

# Using personas to guide iterative development

Anders Toxboe

anderstoxboe.com – anders@toxboe.net

Copenhagen, Denmark

+45 24409811

## ABSTRACT

In this paper, I introduce a combination of User Centered Design (UCD) and eXtreme Programming (XP) that seeks to bring the users closer to the center of attention than in XP. The new methodology is used to guide decision-making in XP with use of personas and scenarios.

## Keywords

Extreme programming, agile development, personas, scenarios, iterations

## 1. INTRODUCTION

In this paper, I introduce the use of personas to guide decision-making in the agile development methodology of eXtreme Programming (XP). The use of personas is a User-Centered Design (UCD) tool that allows the development team to engage in- and understand the minds of fictional users. A connection between the user stories of XP and personas of UCD is the link between XP and UCD.

First, XP and UCD are each introduced and compared, where after relevant critique from the literature is reviewed. Further, it is examined how the two can complement each other to form a new approach, which combines the two.

## 2. EXTREME PROGRAMMING

Extreme Programming (XP) is an explorative and agile development method[1] that seeks to satisfy the customer through early and iterative delivery of valuable software. XP welcomes changing requirements through the whole development process by being flexible with its short iterations (2-4 weeks), that allow rapid feedback, and tools like refactoring[2] and unit testing, that allows simplistic coding[3]. Together, this steers the project in the right direction, which in turn decreases the risk of failure[4].

XP is based on old and proven practices combined in a new way to make up for weaknesses in the more traditional phase-divided methods like the waterfall-model.

XP swears to the principles of the *Manifesto for Agile Software Development*[1]. The Agile Manifesto represents the thoughts behind XP:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

There is a focus on quality rather than scope as in non-iterative methods. This is exemplified by for instance pair programming, collective code ownership, and setting a 40-hour workweek to relieve the programmers from stress.

Roles in XP are divided into two: The customer and the developers. The term “customer” is broad in XP. It covers both the person who makes the business decisions, users, business management, operations, and more. The customer represents both the end-user and the decision-makers[3,4].

### 2.1 User stories

The customer starts the project by presenting the complex problem domain to the development team. The problems to be solved are divided into smaller bits of functionality called user stories, that can each be implemented in the time of one or two iterations. A user story is a one to two sentences long description of the specific functionality. An example of a user story is:

*“The system should check the spelling of all words entered in the comments field.”* [4], p. 14

In the beginning of each iteration, a planning meeting is held. Here the user stories, which are to be implemented in the oncoming iteration, are chosen. The prioritization of user stories is based on what user stories bring most value to the project as well as an estimate on how long they take to implement.

### 2.2 Testing

Functional acceptance tests with XP test scenarios for each user story, is used to verify that stories have been correctly implemented. XP test scenarios are mechanistic in nature, and can automatically run without human involvement.

## 3. USER-CENTERED DESIGN

User-centered design exists in many shapes and forms. From being an integrated part of system development methodologies like OOA&D[6] to being stand-alone tools with no plan for implementation.

UCD places the person and not the system at the center, by trying to answer questions about the goals and tasks of users, in order to direct the development and design of a product. The final product can be either a physical product, a software product or even a process: anything with which the users interact. One UCD technique is a plain evaluation of an existing system – another, writing personas, is a detailed way of describing the product’s users.

UCD in itself has no plan for implementation, as it is just a set of tools, which all bring the user to the center of design.

The UCD toolbox is large, but a few select techniques are appropriately combined with XP. These techniques go along well with the principles behind the Manifesto for Agile Software Development. Especially personas[8] and the complementary scenarios[8] were found to be directly connectable to work within the iterative XP development loops. In this particular project, I also used cultural probes[9] for incepting the basic idea.

### 3.1 Personas

Personas are specific and realistic written descriptions of archetypical fictional users. The personas serve as a communications tool to explain and end design discussions.

The developers design for some personas more than others, why arguments from the perspective of one persona can have larger weight than from the perspective of a less important persona. A few personas can even be created to define whom the designers should not design for.

