TOWARDS SMART ZERO CO_2 CITIES ACROSS EUROPE

VITORIA-GASTEIZ + TARTU + SØNDERBORG

Deliverable 6.5: Designing guide and tool catalogue

WP6, Task 6.5

Date of document

31/10/2017 (M 21)

Deliverable Version:	D6.5, V1.0
Dissemination Level:	PU[1]
Author(s):	Jose Luis Izkara/TEC, Daniel Reguera/MGEP-MU, Ganix Lasa/MGEP-MU, Mauricia Benedito/GIS

[1] PU = Public

PP = Restricted to other programme participants (including the Commission Services)

RE = Restricted to a group specified by the consortium (including the Commission Services)

CO = Confidential, only for members of the consortium (including the Commission Services)

This project has received funding from the European Union' s Horizon 2020 research and innovation programme under grant agreement No 691883

Document History

Project Acronym	SmartEnCity
Project Title	Towards Smart Zero CO2 Cities across Europe
Project Coordinator	Francisco Rodriguez // Tecnalia francisco.rodriguez@tecnalia.com
Project Duration	1st February 2016 - 31st July 2021 (66 months)

Deliverable No.	D6.5 Designing guide and tool catalogue	
Diss. Level	Public	
Deliverable Lead	*MGEP-MU*	
Status	X	Working
		Verified by other WPs
		Final version
Due date of deliverable	31/10/2017	
Actual submission date	31/10/2017	
Work Package	WP 6 - City Information Open Platform (CIOP)	
WP Lead	*ET*	
Contributing beneficiary(ies)	*TEC, AVG, MON, ACC, CAR, GIS, ZERO, TAR, ET*	

Date	Version	Person/Partner	Comments
03/06/2017	V00	Ganix Lasa/MGEP-MU	Creation of table of contents (ToC) and main guidelines for each section
14/07/2017	V0.1	Ganix Lasa/MGEP-MU	Main sections filled up.
30/07/2017	V0.2	Ganix Lasa/MGEP-MU	Corrections.
07/09/2017	V0.3	Ganix Lasa/MGEP-MU	Corrections.
14/09/2017	V0.4	Ganix Lasa/MGEP-MU	Corrections.
21/09/2017	V0.5	Ganix Lasa/MGEP-MU	Corrections.
28/09/2017	V0.6	Ganix Lasa/MGEP-MU	Corrections.
06/10/2017	V0.7	Ganix Lasa/MGEP-MU	Corrections.
07/10/2017	V0.8	Daniel Reguera/MGEP-MU	Corrections.
08/10/2017	V0.9	Daniel Reguera/MGEP-MU	Document delivered for review.

30/10/2017	V 1.0	Daniel Reguera/MGEP-MU	Final reviewed version delivered to coordinator.

Copyright notice

Table of content:

Table of Tables:

Table of Figures:

Abbreviations and Acronyms

Abbreviation/Acronym	Description
AEC	Architecture, Engineering and Construction
ASD	Agile Software Design
CEN	European Committee for Standardization
CIOP	City Information Open Platform
ESCO	Energy Savings Company
GIS	Geographic Information Systems
ICT	Information and Communication Technologies
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
LH	Lighthouse City
M2M	Machine to Machine
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OS	Operating System
POC	Proof Of Concept
REST	Representational State Transfer
SmartEnCity	Towards Smart Zero CO2 Cities across Europe
SDK	Software Development Kit
SQL	Structured Query Language
UCD	User Centred Design
WP	Work Package
API	Application Programming Interface
KPI	Key Performance Indicator
EU	European Union
XML	

Table 1: Abbreviations and Acronyms

0 Publishable Summary

Smarten City's vision is to create Smart Zero Carbon Cities that are more sustainable and inclusive, improve citizens' quality of life, create jobs and wealth, and offer equal growth opportunities. Thus, the project aims to develop a systemic approach for transforming European cities into sustainable, smart and resource-efficient urban environments in Europe.

