Dogana

ADVANCED SOCIAL ENGINEERING AND
VULNERABILITY ASSESMENT FRAMEWORK

D4.3 DOGANA Interaction Design Guidelines

Work Package:	WP4
Lead partner:	Austrian Institute of Technology (AIT)
Author(s):	P. Wolkerstorfer (AIT), Florin Dragomir (RATB), Liliana Denisa Andrei (RATB), Alan Ferrari (SUPSI), Davide Andreoletti (SUPSI), and Angelo Consoli (SUPSI)
Submission date:	15. September 2016
Version number:	1.0 **Status:** Final

Grant Agreement N°:	653618
Project Acronym:	DOGANA
Project Title:	Advanced Social Engineering and Vulnerability Assessment Framework
Call identifier:	H2020-DS-06-2014-1
Instrument:	IA
Thematic Priority:	Trustworthy ICT
Start date of the project:	September 1st, 2015
Duration:	36 months

Dissemination Level	
PU: Public	✓
PP: Restricted to other programme participants (including the Commission)	
RE: Restricted to a group specified by the consortium (including the Commission)	
CO: Confidential, only for members of the consortium (including the Commission)	

Horizon 2020
European Union Funding
for Research & Innovation

Project co-funded by the European Commission under the Horizon 2020 Programme.

Revision History

Revision	Date	Who	Description
0.1	19/07/2016	Peter Wolkerstorfer (AIT)	D4.3 TOC structure
0.2	22/08/2016- 1/09/2016	Peter Wolkerstorfer (AIT)	First DRAFT with input from partners RATP and SUPSI
0.3	8/09/2016	Peter Wolkerstorfer (AIT)	Error correction, reference clean-up, document consolidation
0.4	12/09/2016	Peter Wolkerstorfer (AIT)	Published for internal review
0.5	15/9/2016	Peter Wolkerstorfer (AIT)	Integrated feedback from Sebastian Egger, Alessio Mulas, and Enrico Frumento

Quality Control

Role	Date	Who	Approved/Comment
Internal reviewer	14.9.2016	Enrico Frumento	none
Internal reviewer	15.9.2016	Alessio Mulas	none

Disclaimer:

This document has been produced in the context of the DOGANA Project. The DOGANA project is part of the European Community's Horizon 2020 Program for research and development and is as such funded by the European Commission. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability with respect to this document, which is merely representing the authors' view.

Table of Contents

1. Executive Summary 7
2. Introduction 8
   2.1. Deliverable D4.3b 8
3. GUIs in DOGANA 8
4. User Centric Development Process 9
5. Usability Principles 12
6. User Interface Design Guidelines 13
   6.1. Visual representations 13
   6.2. Design of interaction devices 14
   6.3. Inference methods 15
7. Personas 16
   7.1. Personas Foundations 16
   7.2. Persona Example 17
   7.3. Persona Workshop 19
8. Essential Use Cases 20
   Excursus: User Scenarios & Use Cases Scenario 21
9. HCI Design Patterns 21
10. Relationship to DOGANA 22
   10.1. User Centric Development Process 22
      10.1.1. Agile Development 22
   10.2. Usability Principles 22
   10.3. User Interface Design Guidelines 22
   10.4. Essential Use Cases 22
   10.5. Personas 23
   10.6. HCI Design Patterns 23
11. References 24

List of figures

Figure 1 UCD illustrated 10
Figure 2 Illustration of the example persona Inga 17

List of Tables

Table 1 EUC ATM example “Withdraw cash from an ATM” 20

Definitions and acronyms

API	Application Programming Interface
ASE	Automated Social Engineering
CC	CyberConnector
DOW	Description of Work
EUC	Essential Use Case
HAV	Human Attack Vector
IG	Information Gathering
IGAS	Information Gathering and Analysis Services
N/A	Not Available
OSINT	Open Source INTelligence
SDVA	Social-Driven Vulnerability Analysis
SMS	Short Message Service
SE	Social Engineering
SN	Social Network
SQL	Structured Query Language
SW	Software
TAHP	Tools for the Attack and Hook Preparation
TEAT	Tools for the Execution of the Attack
TIAR	Tools for the Information Aggregation and Reporting
UC	Use Case
UCD	User Centric Design Process
UI	User Interface
VCS	Victim Communication Stack
XSS	Cross-Site Scripting

1. Executive Summary

This document yields as a guideline for developers within DOGANA. It guides developers through design-decision processes and supports them finding the correct UI design for specific tasks. It gives a short introduction to the HCI building blocks within DOGANA. We line out the specific building blocks of user-centric design within this deliverable and relate them to their practical application in DOGANA.