## 4. CRITIQUE OF METHODS

### 4.1 Extreme Programming

Even though a large part of the critique directed towards XP has been rejected by prior mentioned research, one point still remains un-defended: too little user involvement in the design process. XP discards the phase-structure in the traditional waterfall-model by letting the programming, requirements gathering, and planning iterate, thus making the scope of the project variable instead of fixed. Big initial requirements gathering and documentation are dropped in order to focus more on quality of the product itself inside the development loops.

XP techniques like pair programming have been suspected to be cost-inefficient, but research results[5] show that efficiency increases by 15% when using pair programming. Other gains from pair programming include a higher quality of work, collective ownership of code, and less stress. These results reject the majority of the critique of XP.

When Kent Beck originally introduced XP's customer, a real end-user was in mind, as he believes in the team's direct accountability to the end-users. Unfortunately, not all XP projects have followed this model. The broader picture with multiple end-users was cut away with the big initial requirements analysis and degraded to fit under the hat of the "customer", who makes business decisions as well as represent the end-users. Even though many advantages have been achieved by the new structure, a new problem has been created: end-users are not represented in their *diversity* and detail in XP.

### 4.2 User-centered design

When integrating UCD and XP, the big question is where decisions are made. Many UCD tools suggest that most if not all decisions about the final product are written down in the initial requirements document, which is created before any actual coding starts. These decisions are made on a

basis of thorough initial analysis of the end-users. From an XP perspective, this is a point of critique as XP argues for the opposite. Considerable decision-making is happening in each development loop in XP (every 1-3 weeks), instead of only once before coding starts. The decision-making and requirements gathering in XP is spread out over the life of the project.

In many cases, interaction design and UCD can be seen as the same thing as interaction design uses UCD tools in its processes. In a debate with Alan Cooper (Interaction Design), Kent Beck (XP) argues that the interaction designer becomes a bottleneck, as all decision-making are at one point. Instead of adding early requirement gathering and design phases to XP, Kent Beck argues that interaction design would be more powerful if integrated inside the loop of development according to the extreme programming practices[10].

Most user-centered design tools wish to give an answer to the whole solution of the problem up front. When the solution to the design-problem is created, implementation of the solution is estimated. The estimation leads to the allocation of a fixed time-slot and a fixed amount of resources for developing the solution to the design problem. This shows a focus on defining scope from the start of the project that is not found in XP.

Lene Nielsen explains[8], that a main obstacle with using personas is poor description of the step from personas to code. This paper seeks to do give a description.

### 4.3 Comparing XP and UCD methods

	XP	UCD
Design	Iterative delivery of design	Up-front design
Changing requirements	Embraces changing requirements	Neglects changing requirements
Focus	Quality	Scope
Choice of functionality	Most value to the system.	The found solution that solves the problem.
Testing	Continuous unit- and acceptance testing	Test in the end
Documentation	No written documentation. Real-life customer/user serves as documentation	Designs up-front in written requirement documents
Planning	Evolving plan: responding to change.	Fixed plan: trust in up-front solution

**Figure 1.** Characteristics of traditional XP and UCD

The mentioned characteristics of XP and UCD have been summed up in Figure 1 to provide a clearer view of the differences in the principles behind respectively XP and UCD.

## 5. BRINGING THE USER CLOSER TO XP

Among UCD and XP tools, I found that the best match for combination between UCD and XP was between personas[8] from UCD and user stories from XP.

## 5.1 Matching personas with user stories

Personas can serve as an expansion of the user representation within the XP methodology, support arguments about features of the system[8] as well as helping prioritize between them[11]. As each XP user story describes a feature, the mating between personas and XP seems obvious.

Programmers and decision-makers can engage in the worlds of the persona by understanding the motivation for actions done by the persona. This is why personas can make up for the lack of focus on the user in XP and make XP more user-centered.

Personas require field study of the target user group that the personas should represent. If done thoroughly, this field study takes longer than the length of an XP iteration, and should therefore not take place inside an XP iteration loop, but before the loops begins.

