# Enhancing user-centredness in agile teams: A study on programmer's values for a better understanding on how to position usability methods in XP

Michael Leitner[1]; Peter Wolkerstorfer[1], Arjan Geven[1], Manfred Tscheligi[1,2]

*(1) CURE – Center for Usability Research & Engineering*
*Modecenterstraße 17 / Objekt 2*
*1110 Vienna, Austria*
*{leitner | wolkerstorfer | geven | tscheligi}@cure.at*

*(2) ICT&S Center – University of Salzburg*
*Sigmund-Haffner-Gasse 18*
*5020 Salzburg, Austria.*
*manfred.tscheligi@sbg.ac.at*

Keywords: XP Programmers, programmer' high-level goals, user centred methods in agile teams

## Abstract

We present a study on programmer's high-level goals in eXtreme Programming settings (XP). We talked to 10 programmers using so-called "laddering interviews". The result presented is a "Hierarchical Value Map (HVM)" indicating agile programmer's high-level goals. This study was done to better orchestrate usability methods and integrate them into agile development processes. The study's results were used to position and adapt usability methods in a way that they are better aligned to the programmer's goals and therefore are more likely to be accepted. We draw conclusions on the basis of the study's results and experiences using agile usability methods in practise.

## 1.Introduction

User-centeredness describes the fact that technology design and development are aligned and targeted at the end-user needs, requirements and limitations. To achieve this, different methods are applied to secure the usability and user experience of hard- and software. In classical development processes, these methods (e.g. user requirement analysis, usability reviews, etc.) are included in the development process iteratively and at certain stages of the process. There is plenty of know-how in reference to the use of usability methods in classical soft and hardware development processes. In contrast, for agile settings different authors have discussed the application of usability methods in agile settings, but still little explicit know-how is available of how to adapt and position these methods in agile development settings. With this work we define a further step towards the adaption and use of usability methods in agile teams by proposing essential enhancement of these methods.

In order to identify these necessary adaptations of usability methods and to define a strategy to better position these methods in the course of the agile work processes we need to consider that usability methods - in most of the cases - are not the prime focus of XP programmers, not under their constant attention nor necessarily fit for application in XP processes. Therefore, to achieve a higher user-centeredness and an enhanced usage and acceptance of usability methods in agile teams the following two pillars need to be fostered:

a) Adapt and position usability methods in a way that they fit the agile team structure and process without disturbing the primary task: software development (=adoption towards the organizational and process goals).

b) Align and position the usability methods towards the programmer's goals and values[1] in order to achieve acceptance and use of these methods beyond indoctrination (=adoption towards psychological and team member's goals).

[1] In this paper the term "(high-level) goals" equal the term "values". To define and describe "values/high-level goals" we refer to Rokeach's Value List, which classifies terminal and instrumental values. Terminal values describe desirable end-states of existence (= goals that a person would like to achieve during his or her lifetime). Instrumental values describe preferable modes of behaviour to achieve a terminal value (Rokeach, M.).

With the study at hand we especially want to aim at pillar b) (programmer's goals and values) trying to align and adapt usability methods on the basis of programmer's goals and valued circumstances of agile development. Hence, the objectives of this study are:

1. *Identifying programmer's high-level goals:* We conducted a study on programmer's goals (and values) in agile team settings to enhance the understanding of how to position and adapt usability methods respectively. For this study we chose a semi-structured interview technique called "laddering" that is used to identify people's high-level goals in reference to a given object (in our case: the agile team and development setting) and the interview data is presented using a so-called Hierarchical Value Map (HVM). We discuss these findings in reference to the current know-how in the area of "perception of agile teams" and highlight the relevant findings for our study.
2. *Adapt and position usability methods accordingly:* On that basis we discuss how usability methods need to be adapted and positioned in agile settings in order to be better accepted and aligned to the programmer's values. We are discussing the value study's results and use these as a basis for a deeper look into the need of methodological adaption. We propose a set of particular improvements of selected usability methods for their use in agile settings. Finishing up we discuss some practical experiences that underline our argumentation.

Overall our study extends the State of the Art in two areas. On the one hand we are able to show new findings and aspects in the perception of agile methods by agile team members (programmers) and on the other hand we extend the know-how in the area of usability engineering in and for agile teams.

In general this study was conducted in a project dedicated to orchestrating usability and agile methods. In this project an agile developer team is working on a certain piece of technology. The usability methods that are used in this project are "personas", "usability evaluation and ad-hoc usability (expert) evaluation/input" and to some extend as well "automated usability evaluation tools[2]". Hence, in this paper we focus on improving these specific methods.

Further, it is necessary to mention, that in the team setting of this project no usability experts are part of the team, in contrast all team members are dedicated to programming. Therefore all usability input is brought to the team from as external source (The external source in this case are the authors ["we"] of this paper, which are supporting the agile team with usability input and methodological know-how. Hence, all practical experiences reported in this paper describe the author's observations as "external members" of the agile team).

The exact setting of the project and methodological approaches are described in (Wolkerstorfer, 2008). We are currently using the above mentioned usability methods in our project together with the agile development team, so far mainly without the suggested improvements in this paper. Our aim is to adapt the methods accordingly (on the basis of the improvements in this paper) and evaluate these in the field and together with the agile team we are working with.

We start with a discussion on current state of the art in related fields, discuss the study setting on programmer's goals and finish up with a discussion on enhancement of usability methods for a fruitful integration in agile development settings.

## 2. State of the art

In this section we summarize related work that is relevant for the study. This is on one hand research in the area of "perception of agile development and agile team settings" and on the other hand research and studies towards the use of usability and user-centred methods in agile team settings.

[2] In this context an "automated usability evaluation tool" describes software with the ability to assess the usability of a graphical user interface (GUI) automatically - for instance through the means of image analysis. The tool's feedback might be an index describing the GUI's visual complexity from 1 (bad) to 100 (good).