NOTE: There will be an update to this deliverable in M36 of the project. The update will contain the essential use-cases and the Personas. It will be numbered D4.3b.

2. Introduction

This deliverable is a guide for developers to help implementing usable and end-user centric solutions. The goal is to support the developers and free them from taking design decisions, which affect the final user experience. AIT defines the work done in T4.3 as an HCI service to the developing partners. Seen from this position this deliverable D4.3 can be seen as an HCI service catalogue for the developers. The title of the document "DOGANA Interaction Design Guidelines" means that this document provides guidelines, which should be helpful when designing the interaction flows in DOGANA. This is distinct to the "User Interface Design Guidelines", which form a set of concrete rules and are introduced in chapter 6.

2.1. Deliverable D4.3b

This deliverable will undergo an update. Whereas this version provides the methodological building-blocks for the user-centric activities in DOGANA the update (D4.3b) will hold the results of the methods applied. The updated deliverable D4.3b will be presented in month 36 of the DOGANA project.

The updates will contain:

1. The DOGANA personas (see chapter “Personas”)
2. The DOGANA essential use cases (see chapter “Essential Use Cases”)

3. GUIs in DOGANA

The DOGANA framework will include third-party tools in its toolchain. Within the DOGANA projects graphical user interfaces of third-party tools will be used in different ways. The differences lie in the degree of integration into the DOGANA toolchain:

1. DOGANA GUI: this is the part of the DOGANA framework where tools are integrated, which allow complete and deep integration. This deep integration allows the DOGANA GUI to handle third-party tools in the background. This allows DOGANA to provide a complete new GUI to a third-party tool.
2. Import GUI: some tools, which will be integrated into the framework will not allow deep integration. For those tools the DOGANA GUI part is to provide a GUI for according import-functionality.
3. Form GUI: some tools do not even provide export-functionality but only visual output. For those a form will be provided

We know that not all GUIs of the third-party tools perfectly satisfy usability and user experience goals. Still DOGANA will strive to at least allow the DOGANA GUIs to be as user-centric as possible. The following chapters explain the HCI methods available within DOGANA to achieve good and usable UIs.

4. User Centric Development Process

What was formerly known as ISO13407 and is now ISO 9241-210 is the basic guideline for the process of developing usable software. User-centred Development (UCD) (a.k.a. User-driven development) describe design a process where end-users impact the design of the final product. The term was created by Donald Roman's research laboratory at the University of California San Diego (UCSD) in the 1980s becoming now widely adopted in a wide variety of domains.

There is a full set of benefits in adopting UCD; the key point is that ad the end the result is more likely to be useful, usable and meaningful to end users.

In software design users are involved during all the product cycle with a multitude of methods (e.g. prototype and usability testing…) but all of them must following 6 principles:

1. The design is deeply based on the understanding of users, tasks and environment.
2. User are involved through all the production cycle.
3. The design is driven and refined by user evaluation.
4. The development follows a cyclic approach.
5. The design addresses the entire user experience.
6. The work is done by a multidisciplinary team.

In the reaming part of this section we will put strong emphases on UI in software development that are the target of the DOGANA platform.

UCD incorporates the principles of the **Lean start-up** [1] with **Agile principles** [2] as well. Lean start-up methods proposed by Eric Ries are a set of production practices uses to apply the so called lean manufacturing; a philosophy on the production process that make the whole process "lightweight".

According to Lean methodology the first action to be performed when a new product is needed is to create a so called Minimum Viable Product (MVP that is the first version of the product) as fast as possible and then it has to be shown to the end-users to gain feedbacks as soon as possible. This helps confirming or not the assumptions developers had made at the beginning and create a strong foundation for the remaining part of the development. Continuous deployment is another key element; the previously mentioned process is repeated as many time as needed until the development is fully completed. Alpha and Beta testing phases are strongly encouraged when the envisaged product has a considerable size.