Personas are documentation of the results from initial field study of the users. Agile development despises documentation in large forms (documentation is not bad, but working software is better) as it forces the developers to use more time on updating and keeping updated on documentation than on delivering working software. It is therefore needed to find methods of letting the programmers engage in the personas, that do not require the programmers to spend much time. One example of this could be to randomly send out fictitious emails from the personas to the programmers[11], another is printing easy-to-read comparison posters.

There is a risk that using personas to represent the user will not make up for real-life feedback from an actual user. Using personas should, if used, be seen as a complimentary to- instead of a substitution of real users.

## 6. THE NEW METHOD

The method was tested with a project inception phase, where cultural probes, personas, and scenarios were used for incepting ideas. The output was a basic idea about the system should do and set of personas to be used to guide decision-making inside the development loops.

### 6.1 Implementing UCD with XP iterations

In traditional XP development, it is the customer, who makes the business decisions on what is to be implemented in future iterations of development. I suggest letting the personas and their complimentary scenarios play a part of this role. Needs, goals and personalities of the users as well as financial considerations should be used to evaluate which user stories (features) are more important than others.

A solution for prioritizing between user stories is to rate each story by how much value they bring to each persona as well as the financial backup. The rating should be friendly to the budget and consider that even though one user story may be more valuable to the biggest amount of users, there is only enough money to implement a user story that is half as valuable. Another consideration is that

one type of user might bring in more value than the other types of users. All considerations should therefore have separate weight in the decision-making.

The user stories are short. This makes it possible to cope with many user stories (features) when making a decision on which stories to implement and how long it will take to implement them.

The users' wishes are elaborated for the decision-makers at the planning meetings through personas. If we for instance have 4 different personas and a user story is only highly prioritized by 1 persona, we might choose to find another user story that has a broader prioritization among all 4 personas. In other words, we can use the personas both as a tool to establish a common understanding of the user group, but also as a tool for the project coordinator to prioritize between user stories.

When user stories are created in XP, the customer creates scenarios along with them. These scenarios are different from the UCD scenarios that are usually built from personas, as they are used for acceptance tests at the end of each iteration. These acceptance tests contain small steps of interaction, that can determine whether a user story has been implemented the right way or not. To test whether an implemented user story fits the needs of all personas, acceptance tests can be created for each persona.

## 7. PUTTING THE METHOD TO A TEST

A series of iterative workshops were created with a student group to test the method. The phases of each workshop, which was to act as an XP planning meeting, had the following structure:

- **Brainstorm** for new user stories (features).
- **Release planning.** Rating of each user story from the perspective of each persona as well as the customer – the financial perspective. The result was a list of user stories ordered by importance. The most important user stories were to be implemented in the next iteration.
- **Acceptance tests** were created for each user story. Each test was a black-box test that would determine whether or not the user story had been implemented correctly.

### 7.1 Brainstorm

The brainstorm was a time for reflection. Challenges faced with requirements in the previous iteration were discussed while looking at the delivered product. The change in the perception of requirements led to a brainstorm for new user stories to meet the new perceived requirements. The new user stories were written down on note cards and estimated. After this, they were shuffled with the note cards from previous iterations, so they could be part of the release planning.

### 7.2 Release planning

The resulting list of user stories at the release planning could look like this:

	Persona 1	Persona 2	Customer	Weighted sum
Weight:	20	30	50	
Show how long the movie is	2	2	0	100
Show link to trailers	2	2	2	200
Gift cards	1	1	2	150
Drinking games for films	1	-1	1	40
Free simple games	1	2	2	180
Etc.	-	-	-	-

**Figure 2. Example of how user stories were prioritized in planning meetings.**

To figure out which user stories to implement in which iteration, each user story was scored in importance by each persona. Zero was given in score if the persona had no interest in the user story, 1 if the story had some value, and 2 if the user story would do something important for the persona. The scores were then added up for each persona and soon the user stories with the highest priority were found.