### Perception of agile development settings

Different authors have studied the perception of agile development methods in the last years. Several positive effects and perceptions were reported in reference to this organisational structure and work setting.

Dyba and Dingsøyr (2008) conducted a comprehensive state of the art analysis of research in the area of XP Programming. Citing Mannaro et al. they state that 90 % of XP Programmers would like to continue in the company they work, whereas only 40 % of Programmers in non-XP companies would like to do so. According to Mannaro's study XP Programmer's job satisfaction is higher (in comparison to non-XP Programmers). Further, XP programmers claim that their productivity is higher in XP settings. Citing Mann and Maurer Dyba and Dingsøyr state that XP Programmers think that the XP processes allows them to work more pinpointed towards the customer's goals.

Tessem and Maurer (2007) conducted semi-structured interviews and concluded that agile programmers feel "autonomy" (in daily work), "variety" (of type of work), "significance" (of each agile programmer in the team), "feedback" (related to own work done), "ability to complete whole tasks" (be responsible for the whole thing) and "motivation and job satisfaction". Likewise Law and Cheron reported the important factors "motivation" and "autonomy" in XP teams (2005). Whitworth (2007) conducted interviews with agile teams and reports similar findings (beside others). According to this work programmers reported that they "appreciate the knowledge share" and that the agile environment "amplifies the effectiveness of team meetings" and furthermore that these are a "forum for motivation". The agile setting would "enhance the team awareness", "foster the team-awareness", "the awareness of contribution of individuals". According to Withworth's study agile team setting supports team members taking a more active role and overall a positive self-image and self-esteem is emphasised/leveraged.

So far different studies and papers discussed programmer's perception on XP team settings, however, with our study we introduce a more "personal" view on these settings. Similar work was done by Withworth who introduced a more psychological view on XP team settings. With our study we enhance this and other studies by the following points:

a) Research the personal high-level goals (values) of programmers in XP work settings. So far no studies were explicitly dedicated to this factor in agile teams.

b) By the use of the means-end theory we are able to display the results gathered in a visual way (in a so-called "Hierarchical Value Map"). This visualization technique helps to understand the correlation between developer's high-level goals (values) and the attributes of the XP work setting.

### Usability methods and user-centredness in agile teams

When Kent Beck introduced XP (extreme programming) in 1999 (Beck, 1999) UCD (user centred design processes) did not play a remarkable role in his considerations. These considerations followed later, as agile development was adopted by companies with greater frequency every year which lead to a special interest group on agile user experience (Miller & Sy, 2009). Some experts doubt that the XP process leads to true user-centred design (Hudson, 2005). The issues arising from this problem statement suggest that XP and UCD won't fit. But this perception is simplistic and misguided which has already been shown in practice where success is reported (McInerney & Maurer, 2005). We can see succeeding practitioners combining UCD (user centric development) and XP/agile methods by varying approaches (Constantine & Lockwood, 2003; Ferreira, Noble & Biddle, 2007; Göransson, Gulliksen & Boivie, 2003; Holzinger, 2005; Holzinger, Errath, Searle, Thurnher & Slany, 2005; Norman, 2006; Wolkerstorfer et al, 2008).

Various studies have been directed to usability in agile settings. However, so far no particular improvement and necessary adaptations of methods have been discussed. Holzinger (2005) claimed that it is very important to know which usability method to select. Ferreira et al (Ferreira et al, 2007) give insight into the changes of the XP process to better integrate UCD work. They argue that 1. iteration planning affects UI design, 2. the development iteration drive the UCD activities, 3. usability testing results in changes in development and 4. that agile development changes the relationship between software developers and HCI engineers. Finally Wolkerstorfer (2008) discussed the idea of adopted personas (adopted to better fit XP), expert evaluation (the adoption to XP was that the expert feedback given was presented in XP-story style cards), and automated usability evaluations included in the unit-test system as a method-mix to provide flexibility for this selection.

Overall, the current state of the art especially shows a lack of adequate user-centred methods for agile settings. Although work has been directed to usability methodology in agile settings so far no particular improvement and necessary adaptations of particular methods have been presented. With this work we extend the current state of the art by proposing particular improvements of usability and user-centred methods in order to be more usable in agile teams and further more accepted by programmers. We try to close this gap by postulating necessary adaptations to certain usability methods.

In the next section we present the study setup of the conducted study on programmers high-level goals (values) and discuss the results presented in as a Hierarchical Value Map (HVM). On the basis of these results we then discuss the necessary adaption towards goal-oriented usability methods for agile settings.

## 3. Study: Programmers high-level goals (values) in XP settings

This study was done to better understand programmers in agile teams in order to adapt and introduce usability methods. We start with the definition of the study's goals and proceed with a reflection on the method that we used ("laddering interviews") and its underlying theory: the means-end theory.

### 3.1 Study goals

This study was conducted to uncover the motivational aspects of agile settings. In particular the studies goals are:

- Identify programmer's high-level goals (values), corresponding attributes and consequences as perceived by agile developers.
- Display the results in a visual way in order to show correlations between attributes, consequences and high-level goals (values).
- Align our study's results with existing knowledge and postulate new findings in reference to agile team perception.