The **Agile Development** described in the Agile Manifest has as main focus to change the traditional software development model (i.e. waterfall model) into a less structured but more dynamic approach that include frequent deliveries to gain fast and punctual feedback from users.

Commonalities between UCD and Agile: Both Agile and UCD are development and design philosophies more than a mere set of tools. Agile comprises a wide variety of techniques like Extreme Programming (XP), Scrum… which can be used in UCD (which we recommend to to).

Conflict between UCD and Agile: Agile development is predominantly led by developers while UCD is made by a heterogeneous team (Designers, Developers, User Experience professionals, even though in many industries the first two are the same person). Looking at the principles behind the two, the main difference is into the final target; UCD is targeted to interfaces while Agile to the entire software manufacturing process.

In conclusion both UCD and Agile must work together such that the final product has a good quality for both User Interface and Software design.

The overall UCD process is made by 4 main steps showed in the next figure, processes are iterative and it is allowed to move back and forth between stages if needed. The first step (the identification of the need for human centric design) is always number one. The next steps (specify context of use, specify requirements, produce design solutions, and evaluate designs should be done in that order, but one can decide, which step is the next one. Especially when developing new artefacts producing a design solution could be a valid second step after the need identification.

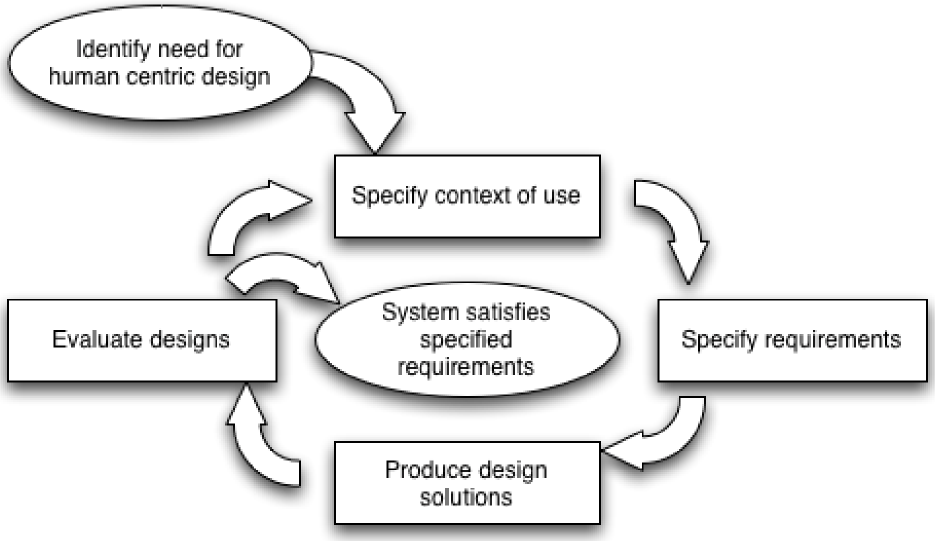

Figure 1 UCD illustrated

Here a short description of the UCD building blocks:

- **Identify need for human centric design:** a plan to the development shall be made and a team that must work on the UI should be selected. The team must be heterogeneous and at least one element must have knowledge of Usability.
- **Specify context of use & specify requirements:** Where the team must start performing researches on the end-users. There is a wide variety of techniques that can be selected like Personas (creation of the representation of a user via user interviews and personal data) or Task Analysis (learning about the goals including what the user want to do with the manufacture).[1]
- **Produce design solutions**: In this phases the design of the UI in the form of a prototype shall be implemented bases on the previous information. Several techniques can here still be used like Parallel Design where we must ask to different people to create a prototype starting from the same requirements; at the end the user will evaluate the results.
- **Evaluate designs:** at the end the UI prototype will be deeply tested. Usability test plays a key role there where usability principles shall be fully covered in the application as well as Alpha testing.

Chapter 10.1 explains the practical application of the UCD in DOGANA.

[1] Refer to: https://www.usability.gov/what-and-why/user-research.html for a full coverage of the available techniques.

5. Usability Principles

The international standard ISO9241-11 defines usability as:

> *The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.*

The usability principles provide basic human requirements and break down "effectiveness, efficiency and satisfaction" of the ISO definition into principles, which can be easier handled by humans. When those principles are met by a man-made artefact, the artefact is user-centric in a way that it meets human needs. The principles shown below are based on Nielsens Heuristics [3] and have been extended by AITs HCI experience. Heuristics are valid for every interactive system, hence should always be taken into account when developing such.