After evaluating the first workshop, it was found that there needed to be some voice of the financial backup of the project. Therefore it was decided to add another point-giver called “the customer”, that could also score each user story from 0 to 2. After reviewing the literature, it was found that Jonathan Grudin and John Pruitt[11] had already done something similar before. Ideas from their priority matrix resulted in adding a -1 to the score if the persona or customer was confused, annoyed or harmed by the feature as well as adding weighted scores so that opinions from either one persona or the customer would weigh more or less than the others. The sum of the weight should be 100. That means that if everybody approves the feature and rates it 2, the sum would be 200. If the feature harms everybody, the sum would be -100.

The weight of each persona and the customer was determined by a discussion in the student group. The specific rating of a user story is based on arguments that might not have any connection to the rating of another user story. For this reason, it is important to underline, that the sum is not a definitive tool to find which user stories to implement in the next iteration, but merely a guide to help in making the decision. Adding comments to the difficult-made ratings can further help the final decision-making.

The top 20 rated user stories were chosen for further investigation. It was then decided which user stories were to be implemented in the next iteration among these regardless of their scored sum, but instead based on a human judgement of what seemed more important to implement first.

As the participants had no experience in estimating how much time user stories take to implement, we decided to skip this part. Instead the judgement of how many user stories it was possible to implement in the next few iterations were made, by dividing the top rated user stories into groups that seemed within the reach of being implementable in one iteration.

### 7.3 Acceptance tests

For each user story that was selected for the on-coming iteration, one or more acceptance tests were created. An acceptance test is a basic step-by-step scenario that should be able to run, when the user story has been implemented. XP test scenarios are mechanistic in nature, and can automatically run without human involvement. Examples of the acceptance tests are:

<i>Search for actor</i>	<i>Send link to film-trailer</i>	<i>Pay with credit card</i>
<b>Step 1:</b> Fill in “Tom” in the search text field <b>Step 2:</b> Click “Search” button <b>Step 3:</b> Find and click “Tom Cruise” link in the list of search results <b>Step 4:</b> Find the text “Tom Cruise” in the info page about Tom Cruise	<b>Step 1:</b> Find film <b>Step 2:</b> Click “Send trailer to a friend” <b>Step 3:</b> Fill in e-mail in e-mail text-field <b>Step 4:</b> Click “Send link”	<b>Step 1:</b> Click “Pay” link <b>Step 2:</b> Find the text “Order information” and title of the film selected on the page <b>Step 3:</b> Check “I accept the conditions” <b>Step 4:</b> Fill in credit card info in the form fields <b>Step 5:</b> Click “Pay” button

**Figure 3. Examples of acceptance tests on user stories**

After creation of the acceptance tests, the student group parted to develop the prototype. At the second planning meeting, the acceptance tests were run. Normally this would take place well before the planning meeting, so that any errors can be corrected. Since I had decided only to do workshops to test the method, time from the beginning of the next planning meeting were taken to verify that the acceptance tests could run.

As all acceptance tests could run, it was time for brainstorming for new user stories as the uncertainty about which way development could go had been made smaller during development. The whole treadmill of the planning meeting had then started, as the next step was release-planning, creation of acceptance tests, development, etc

### 8. WHAT IS NEXT?

The next step is to take the new method further beyond the planning game and into actual development. The main challenge here is to communicate the personas to the developers by letting them engage in the personas. Writing personas and persona-based scenarios for the developers to read is one way – another is to let the personas come to life.

Grudin & Pruitt[11] has used a variety of initiatives to let the personas come to life. Their idea is to routinely put small bits of persona information in front of the team. This can for instance be done by creating e-mail addresses for each persona in order to send out fictitious mails to the developers from each persona. They also maintained a website with persona information as well as creating posters, flyers, handouts, and giveaways that has persona information on them. Posters can include persona comparison information or just information about one persona. Grudin & Pruitt think of the persona effort as an on-going campaign and it should be.

Personas can be used for more than just prioritizing user stories. A straightforward expansion of the use of personas

is to create acceptance tests with a base in the personas. As different personas have different ways of reaching their goal, they also have different ways of using the feature implemented in the user story. One or more acceptance tests could be created for each persona rather than for what the developers think is right.