### 3.2. Hierarchical value maps and laddering interviews

Hierarchical Value Maps (HVM) are based on the means-end theory and represent users´ underlying cognitive structures when using a product. HVMs are built to identify important meanings that consumers associate with products (Reynolds, 2001). Overall, the means-end theory distinguishes three abstraction levels of meanings: *attributes*, *consequences* and *values*. First, attributes are on equal terms with characteristics of a product (e.g. a hard disk on a mobile multimedia product). Consequences are more abstract and describe possibilities offered by the product's characteristics; hence, the product is enabling the user to execute particular actions and behaviours (e.g. the hard disk of a mobile device may be used to store certain amounts of songs, which may then be shown to friends). Lastly, values represent abstract meanings, motivational constructs and beliefs that are directly tied to emotions (e.g. presenting a song-collection to a friend is *entertaining* and makes someone *happy*). As a result the means-end theory creates links between the different levels of abstraction and shows which attributes are important to users and to what consequences and values these attributes relate. The link between attributes, consequences and values is referred to as a Means-End Chain (MEC).

The methodology has its origin in marketing research and was used – as the description above highlights – to link basic product features to high-level goals (values). However, we have used the method for different purposes and concluded that under certain restrictions it is as well applicable for the purpose of this study, which is not dedicated to particular product but to an organisational setting.

Different problems have been reported in reference to laddering and means-end theory depending on the way HVMs are used[3] (Grunert and Grunert 1995; Gutman, 1997). These problems mainly occur when HVMs are used as a representation of cognitive structures that are used to explain or predict (for instance) buying behaviour. In our study – in contrast to this "cognitive view" - we use a HVM that represent a "motivational view" that gives a deeper insight to a specific area – in our case agile development – without claiming any power to predict or explain behaviour. HVMs that are used as a "motivational view" (which is how the authors interpret the results of this study) mainly avoid the reported problems (Grunert and Grunert 1995; Gutman, 1997).

### 3.3. Study setup and participants

We conducted semi-structured, qualitative, "laddering" interviews with 10 participants. We chose this method as it was designed to extract relevant means-end chains and values related to a product or service. We had already used the method in the realm of mobile multimedia and experienced it as a powerful tool for the extraction of particular behaviours and motives. Respondents were in the age of 25 to 40 (mean 33) and professional programmers. All participants were male and were at the time of the interview an active member of an XP developer team. The programmers interviewed were not part of the same agile development team; in contrast they were members of different agile teams (working even in different companies). As mentioned in the introduction of this paper, this study is part of a project in which the authors are collaborating with an agile developers team in order to orchestrate usability methods with agile structures. It is important to mention that for the interview series no programmers of this particular project were interviewed. Instead all participants were recruited from project external sources.

To create a better understanding of the methodology that were used, the next chapter gives an overview on the interview conduction as well as on the data analysis phase.

### 3.4. Interview conduction and data analysis

To indicate how Laddering interviews are conducted the following example taken from the study at hand is provided:

*Interviewer: How do you personally perceive the XP team setting, how would you describe the XP setting?*

*Interviewee: I think it's a pinpointed work setting.* [Attribute: "pinpointed"]

*Interviewer: Why is this important to you?*

*Interviewee: I like this as this gives me the impression to work on something qualitative, something that is more likely to be used.* [Consequence: "increase quality of the product"]

*Interviewer: Why is this important to you?*

*Interviewee: It's somehow that I feel my work is honoured.*

*Interviewer: Why is this important to you?*

*Interviewee: It's a kind of award and appreciation to my work.* [value: "appreciation / reward"]

The interviews talking to 10 participants provided us with 33 valid interview sequences - like the one described above - starting from attributes ("how do you personally perceive the XP team setting?") over consequences to values ("why is this important to you?"). Note that an interview sequence was considered as valid if it was started by an attribute, contains at minimum one consequence and

[3] The authors do not consider any further discussion on methodological issues in relation to means-end theory as relevant for the focus of this paper.

terminates in a high-level goal/value – the interview sequence was considered as invalid if it failed to terminate in a high-level goal/value due to different reasons, e.g. the respondent was not able to formulate a high-level goal. Overall, there were 4 invalid interview sequences that were not taken into account in data analysis. One interview sequence is also called a "Ladder", describing the subsequently raised level of abstraction in the interview sequence.

From answers to Content Codes: The next step in the analysis procedure was to merge the 33 valid interview sequences. In this step similar answers were grouped and summarized in certain categories, so-called "Content Codes". The challenge in this step was to identify answers with similar meanings among respondents and to find meaningful names for these categories ("Content Codes"). For instance, answers from respondents that refer semantically to the attribute "pinpointed" were classified under one Content Code, even if each of the respondents had named it by different wordings (e.g. "very accurate" and "exactly defined" would be classified under one category as they refer semantically to the same quality of the object in question). The result of this step was an overview of several groups of attributes, consequences and values that were mentioned by the participants and clustered by the authors. The groups are summarized in Table 1. Overall we identified 5 groups of attributes, 12 consequences and 4 groups of values. For each group we chose a meaningful name representing similar types of answers.

| | **Attributes** | **Nr.** |
|---:|---|---:|
| 1 | challenging | 6 |
| 2 | flexible and adjustable | 3 |
| 3 | team-oriented and communicative | 12 |
| 4 | pinpointed and quality enhancing[4] | 7 |
| 5 | easy and relieving | 5 |
| | | |
| | **Consequences** | |
| 6 | create something new | 2 |
| 7 | like diversified work | 3 |
| 8 | like challenging work | 4 |
| 9 | work on a democratic basis (communication and feedback) | 8 |
| 10 | define exact goals and pieces of work | 2 |
| 11 | work on something valuable/qualitative (enhancing quality) | 7 |
| 12 | work within a simple / changeable process / environment | 6 |
| 13 | work on interesting / personally preferred things | 2 |
| 14 | like felt progress of work (productivity) | 8 |
| 15 | avoid errors | 2 |
| 16 | learn new things/solutions at work | 10 |
| 17 | more motivated / engaged | 6 |
| | | |
| | **Values** (taken from Rockeach's Value List) | |
| A | ambition / professional and personal advancement | 12 |
| B | appreciation / reward | 3 |
| C | satisfaction | 5 |
| D | happiness / relief / relaxation | 13 |

*Table 1: Content Codes are categories summarizing similar respondent's answers. The last column indicates the number of answers that were classified in this category.*

[4] The expression "quality enhancing" summarizes participant's comments, which indicate that agile setups enhance the ability to develop a software product with a higher quality (in comparison to non-agile settings).