The 12 usability principles are:

1. **Consistency:** Consistency describes a common design of elements and processes from the users' point of view; All user interface concepts should thus be consistently designed
2. **Feedback:** Feedback means that users expect a sufficient system reaction to all of their actions
3. **Efficiency:** The user interface must enable the users to carry out their tasks efficiently
4. **Flexibility:** The system must allow different users to work differently, or a single user to work differently if she wishes or needs to, in order to accomplish goals
5. **Clearly marked exits:** The user must always know how she can leave a specific context, window or display when working with a user interface, and how she can return to her starting position
6. **Wording in users' language:** Wording in the user interface must be known and easily understandable to the user
7. **Task orientation:** A user interface shall always be designed to best possibly suit the users' tasks; Never shall a user need to adapt to a system
8. **Control:** The user must always be in control of the system; the user must never have the feeling of the system controlling her
9. **Recovery and forgiveness:** The system must prevent the user from (unknowingly) taking severe actions; The user shall be able to undo changes or actions easily
10. **Minimize memory load:** The user shall be enabled to focus totally on her task, not being troubled with the user interface as such; Therefore the user interface must require as little cognitive effort as possible
11. **Transparency:** The user must always know what will happen when she takes an action- the user interface must be transparent
12. **Aesthetics and emotional effect:** Everything has an emotional effect; If a user interface has an inappropriate emotional effect, it will interfere with the user's tasks

The practical application of the application of those principles is explained in chapter 10.2.

6. User Interface Design Guidelines

User interface design guidelines summarise knowledge of good UI design practice. They provide useful high and low level guidance on the design of usable man machine interfaces. Adherence to specific guidelines is part of the human requirements of DOGANA. Designers and developers should familiarise themselves with the relevant guidelines, and apply them during development.

All major operating systems have according design guidelines – here three examples:

1. Windows: Guidelines https://msdn.microsoft.com/en-us/library/windows/desktop/dn688964(v=vs.85).aspx
2. Apple: OS X Human Interface Guidelines https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/
3. Linux: due to the fact that there is not one single window manager on Linux different user interface design guidelines exist – one example are the KDE Human Interface Guidelines https://community.kde.org/KDE_Visual_Design_Group/HIG

The interaction between users and computer systems is a complex topic that is based on many different theories, ranging from cognitive models to theories of representation. The design of a system must be done in accordance to some general guidelines, which will be described in the following. The main categories that have been identified to show the design guidelines are: theories on visual representation, theories on the design of interaction devices and theories on inference methods. This chapter is structured in three sections, one for each of the aforementioned main topics.

6.1. Visual representations

The scope of a visual representation is to define conventions for making marks on a surface meaningful to the user. Thus, it is a key aspect that interaction designers should take into account. Most of the information presented in this Section has been found in [4] and [5]. The aim of this section is to present some common conventions that are used to represent information on computer displays.

Textual contents are represented according to well-known conventions and to the styles borrowed from newspapers, books, magazines, etc. Tabulated columns, indentation, borders are just some examples.

The representation of maps, graphs and schematic drawings follow conventions that apply to their not digital counterparts as well. For example, white spaces may be used to distinguish the core components of a graph, and labels help to explain them.

Also the representation of pictures relies on shared conventions. For example, the use of perspective may change according to the rendering that the picture is expected to provide. Icons and symbols are special types of pictures, since they are often used to deliver an abstract message. Concerning icons and symbols, it is hard to find widely-used conventions. The designer must always take into account the correspondence between the meaning and the icon used, in a way that is as much intuitive as possible.

In order to lower the barriers to user accessibility, the digital environment must be designed to be familiar to the user. Visual metaphors are meant to achieve this goal. An illustrative example is the computer desktop, in which the wastebasket is a clear metaphor of office trash can.

6.2. Design of interaction devices

This section illustrates the principles used in the design of interaction devices. Beside their particular implementation (e.g., type of hardware used), it is worth specifying the principles of the interaction of the user with the content: 1) How to get the content? 2) How to navigate around it? 3) How to manipulate it?

