HCI design decisions.

Another consideration to improve the quality of our results would be to use new features in the questionnaire. The questionnaire used in the case study asked the users to evaluate features they were already familiar with. However, this can skew the results due to the familiarity bias (individuals who use a tool often become attached to it and value it higher than comparable tools [20]). Our future questionnaire will ask the users to imagine a new system being built and will include features not present in their current systems.

6. Conclusion and outlook

Our research shows that it is possible to elicit value-related information and correlate it to software features. We have developed an early version of an elicitation method geared towards easy integration with established RE techniques. Specifically, we aimed at the reuse of information gathered in classic RE, like tasks, and we also provided a second, attitude-based version of our method, which can be applied if users object to reveale their values. Based on the experience gathered in empirical studies, we are currently working on a finalized version of the method which is better suited for use in real life projects.

The derivation method in its current form is dependent on the existence of a domain specific reference model mapping feature types to personal values. We are currently conducting a large empirical study which will allow us to create this model for the medical domain. In parallel, we are working on a variant of the derivation method, which will not require a reference model. A requirements engineer will need some prior knowledge about values in order to use this version, but it will be usable in domains for which a reference model does not exist.

We are also refining our elicitation method to provide higher quality features by addressing the issues of granularity and evaluation scale we discussed in the previous section. This will allow for the discovery of "hidden" requirements and for their classification according to both the usefulness for the business process the software will support and their appeal to the users, resulting in requirements for software the users will be glad to use.

7. Acknowledgements

The authors thank all participants of the study for their contribution to our research project. The research presented here was made possible through a grant by the German research association DFG.

8. References

[1] Sommerville, I., *Software Engineering 8*, 8 ed.: Addison Wesley Publishing Company, 2007.

[2] Alexander, I. and Beus-Dukic, L., *Discovering Requirements*, 1 ed. Chichester: Wiley, 2009.

[3] Tesser, A. and Schwarz, N., Eds., *Intraindividual process* (Blackwell handbook of social psychology 1). Malden: Blackwell publishers, 2001.

[4] Wetter, T. and Paech, B., "What if „business process" is the wrong metaphor? Exploring the potential of Value Based Requirements Engineering for clinical software," in *Accepted at MedInfo 2010*, CapeTown 2010.

[5] Eliason, B. C., Guse, C., and Gottlieb, M. S., "Personal values of family physicians, practice satisfaction, and service to the undeserved," *Archinces of family medicine,* vol. 9, 2000.

[6] Mumford, M. D., Connelly, M. S., Helton, W. B., Van Doorn, J. R., and Osburn, H. K., "Alternative approaches for measuring values: direct and indirect assessments in performance prediction," *Journal of Vocational Behavior,* vol. 61, 2002.

[7] Schwartz, S., Melech, G., and Lehmann, A., "Extending the Cross-Cultural Validity of the Theory of Basic Human Values with a Different Method of Measurement," *Journal of Cross-Cultural Psychology,* 2001.

[8] Ajzen, I. and Fishbein, M., "The influence of attitudes on behavior," in *The handbook of attitudes*. vol. 173, Albarracin, D., Johnson, B., Zanna, M., Ed., ed Mahwah, NJ: Lawrence Erlbaum, 2005.

[9] Thong, J. Y. L., Hong, W., and Tam, K., "What leads to user acceptance of digital libraries?," *Commun. ACM,* vol. 47, 2004.

[10] Ramos, I. and Berry, D., "Is emotion relevant to requirements engineering?," *Requirements Engineering,* vol. 10, 2005.

[11] Ramos, I., Berry, D. M., and Carvalho, J., "Requirements engineering for organizational transformation," *Information and Software Technology,* vol. 47, 2005.

[12] Thew, S. and Sutcliffe, A., "Investigating the Role of 'Soft Issues' in the RE Process," in *Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*, 2008.

[13] Friedman, B. and Kahn, P., "Human agency and responsible computing: Implications for computer system design," *Journal of Systems & Software,* vol. 17, 1992.

[14] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., and Grunbacher, P., *Value-based software engineering*. New York: Springer, 2006.

[15] Proynova, R. and Paech, B., "Use of Personal Values in Requirements Engineering- A Research Preview," presented at the 16th International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen (Germany), 2010.

[16] Lauesen, S., *User Interface Design - A software engineering perspective*. Harlow: Pearson education limited, 2005.

[17] Paech, B. and Kohler, K., Eds., *Task-driven Requirements in object-oriented Development* (Perspectives on software requirements. Dordrecht: Kluwer Academic Publishers, 2003.

[18] Peterson, R. A. and Wilson, W. R., "Measuring customer satisfaction: fact and artifact," *Journal of the academy of marketing science,* vol. 20, 1992.

[19] Sauerwein, E., Bailom, F., Matzler, K., and Hinterhuber, H. H., "The Kano model: How to delight your customers," in *International Working Seminar on Production Economics*, Innsbruck, 1996.

[20] Arieli, D., *Predictably irrational: The hidden forces that shape our decisions*: Harper New York, 2008.