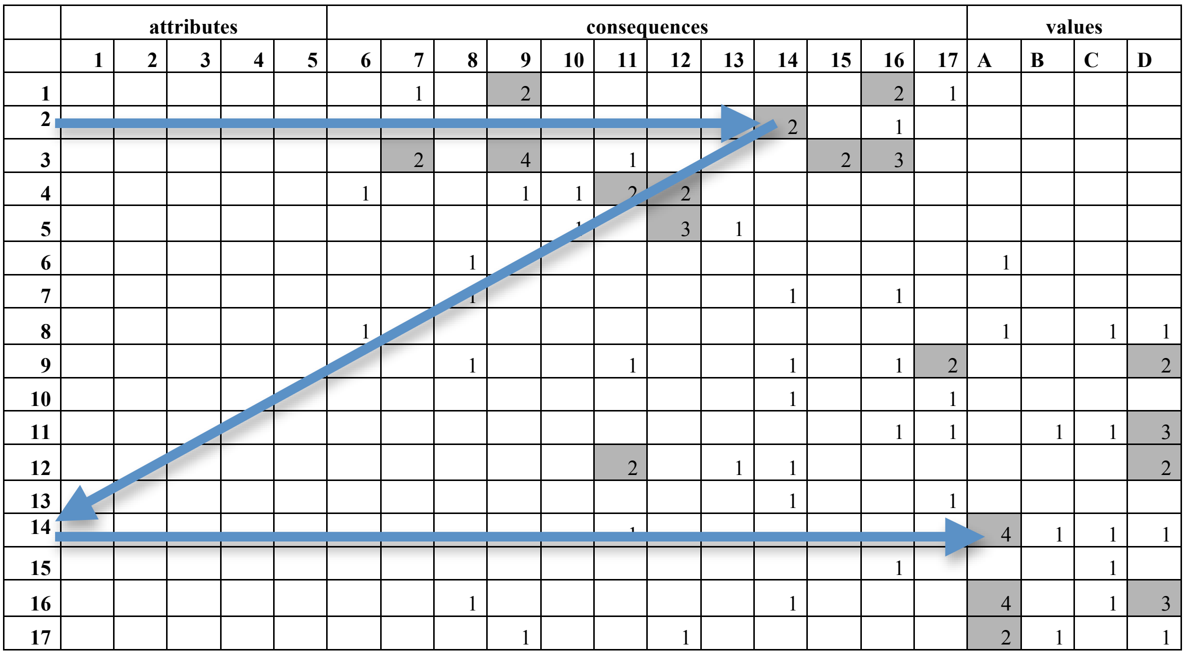
<u>*From Content Codes to Ladders:*</u> In the subsequent task this classification and grouping of answers on the basis of Content Codes allows the identification of the most relevant means-end chains. As discussed above, a means-end chain describes the correlation between attributes, consequences and high-level goals (values). In other words a means-end chain describes how a particular attribute, for instance an organisational structure that is perceived as pinpointed and quality enhancing, is supporting a particular consequence, for instance the ability to "increase the quality of a product" and finally is terminating in a certain high-level goal (value), for instance in the feeling of "happiness and relief". Note that a means-end chain is a result of several Ladders (interview sequences).

Once Content Codes were defined, each participant's interview sequences were translated into a sequence of Content Codes. Table 2 indicates a respondent's (respondent 2) interview sequence already translated to Content Codes. The proper translation of the Ladder displayed in bold characters indicates that respondent 2 stated that XP programming in his opinion is "team oriented and communicative" (Content Code 3). This is important to him as he "likes diversified work" (Content Code 7). Further, this is important to him as he "likes the felt progress and felt productivity" (Content Code 14), which is enhanced by the team-oriented and communicative setting of XP. Finally the felt progress is important to him as this increases his feeling of "satisfaction" (Content Code C). Note that Ladders are not bound to any particular length, however, each Ladder starts with an attribute, succeeds with one or more consequences and terminates with a high-level goal (value). In the given example the attribute ("team oriented and communicative"), as well the two consequences ("like diversified work" and "felt progress and productivity") support the value "satisfaction". This means that every item of a means-end chain is important to support a specific value. However, some of these items are directly "connected" to a value, some of them are indirectly related to a value (In the given examples "satisfaction" has a direct relation to "felt progress and productivity" and two indirect relations to "like diversified work" as well as to "team oriented and communicative").

| Attribute | Consequences | | Value |
|---|---|---|---|
| 4 | 6 | 8 | C |
| 3 | 15 | 16 | A |
| **3** | **7** | **14** | **C** |

*Table 2: Respondent's 2 interview sequence transformed to Ladders indicating the Content Codes in which the answers where classified.*

<u>Implication Matrix:</u> Identifying means-end chains on the basis of several Ladders follows a pre-defined procedure (Reynolds, 2001). The categorized total amount of 33 Ladders was summarized by the so-called "Implication Matrix" as shown in Table 3. This matrix indicates the number of direct relations between the particular Content Codes in the total expenditure of Ladders. Once the Implication Matrix is constructed one is able to identify the most important nodes. In Table 3 the Implication Matrix indicates all relations higher than 2 (marked in grey). This is the cut-off level we chose for the given study. This means that relations are considered as relevant if at least two Ladders named by respondents show a direct relation. To analyze one starts with the first row in the application matrix observing nodes with a number of relations higher than the cut-off level. For instance, in Table 3 junction 2/14 shows a number of 2 direct relations. Further, one skips the row to Content Code 14. Again, node 14/A shows an amount of relations higher than the cut-off level. The constructed means-end chain is: 2, "flexible and adjustable" (attribute) – 14, "like felt progress of work and productivity" (consequence) – 4, "ambition / professional and personal advancement" (value).