There are basically two methods for accessing a digital object: 1) by naming it or 2) by pointing at it. The first method is rather antique and inefficient. The second method is called “Deixis”, and it requires the use of a pointing device (e.g., a mouse). The essence of any method belonging the deixis paradigm is to relate user “gestures” with a media “location”. Deictic methods allow to easily get the content and also to navigate around it (often with the support of scroll bars, zooming, etc.…).

Pointing devices can be evaluated for their ability to optimize the efficiency of the interaction. Early Human Computer Interaction (HCI) models based this process according to the Fitts’ law [6]. This law relates the time needed to take to point at a given location with the size of the target and the current hand position. Most of the today interfaces are based on Graphical User Interfaces (GUI), which allow a direct manipulation of digital objects.

6.3. Inference methods

This section presents two main topics: 1) inference that users make on the behaviour of the computer systems and 2) inferences that computer systems make to predict the behaviour of users.

Concerning the first topic, theories have been developed to explain the possible mental models that users create to understand the functioning of a computer system. The interaction between user and computers can be divided in two main parts: evaluation and execution [7]. The former is about how the user knows what the system is doing, while the latter is about how the user knows what he needs to do in order to achieve its goals (i.e., what does he need to know about digital objects in order to obtain what he wants?).

During the evaluation phase, the user has to make inferences on what is happening inside the system. Thus, the designer should design the system in order to give sufficient clues to the user and facilitate the inference process. The scope is to help him form an adequate mental model. This scope can be achieved by using visual metaphors, i.e., the screen simulates some familiar real world object so that the user can understand the system by analogy to the real world (see the Visual Representation Section).

The use of metaphors and analogies helps in the execution phase as well. In fact, when the system behaves exactly as the real objects that are used to represent it, then the user knows what to do with them. In practice, it is very hard to find analogies that totally explain the behaviour of a system. Moreover, designers often create metaphors that correspond to their understanding of the system, which is far more complex that what normal users should know about it. Studies must be conducted to identify what the user needs to know.

Concerning the second topic, the way users create mental models is studied from different perspectives, ranging from studies on memory to theories of learning. Their aim is to understand how the user thinks in order to predict his behaviour and thus propose more suitable interfaces (e.g., presenting short-cuts and other kind of aids). Machine learning fits particularly well to predict user behaviour (e.g., Bayesian inference techniques).

How user interface design guidelines and the DOGANA project are used can be read in chapter 10.3.

7. Personas

7.1. Personas Foundations

First of all we want to point out that this chapter deals with Personas as a method for user centric-development. Within DOGANA a variant of the Personas method is part of SE 2.0. The application of the Personas method within SE 2.0 is described in chapter "Persona modelling" in the deliverable "D4.1 Human attack Vectors in SE 2.0".

Alan Cooper [8] - “inventor” of the personas method describes it like this:

> *“Personas are not real people, but they represent them throughout the design process. They are hypothetical archetypes of actual users. Although they are imaginary, they are defined with significant rigour and precision.”*

Or, how Lindgren [9] state it:

> *“...a hypothetical archetype of real users described in great detail and defined by their goals and needs, rather than just demographics.”*

In general personas show the scope and nature of the design problem. Until “the user” is precisely defined, we can always imagine that we ourselves are the users. We have different pictures in mind which hinder fluent communication [10]. Personas as communication tool allow defining which users we are building for and synchronize the pictures in our minds.

A lot of companies (Microsoft [11], Ford, Chrysler, Sovereign Bank, Amazon, Best Buy, Staples, FedEx, UPS, IBM, SAP, SONY, Razorfish, Pfaltzgraff, Yahoo! Media, Electrolux) use personas successfully because of their advantages.

The advantages of the Personas method are:

1. Support having the same picture of our end-users in mind for everybody in the project; hence reduce communication-complexity which makes communication easier and more fluent (this again saves time explaining “the user” every time he appears in a communication process).
2. Bias the minds of everybody in the project toward user-centered thinking. This gives the otherwise “technical touch” of R&D projects a human touch and brings things to live in a natural way (as human minds deal great with other persons - but human minds have a hard time when dealing with abstract big bunches of data) - personas make use of the “Emotional Mind” of people.
3. Be an evaluation tool as walkthroughs can be conducted with personas (which is very handy to judge design alternatives).
4. Leave the world of possibility thinking as you are very unlikely to fall back to “the user”.
5. They can shorten feature debates which saves time.