In order to find defects or things that could be better, a task-group can be assigned in an iteration to go through the software product in order to see if things can be improved for one specific persona. The result of this examination would then be turned into user stories at the next planning meeting.

## 9. REFLECTIONS

The new method has solved some of the problems that XP had when not using contract development as well as problems with XP in general. Problems solved include too little focus on the actual users of the system. For the method to be able to run perfectly smooth, some problems needs to be addressed: some that are present in traditional XP and some that has hit the surface with the introduction of this new method.

### 9.1 Problems still present from XP

The number one problem in the workshops was the process of concretizing visual design ideas into user stories. The participants requested some sort of tool, which could help concretize the visual design ideas in the time of one iteration. It was a frustration over how important a good user interface was rated in the theory, and how hard it was to use the theory in practice. The participants did not have the experience in designing user interfaces and could not engage in the personas enough to know exactly what concrete ideas the personas had about the design.

### 9.2 New challenges created by the new method

At the start of each planning meeting, the participants had fresh minds. Details of the discussions that went on at the prior planning meetings were forgotten. This resulted in the user stories being rated differently from meeting to meeting. Unless the details had been written down in details in the rating-comments, the rating of each user story had to somewhat start over. Whether this was due to new knowledge of requirements is not clear. A user story that had top ratings could easily go from a 1<sup>st</sup> to 30<sup>th</sup> place in overall rating. With numbers for ratings, and only two personas to make a difference, it is not much that makes the difference. A solution to this problem would have been to introduce more personas in order to get a more detailed overall rating.

Rating user stories with importance for each persona and customer follows a market perspective. As mentioned before, user interface stories are rated very important by the theory, but did not necessarily have a top score between other features. This can be explained by the example that getting the system up and running is more important than the font-headers being colored blue.

## 10. CONCLUSION

The use of personas to prioritize between user stories has proven to be an effective combination of UCD and XP. The conflict in interests the XP customer has in traditional XP development; by both representing financial interests of the client as well as the interest of specific end-users, has come closer to a solution. The customer, who in XP speaks the voice of all end-users, has been elaborated to include a more detailed picture of the different end-users.

The method is still agile, but is so in a more detailed sense. The different personas each represent different directions the project can go, thus the risk of missing out on certain end-user problems has been minimized.

With the extra UCD tools added to the initial phase as well as inside the development loops, the method is more explorative than XP or UCD techniques alone. Cultural probes explore and bear ideas and personas concretize them.

XP has been expanded to take more organizational problems into considerations with the personas. There has been a shift from XP's focus on the system to a more equal focus on both the system and the users.

## 11. ACKNOWLEDGMENTS

I would like to thank Lene Nielsen and Jesper Holck at the Copenhagen Business School, Department of Informatics for providing helpful feedback on my ideas.

## 12. REFERENCES

1. Manifesto for Agile Software Development. Available at: <http://agilemanifesto.org/>
2. Fowler M., *Refactoring*, Addison-Wesley, 1999
3. Beck K., *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000
4. Beck K. & Martin Fowler, *Planning Extreme Programming*. Addison-Wesley, 2000
5. Cockburn A. & Williams L., *The cost and Benefits of Pair Programming*, University of Utah Computer Science, 2002
6. Matthiasen L. et al., *Object Oriented Analysis and Design*, Marko Publishing Aps, 2000
7. Norman D., *Emotional Design*, Basic Books, 2004
8. Nielsen L., *Engaging Personas and Narrative Scenarios*, Samfundslitteratur, 2004
9. Gaver B., Boucher A., et al., *Cultural Probes and the Value of Uncertainty*, Interactions, 2004
10. Nelson E., *Extreme Programming vs. Interaction Design*, FTP Online., 2002-01-15, Available at: [http://fawcette.com/interviews/beck\\_cooper/](http://fawcette.com/interviews/beck_cooper/)
11. Grudin J. & Pruitt J., *Personas, Participatory Design and Product Development: An infrastructure for Engagement*, PDA 2002, Malmö, 2002