| | attributes | | | | | consequences | | | | | | | | | | | | values | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **A** | **B** | **C** | **D** |
| **1** | | | | | | | 1 | | 2 | | | | | | | 2 | 1 | | | | |
| **2** | | | | | | | | | | | | | | 2 | | 1 | | | | | |
| **3** | | | | | | | 2 | | 4 | | 1 | | | | 2 | 3 | | | | | |
| **4** | | | | | | 1 | | | 1 | 1 | 2 | 2 | | | | | | | | | |
| **5** | | | | | | | | | | 1 | | 3 | 1 | | | | | | | | |
| **6** | | | | | | | | 1 | | | | | | | | | | 1 | | | |
| **7** | | | | | | | | 1 | | | | | | 1 | | 1 | | | | | |
| **8** | | | | | | 1 | | | | | | | | | | | | 1 | | 1 | 1 |
| **9** | | | | | | | | 1 | | | 1 | | | 1 | | 1 | 2 | | | | 2 |
| **10** | | | | | | | | | | | | | | 1 | | | 1 | | | | |
| **11** | | | | | | | | | | | | | | | | 1 | 1 | | 1 | 1 | 3 |
| **12** | | | | | | | | | | | 2 | | 1 | 1 | | | | | | | 2 |
| **13** | | | | | | | | | | | | | | 1 | | | 1 | | | | |
| **14** | | | | | | | | | | | 1 | | | | | | | 4 | 1 | 1 | 1 |
| **15** | | | | | | | | | | | | | | | | 1 | | | | 1 | |
| **16** | | | | | | | | 1 | | | | | | 1 | | | | 4 | | 1 | 3 |
| **17** | | | | | | | | | 1 | | | 1 | | | | | | 2 | 1 | | 1 |

*Table 3: Implication Matrix: From Content Codes and Ladders to means-end chains and the HVM. Arrows indicate the construction of one "mean-end chain" on the basis of the implication matrix. Other means-end chains are constructed similarly and are displayed in the HVM. In this study we did consider a "cut-off" level of 2 meaning that attributes, consequences and high-level goals (values) are considered if they exceed two mentions (entries marked in grey).*

In this manner the whole HVM (see Figure 1) was constructed. Doing all relevant nodes of the Implication Matrix where taken into account (= all nodes that show a number higher than the cut-off level, in Table 3 highlighted in grey). The HVM is a visual summary of all relevant relations and important Content Codes (Figure 1). Note that Content Codes with no relevant relations are not further considered (In our case Nodes below 2). Further details on the analysis process are presented in (Reynolds, 2001).

## 3.5. Study Results

On the basis of that analysis procedure the visual layout in Figure 1 was derived, representing the Hierarchical Value Map for XP Programmers. It describes the most dominant answers grouped and classified by the defined Content Codes. Note that overall the interview series with the 10 respondents produced 33 Ladders (One participant constructed more than one Ladder). Furthermore only the groups of answers (Content Codes) with a number of mentions higher than the cut-off level are displayed in the Hierarchical Value Map meaning that only the strongest correlations between Content Codes are considered in the construction of the HVM.