6. A tool to help the whole project team focus on the needs of our target users instead of using a different ad hoc “the user” definition which comes to mind at a point in time (Humans have just one locus of attention. It lies in the nature of R&D that the developers’ locus of attention is focused on the technical issues and not the real end-users. Therefore there is a need for methods solving this missing focus on the end-user. Personas are one way to do it.).
7. Personas are unobtrusive - they do not modify any existing processes and unfold their power subtly in the minds of people; they make thinking about “the user” more convenient.
8. Unlike bunches of data personas support informed design
9. According to Cooper the design process becomes “enlightened”.

7.2. Persona Example

This example Persona was taken from the PrimeLife project[2]. The PrimeLife project was about privacy, hence the focus on private data in the example persona. Furthermore, within the PrimeLife project a decision was made to use illustrations instead of photos for the personas. For DOGANA we will use photos of real-word users.

Example Persona: Inga Vainstein

Figure 2 Illustration of the example persona Inga

Description

Inga is 46 years old and is currently working as journalist. As a part of her job she is travelling to various countries. She is very anxious to keep her public profile clean in order to have a good reputation. Inga likes to take pictures of every city she has been to. A few years ago she bought professional photo equipment.

[2] PrimeLife http://www.primelife.eu , last visited: 1.9.2016

Social Situation

Inga was married once but got divorced 8 years ago. Since then she is single and likes dating.

Health Situation

With her growing age, travelling and the stress in connection with the journeys start to become a problem for her.

Financial Situation

Inga is free of debts and her salary is situated in the higher middle-class.

Technology Usage

Inga is using mobile phones since 12 years (she regularly switches provider and phone). She uses her phone mostly to make calls and write messages. Furthermore she likes having her electronic calendar always with her. She has her own laptop for writing her articles. Inga uses common office and e-mail tools on a daily basis. Since she is a fan of photography and recently started to provide the pictures for her articles, she knows some photo editing tools. Although Inga is using computers regularly for work and leisure purposes, she is still insecure and afraid of problems that might occur.

Web Usage

Inga uses the internet on a daily basis for private as well as professional purposes.
What she does online:

- Photo- blogging of her trips
- Shopping (clothes & camera equipment)
- Audio and video conferences
- Community networking (social networks, online forums)
- Dating-services
- Collaborative Workspace

Inga uses mobile internet and diverse wireless connections when she is on a business trip. At home she has a broadband connection.

Negative Experiences

Friends posted party pictures of her being drunk at a costume party and linked them to her, which resulted in a bad reputation at a job interview.
Sometimes Inga can’t cope with the amount of SPAM mails she gets. Once Inga lost a USB stick with important data (articles she was working on) during a trip.

Wish List

Inga hates it when she has the feeling that the system controls her. She wants to decide who is able to see and use her data. She frequently googles herself to keep track of what data is publicly available about her. Inga doesn’t want that private information such as her dating profile becomes public. Therefore she uses a pseudonym on such sites.
Since her reputation is important to her, Inga does not want her colleagues to find out that she orders her clothes in mail-order companies.

Generally Inga wants her (private and professional) data to be safe and separated. She does not have a lot of time to spend on security. Inga has tried to use security software to keep her data safe, but it was too complicated for her. At the beginning she was reading almost every popup message. After a while she got frustrated because she did not understand the information provided and since then most of the time she just clicks okay to make the pop-ups disappear.

Additional Information
Inga likes to use free wireless hotspots at airports which are not encrypted most of the time. She is afraid that somebody might fetch her passwords, still being online is more important than these concerns. Inga does some of her shopping online. When she is looking for special items, she has to go to the shops of smaller companies. She is afraid of fraud and therefore does not want to give away her credit card information. Together with some colleagues Inga is working as a freelancer on different projects, some of them are also involving photography and art design. They exchange their ideas and drafts on a collaborative workspace. Therefore every member of this community has access to the current version of the project.

7.3. Persona Workshop

AIT uses the “Research Personas Method”. Therefor there is a need for a Persona Workshop, which consists of 3 major activities:

1. Workshop participants write down goals, activities, actions or problems of a fictive person on sticky-notes. This step is conducted by each participant on his or her own.
2. Workshop participants fill sticky nots of another colour than the first ones with data coming from studies, articles, publications and so forth
3. In a final step the workshop participants collaboratively apply affinity-diagramming to the created sticky-notes, cluster the, and naming them.