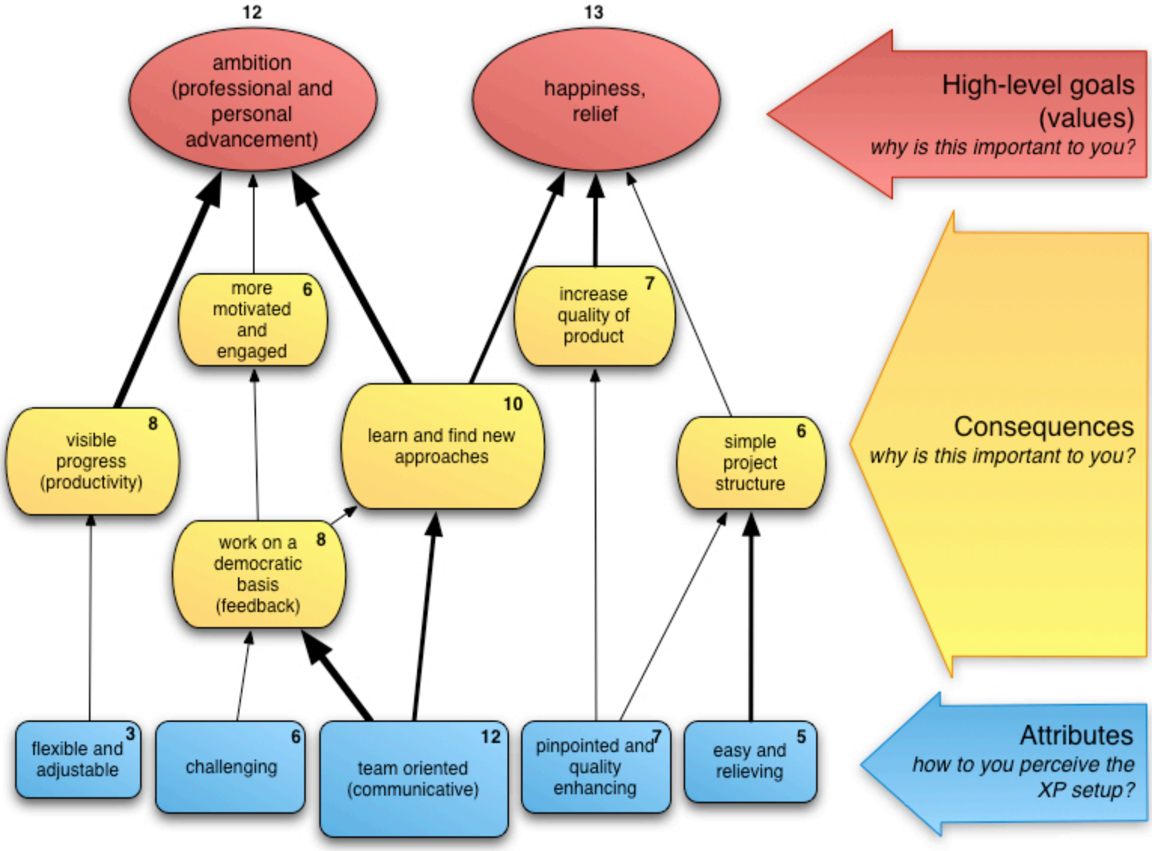

12
ambition (professional and personal advancement)
13
happiness, relief
High-level goals (values)
why is this important to you?
more motivated and engaged 6
increase quality of product 7
visible progress (productivity) 8
learn and find new approaches 10
simple project structure 6
Consequences
why is this important to you?
work on a democratic basis (feedback) 8
flexible and adjustable 3
challenging 6
team oriented (communicative) 12
pinpointed and quality enhancing 7
easy and relieving 5
Attributes
how to you perceive the XP setup?

*Figure 1 – Hierarchical Value Map for XP Programmers. At the bottom the named attributes are displayed, in the middle the resulting consequences and at the top level the high-level goals (values). The digits in the body of the items indicate the number of answers that are categorized under this Content Code (Compare to Table 1). The thickness of arrows indicates the strongness of the correlation between the items (the thicker the stronger a correlation – Compare to Table 3). Note that for reasons of readability the described Content Codes in Table 1 are abbreviated.*

Figure 1 gives an overview of the relevant items that were extracted of this interview series. Overall, the HVM gives a very personal view of programmer's perception of XP settings. As general conclusion we are able to state that XP is highly people centred and emotionally satisfying due to its team and organisational structure. The most important and interesting facts of this study on high-level goals and values are summarized in the following way:

<u>Team orientation and inter-team communication:</u> Figure 1 clearly indicates the most important fact influencing the perception of XP settings is the communicative and team-oriented work setting. This basic attribute supports a) the work on a democratic basis, which is heavily important to programmers as it provides several channels for feedback on their own work and b) due to the dynamic nature of the XP setting it gives programmers the impression to be able "to learn and find new approaches".

<u>Learn and find new approaches:</u> This consequence was identified as the most important consequence. In the interview series the study participants expressed that working on a "collective code" provides possibilities to access colleagues and co-workers solutions on particular problems. Further and on the basis of participant's comments and statements "pair programming" – a more interactive way to learn from co-workers - was perceived as a "forum" to learn and benefit from each other.

<u>Ambition (professional AND personal achievement):</u> As indicated in Figure 1 the study series revealed that XP settings are able to benefit programmers beyond the professional working borders. In the interviews participants expressed that by this flexible work setting they are able to align their work

to their personal interests. This lead to a perception of achieving something in a professional but as well in a personal way as the setting is able to support personal interests and ambitions.

The findings displayed in the Hierarchical Value Map confirm the results found in other studies. Especially findings of Whitworth (2007) who indicated that team-awareness, motivation and enhanced effectiveness and increased self-esteem are one of the most important points in agile settings. Other findings referring to satisfaction and better self-image are related to the findings of this study. However, by applying the means-end theory and using laddering interviews we tackled this area of research with a new and fruitful approach. Our study clearly provides new insights by structuring the findings in attributes, consequences and values. Secondly, an added value of the hierarchical visualization (=HVM) is identified, which shows linkages between attributes, consequences and high-level goals (values) in a structured way.

In the following step we use these findings on high-level goals (values) and motivational aspects of agile programming to discuss the use and necessary adaption of usability and user-centred methods in agile team settings.

## 4. Impact on Usability Methods in Agile Teams

### 4.1 Agile-oriented Usability Methods

In this section the study results are used to discuss different enhancements of usability methods for a better application in agile setups. The objectives are twofold:

a) We discuss the necessary adaptation and positioning of usability methods upon the results of this study on programmers values. In this chapter we define a strategy of how to integrate and adapt the usability methods in order to support the programmer's goals. In order to do so we elicited the most potential programmer values and motivational aspects from the foregoing study from the current point of view. For a deeper discussion about value impacts on the use of user-centred methods a broader range of values can be taken into account. For the current study we aim to consider the most predominant values for our purpose.

b) As mentioned in the introduction of this article the described study on programmer's values is part of a project dedicated to the orchestration and enhancement of usability methods in agile teams. In the following chapters only the usability and user-centred methods that are used in this project are discussed, which are the following (compare to [Wolkerstorfer, 2008]):

   - *Personas:* These are archetypical descriptions of real users, representing the target user group. Personas are often described in a narrative way and are designed to help software developers to get a better understanding of the real end-user they are developing for (Pruitt, J. and Adlin, 2005).
   - *Usability evaluation and ad-hoc usability (expert) evaluation/input:* These are ongoing usability reports (of the currently available solution) and as well ad-hoc inputs on current issues in development (given orally or face2face or via video or audio-conferencing).
   - *Automated usability evaluation (system evaluation) as extension to unit tests:* These are test cases that are integrated to the nightly builds. These tools should be able to indicate a certain level of usability of the current version. For this usability method we draw some principle implications but do not go in any detail due to the lack of practical experiences.

For our purposes these methods need to be adapted to the specific agile context in order to be better aligned to programmers' goals. The overall objective is to position these adapted methods in a way that leads to a higher acceptance level and an increased use (for teams that have no specific team member working on usability).

In order to enforce our methodological suggestions we discuss practical experiences that we gathered using "Personas" and "Usability evaluation and ad-hoc usability input" in practise. We are aware that this feedback is no valid proof of the methodological improvement suggested by this paper, however we believe that these short case studies provide a good insight and support our arguments. Note that for automated usability evaluation (AUE) no practical experiences are available at the moment. For this part we draw principle conclusions without any further discussion.

## 4.2. Implications and Conclusions for Usability Methods in Agile Teams

For each of the usability method defined above particular potential and necessary adaptations are required. For each of the methods that are discussed in the following paragraphs we elicited the high-level goals (values) that were found in this study and that have the most potential impact on the methods from the current point of view. On the basis of these selected results we conduct the following discussion. We are aware that for a broader discussion several values of the foregoing study can be used, however, for the purpose of this study we stick to the most dominant ones. We are further aware that these suggestions need to be evaluated in the field before we can draw any final conclusions. In general and on the basis of this paper a more detailed study evaluating the suggested improvements needs to be conducted.

**Personas:** In order to use personas in agile settings this methodology needs to be adjusted towards a more flexible use in the project setting. The traditional definition of personas foresees no changes to a defined personas description. In classical development process the personas are defined once (typically at the beginning of a project) and are not changed during the project. This method is too rigid for XP projects when it is used in its traditional form.

Personas can be used to focus on the programmer's consequences of "visible progress" and "increase quality of the product" in the following ways:

- Visible progress: It is very important for programmers to see the progress of their work. Static personas descriptions do not support this programmer goal. The implication is that personas need a list of "usability problems" (or something similar) that needs to be updated according to the work done by the developer team. This feedback serves as a "visible progress" as the list of usability problems for a specific personas decreases due to a checked-in solution. Likewise the usability problems of a persona could as well be added due to a current solution that was checked in. This would as well lead to a more visible progress in reference to usability problems.
- Increase quality of the product: Overall we argue that the use of personas in agile team settings supports programmer's aim to increase the quality of the product. Whereas in the interview study "quality of code" was mainly used as an indicator of quality, we believe that personas (or usability in general) can serve as alternative "metric" of quality of a software product in XP setups. In order to do so the persona marketing method must be enhanced (e.g. personas can send e-mail to programmers or personas can communicate over the programmer's mailing list discussing progress of the solution). Such approaches would introduce a new metric of the quality beyond "quality of code" (which is of course an important metric – however, new ways of quality assurance can be introduced by this way).

To strengthen these conclusions we report the following practical experiences using personas in agile settings[5]: Applying this method in our project on user centred-methods in agile setups we were using a quite traditional personas approach for the agile developers team. After a while we noticed that we used too little marketing and that we failed to adapt the personas description in a flexible way. The result was that the personas were not visible and therefore not used or considered as important in the project as we expected them to be. As extreme example of our case study the personas description posters were removed from the developer's room. We are aware that this is just one experience of a particular project and the reasons for the removal have not been discussed in detail with the agile programmers team, but we believe that this example is a quite striking describing the potential (mis-) use of personas in XP projects.

[5] Note: These observations were collected in a project that used agile development (as mentioned in the introduction of this paper). In this project the personas method was used without the suggested improvements claimed in this paper.

**Usability evaluation and ad-hoc usability input:** Traditionally, usability reviews done in structured ways result in a usability report that might be considered at various stages of a software project. Such reports may have numerous pages listing different problems and solutions. We believe that this form of input is not suitable for agile settings. Likewise the following programmer's values are not considered by this method:

- Simple project structure: Usability input has to consider the need and the expectation of a simple project structure. Highly elaborated usability reports are not supporting this goal. In contrast usability input needs to be selective in the form of little and well-structured pieces. We propose to feed the developer team with particular "usability user stories" and possibilities of ad-hoc usability input. This fits the programmer's expectations better than whole usability reports (with a lot of pages to consider and to disseminate, etc.)
- Work on a democratic basis: Referring to ad-hoc usability input the usability experts need to be accepted as "full project team member". We argue that by imposing usability reports on the team the developer's sense of a "democratic setting" is decreased. In contrast a bidirectional relationship has to be established.

To strengthen these conclusions we report the following practical experiences for the use of usability reports and ad-hoc expert usability input[6]: Working with the agile programmers we applied both, usability reports and ad-hoc input that was requested by programmers whenever they need the input. We noticed that the programmer teams did not value large usability reports in classical forms as much as expected (note: often a 40 pages report). Programmer's feedback was that they perceived this input as too exhaustive. In contrast, usability bugs reported like user stories had major impact (and were often considered). We got very positive feedback on this form of usability expert input e.g. via given via Skype or face-2-face. We got the impression that orally discussed problems (in contrast to written formal reports) satisfy the attributes of team-orientation and "simple project structure" in a much better way. An other form of creating usability awareness with major impact was observed during usability testing. As has been found elsewhere, programmers who were physically present during usability tests, watching and observing users dealing with their software, were observed to be quite concerned with usability.

**Impact of automated usability evaluation (AUE):** Automated usability evaluation can be a fruitful instrument to create a metric on the basis of ongoing code development. The idea of this setting is to check usability automatically as extension to unit tests. This could serve – similar to personas descriptions – as a qualitative metric of increased or decreased usability index. We argue that this would support the programmer's consequences "visible progress" and the wish to "increase the quality of the product". We conclude automated usability evaluation tools should be an add-on to the unit-tests with visible output in relation to usability metrics for each nightly build report (or similar). A set of metrics for each build (graphs or numbers) has to be introduced that communicates results to programmers in order to guarantee awareness. In this way AUE should be included in the workflow of the developers. As mentioned above, for these suggestions no practical experiences are reportable at the moment.