The result of the workshop are clustered staples of sticky-notes, which yield as the basis for the next steps in the Persona process – the creation of the Personas and the Personas materials. The DOGANA usage plan for Personas can be found in chapter 10.5.

8. Essential Use Cases

According to Constantine and Lockwood [12] an essential use case (EUC) is a structured narrative, expressed in the language of the application domain and of users, compromising a simplified, generalized, abstract, technology-free and implementation-independent description of one task or interaction that is complete, meaningful, and well-defined from the point of view of users in some roles or roles in relation to a system and that embodies the purpose or intentions underlying the interaction.

Conventional use cases describe, in narrative form, interactions that are complete, well defined and meaningful to some user. Within object-oriented software engineering use cases can describe interactions with other systems and equipment in addition to interactions with human users.

Characteristics of EUC compared to conventional use cases:

- Conventional use cases typically contain too many built-in, premature assumptions, often hidden or implicit, about the form of the user interface to be assigned
- EUCs are based on the purposes or intentions that might be carried out
- EUCs are often dramatically shorter and simpler than concrete use cases for the same interactions
- Scenarios, concrete use cases, and essential use cases are related task models representing successive levels of greater abstraction and generalization that takes us closer to the user needs and intentions

Table 1 EUC ATM example "Withdraw cash from an ATM"

User Intention	System Responsibility
Identify self	
	Verify identity
	Offer choices
Choose	
	Dispense cash
Take cash	

Table 1 EUC ATM example shows the classic ATM EUC example. It highlights how general EUCs are. On the left side the goal and/or intention of the user in a very generic way. The same is true for the right side, where the system responsibility is modelled. The example also shows the idea that EUCs are technology agnostic: if the ATM comes as graphical terminal or as an interactive voice system is not determined on this level.

Excursus: User Scenarios & Use Cases Scenario

In order to avoid confusion after explaining what a EUC is we draw clear borders to other terms. User scenarios are narrative descriptions of an activity, taking form of a story within a given context. Scenarios are scripts for work or interaction and are a characteristically rich and realistic. User scenarios are often used to talk about and describe general patterns through specific examples, but separating the specific from the general and keeping track of these can be daunting [13]. Whereas a use cases scenario is a single way through a (UML-modelled) use-case.

How EUCs relate to DOGANA can be seen in chapter 10.4.

9. HCI Design Patterns

A design pattern is a formal way of documenting successful solutions to problems. The idea was introduced by the architect Christopher Alexander [14] and has been adapted for various other disciplines. Alexander describes patterns as follows:

> *Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution.*

The pattern approach was adopted to HCI design [15], [16] to be able to build better interfaces by:

- Capturing the collective knowledge of the project in a way that can be immediately used
- Giving common language for inner- and outer project communication
- Constraining designers to work within the given boundaries to ensure user-centered outcome
- Better being able to keep focused on the end-user's experience

Some of the most sophisticated pattern libraries are:

1. A pattern library for interaction design by Welie: http://www.welie.com/
2. Yahoo Design Pattern Library: https://developer.yahoo.com/ypatterns/
3. Jenifer Tidwell: Designing Interfaces (book): http://designinginterfaces.com/

The relationship of HCI design patterns and DOGANA are explained in 0.

10. Relationship to DOGANA

Within this chapter we relate the building blocks of a user-centric design process to the DOGANA project.

10.1. User Centric Development Process

DOGANA applies a user centric development process from the first idea on: already during the proposal writing phase the consortium agreed to have one focus on the end-user. The inclusion of user-partners is the practical first step of the DOGANA UCD. The next steps are to apply the methods outlined in this deliverable within the DOGANA project. The next method applied is the Personas method (see chapter 10.5).

10.1.1. Agile Development

Somehow related to UCD is the development approach, even when UCD is not bound to any development approach, neither classic nor agile. Nevertheless most partners strive to have at least some “agility” in their development. Due to the different nature of the development teams participating in DOGANA there is nothing like a single DOGANA development approach as the partners are free to choose the approach they prefer.

10.2. Usability Principles

Usability principles should be applied when developing interactive systems. As the application of the usability principles needs experienced HCI experts one cannot expect developers to have this knowledge, hence HCI experts from AIT will apply the principles when evaluating software artefacts within the DOGANA project.