Summarizing we conclude that there is a need for a well-defined mix of usability methods (Without claiming that the methods that are discussed above constitute the whole range of methods that should be used in a setup – of course there should be more methods brought together). However, applying traditional usability methods in agile setups might lead to mis-use or non-use of these methods. In contrast, different adapted methods should be available that can be used flexibly by programmers. Overall the strategy is not to apply as many methods as possible within the agile setting but a good and well-defined range of methods that are adapted to the context of use accordingly.

[6] Note: These observations were collected in a project that used agile development. In this project usability evaluation was conducted without the suggested improvements claimed in this paper.

## 5. Conclusion and Future Work

We conducted a study on programmer's values in agile settings. The results of this study were used to discuss and adapt usability methods for the use in such agile settings.

With this study we are able to extend the current state of the art in two ways:

- *Identification of programmer's high-level goals (values):* We confirmed and extended current studies in the area of "perception of agile team settings" and introduce a new way of visualizing different facts about XP programmer values. We conclude that "team-orientation" is one of the most important facts about agile settings. Programmers like the fact that they are able to "learn and find new approaches" and that XP settings highly support their ambition (in a private AND professional sense). The presented HVM is able to show interrelations and correlations between different findings, which further enhances the understanding in this area of research.
- *Adapt usability and user-centred methods for agile settings*: We discussed the user centred methods "personas" and "usability reporting and ad-hoc usability input" and discussed some practical experiences in adapting them to an agile context. Our main conclusion is that these methods (except ad-hoc usability input) are too rigid in their current and classical way of use. Further, clear metrics need to be defined that communicate the progress in reference to usability. This has to apply to all of the discussed methods. As there is little know-how available so far on how usability methods need to be extended and adjusted for agile teams, with this work we lay the basis for further research in the area of usability in agile team settings by suggesting concrete and necessary enhancement of user centred methods and their positioning in agile teams.

We are aware that our current observations of the applicability and of the added value of our proposed adaptations of usability methods rely on qualitative observations and qualitative feedback of programmers. As future work we are aiming at measuring the impact of these adapted usability methods in a quantitative way.

## 5. References

Bjørnar Tessem, Frank Maurer, Job Satisfaction and Motivation in a Large Agile Team in Agile Processes in Software Engineering and Extreme Programming, (Page 54- 61), 2007; DOI - 10.1007/978-3-540-73101-6_8; http://www.springerlink.com/content/gh18347t4172k769

Constantine, L. L., & Lockwood, L. A. D. (2003). Usage-centered software engineering: an agile approach to integrating users, user interfaces, and usability into software engineering practice. In ICSE 2003: Proceedings of the 25th International Conference on Software Engineering (pp. 746-747). Los Alamitos: IEEE Computer Society.

Ferreira, J., Noble, J., & Biddle, R. (2007). Agile development iterations and UI design. In: Agile 2007 (pp. 50-58). Los Alamitos: IEEE Computer Society.

Göransson, B., Gulliksen, J., & Boivie, I. (2003). The usability design process – integrating user-centered systems design in the software development process. Software Process: Improvement and Practice, 8(2), 111–131.

Grunert G. Klaus; Grunert C. Suzanne; Measuring subjective meaning structures by the laddering method: Theretical considerations and methodological problems; International Journal of Research in Marketing 12, 209 – 225, 1995

Gutman, Jonathan; Means-End Chains as Goal Hierarchies; Psychology & Marketing, Vol. 14(6): 545-560, 1997

Holzinger, A. (2005). Usability Engineering for Software Developers. Communications of the ACM 48(1), 71-74.

Holzinger, A., Errath, M., Searle, G., Thurnher, B., & Slany, W. (2005). From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE→XU). In COMPSAC 2005: Vol. 2. Proceedings of the 29th Annual International Computer Software and Applications Conference (pp. 169-172). Los Alamitos: IEEE Computer Society.

Hudson, W. (2005). A tale of two tutorials: a cognitive approach to interactive system design and interaction design meets agility. Interactions 12(1), 49-51.

Kent Beck, Extreme programming explained: embrace change, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

Law, Amy; Charron, Raylene; Effects of Agile Practices on Social Factors; Human and Social Factors of Software Engineering (HSSE) May 16, 2005, St. Louis, Missouri, USA

McInerney, P., & Maurer, F. (2005). UCD in agile projects: dream team or odd couple? Interactions 12(6), 19–23.

Miller, Lynn and Sy, Desiree, Agile user experience SIG, in: CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, Boston, MA, USA, pages 2751--2754, ACM, 2009

Norman, D. A. (2006). Logic versus usage: the case for activity-centered design. Interactions 13(6), 45-ff.

Pruitt, J. and Adlin, T. 2005 The Persona Lifecycle: Keeping People in Mind Throughout Product Design (The Morgan Kaufmann Series in Interactive Technologies). Morgan Kaufmann Publishers Inc.

Reynolds, Thomas J. and Olson, Jerry C.; Lawrence, Understanding Consumer Decision Making, The means-end approach to marketing and advertising strategy; Erlbaum Associates, 2001

Rokeach, Milton, The nature of human values. New York, Free Press

T. Dyba˚ , T. Dingsøyr, Empirical studies of agile software development: A systematic review, Inform. Softw. Technol. (2008), doi:10.1016/j.infsof.2008.01.006

Whitworth, E. and Biddle, R.; The Social Nature of Agile Teams; Agile, 2007, AGILE‘07, Conference, (26-36), 2007

Wolkerstorer, P.; Tscheligi, M.; Sefelin, R.; Milchrahm, H.; Hussain, Z.; Lechner, M.; Shahzad, S.; Probing an agile usability process; in: CHI '08: Proceedings of the 26th international conference extended abstracts on Human factors in computing systems, Florence, Italy, pages 2151--2158, ACM, 2008