10.3. User Interface Design Guidelines

Within DOGANA developers should make themselves familiar the design guidelines for the operating systems they are developing for. For DOGANA developers, who are not that experienced in professional GUI development, the HCI experts from AIT will be available when questions come up.

10.4. Essential Use Cases

The original intent of this deliverable was to provide EUCs. At the time writing (August/September 2016) the final use-cases are still in review and not final. As the use-cases need to be stable to derive EUCs from them this was not possible until the finalization of this document. Therefore the EUCs will be presented in D4.3b (see also chapter “Deliverable D4.3b”).

10.5. Personas

During the DOGANA Meeting in Copenhagen from the 11th to the13th of July 2016 the consortium decided that we will create DOGANA personas. This will be done in a workshop. The planned workshop in Brussels will be held on:

29th of September 2016

The workshop is a 3-6 hours workshop within the consortium where consortium members are guided through the workshop by the workshop organizers (AIT). After the workshop AIT will create the personas and make them available to the consortium members. The finished Personas will be presented in D4.3b (see also chapter "Deliverable D4.3b").

10.6. HCI Design Patterns

The usage of patterns in DOGANA is twofold. ON the one hand patterns are defined as a means to capture long term experience knowledge. DOGANA cannot provide UI patterns at this state (September 2016) as no UI solutions have been finalized by now. This leaves DOGANA to the usage of already existing patterns. On the other hand during the project patterns might emerge (especially after the field-trial). After the DOGANA field-trials we will see if specific patterns have emerged. If we find design patterns AIT will presented them in D4.3b (see also chapter "Deliverable D4.3b").

11. References

[1] Eisenmann, Thomas R., Eric Ries, and Sarah Dillard. "Hypothesis-driven entrepreneurship: The lean startup." Harvard Business School Entrepreneurial Management Case 812-095 (2012).

[2] Beck, Kent, et al. "Manifesto for agile software development." (2001).

[3] Nielsen, Jakob. Usability engineering. Elsevier, 1994.

[4] BEDERSON, Benjamin B.; SHNEIDERMAN, Ben. The craft of information visualization: readings and reflections. Morgan Kaufmann, 2003.

[5] BERTIN, J. Semiologie Graphique, Paris: Editions gauthier-Villars.BertinSémiologie Graphique1967, 1967.

[6] WHISENAND, Thomas G.; EMURIAN, Henry H. Effects of angle of approach on cursor movement with a mouse: Consideration of Fitt's law. Computers in Human Behavior, 1996, 12.3: 481-495.

[7] ROGERS, Yvonne, et al. Interaction design: beyond human-computer interaction. netWorker: The Craft of Network Computing, 2007, 11.4: 34.

[8] Alan Cooper. 1999. The Inmates are Running the Asylum. Macmillan Publishing Co., Inc., Indianapolis, IN, USA.

[9] Lindgren, A., Chen, F., Amdahl, P., & Chaikiat, P. (2007). Using Personas and Scenarios as an Interface Design. Tool for Advanced Driver Assistance Systems. Paper presented at the Universal Access in HCI, Part II, HCII 2007

[10] John Pruitt and Tamara Adlin. 2005. The Persona Lifecycle: Keeping People in Mind Throughout Product Design. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[11] John Pruitt and Jonathan Grudin. 2003. Personas: practice and theory. In Proceedings of the 2003 conference on Designing for user experiences (DUX '03). ACM, New York, NY, USA, 1-15. DOI=http://dx.doi.org/10.1145/997078.997089

[12] Larry L. Constantine and Lucy A. D. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. ACM Press/Addison-Wesley Publ. Co., New York, NY, USA, 1999.

[13] Cockburn, A., Writing Effective Use Cases (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA 2000.

[14] Alexander, C., The Timeless Way of Building. Oxford University Press: New York 1979.

[15] Borchers, Jan O., A pattern approach to interaction design, in: DIS '00: Proceedings of the 3rd conference on Designing interactive systems, New York City, New York, United States, pages 369-378, ACM, 2000.

[16] J. Tidwell. Interaction design patterns. PLoP'98 Conference on Pattern Languages of Programming, Illinois, extended version at http://www.mit.edu/~jtidwell/interaction_patterns.html, 1998.