In this line, the main objective of this deliverable is to collect both, a new specific designing guide that guarantees the participation of the citizen during the whole process, and a tool catalogue highlighting the more appropriate ones for the project. The created design guide and selected tools enable to define the basis for project future tasks development.

On the one hand, the new designing guide is based on the analysis of different UCD methodologies considering Smart Cities and ICT field development frameworks. On the other hand, the tool catalogue is based on a deep analysis of existing technologies for solution development. Moreover, it has been defined a specific selection criteria which is necessary to identify and select the most appropriate technologies for SmartEnCity project.

The definition of this basis pursues reaching the main goals that are defined for the project. In short, a citizen centered design approach helps creating more sustainable and inclusive cities and increasing the quality of life of its inhabitants

The approach followed in Task 6.5 to analyze the different UCD methodologies considering Smart Cities is based on a process defined by Bhrel et al. (2015). In which the phases are defined in 4 stages (widely explained in section 2.1):

- Stage 1: Automated search in data bases
- Stage 2: Exclude publications
- Stage 3: Obtain primary papers
- Stage 4: Quality assessment

1 Introduction

1.1 Purpose and target group

This report constitutes the Deliverable “D6.5 - Designing guide and tool catalogue”, the main outcome of the task “T6.5-HMI Development Mechanisms” within the work-package 6.

The main objective of this document is to create a new specific designing guide and a tool catalogue, in order to define the basis for project future tasks development. The new designing guide is based on the analysis of different UCD methodologies considering Smart Cities and ICT field development frameworks. The society is getting closer to a new era, where the digital presence will be higher than ever before. The Smart Cities of the future must be citizen centered, and the technologies that will create the technological ecosystem must enable an easygoing and pleasurable lifestyle within such a complex reality. That is way, the new designing guide for this project is focused on citizens.

On this basis, a specific tool catalogue is created based on a deep analysis of existing technologies for solution development. A specific selection criterion is defined in order to identify and select the most appropriate technologies for SmartEnCity project. Being a citizen centered approach, the criteria defined for technology selection is mainly focus on user related factors, such as, ease of access/use, technical skills needed, cost, learning curve etc. Thus, a new tool catalogue is provided for the project that will enable to build more friendly and pleasurable Smart Cities, based on these technologies.

The main activities carried out within this task are listed below:

- Phase 1: Analysis of different UCD methodologies considering Smart Cities.
- Phase 2: Create a designing guide for the project (UCD based).
- Phase 3: Contribute on the analysis of the technologies and tools offered by the frameworks selected in Task 6.2 for solution development
- Phase 4: Participate on the selection of the technologies and tools.
- Phase 5: Contribute to the writing of the deliverable (Deliverable 6.5)
- Phase 6: Deliverable review.

This report is structured in the following main sections:

1. Introduction
2. State of art of existing Frameworks and standards for Smart Cities
3. New designing guide for SmartEnCity
4. Analysis of the technologies and tools for solution development
5. Selection of technologies and tools
6. Conclusions, deviations and outputs for other WPs

The target audience of the defined new designing guide and the state of art would be project manager and Smart City related solutions creators. And, the target group for the specific technology catalogue would be solution developers.

Main target group of the information and conclusions collected in this deliverable are lighthouses to be implemented in the project

1.2 Contributions of partners

The following Table 2 describes the main contributions from participant partners in the development of this deliverable.

Participant short name	Contributions
MON	Task Leader. Responsible of the content of the deliverable. Main contributor of Section 1 (Introduction), 2 (State of the Art of existing Frameworks and standards for Smart Cities) and Section 6 (Conclusions, deviations and outputs). Contributor with different inputs to the other sections.
TEC	Main contribution of the Section 4 (Analysis of the technologies and tools for solution development). Contributor with different inputs to others sections of the deliverable
GIS	Main contributor of Section 5 (Selection of technologies and tools). Contributor with different inputs to others sections of the deliverable
TELIA	Reviewer the deliverable
ETIC	Reviewer the deliverable

Table 2: Contribution of partners

1.3 Relation to other activities in the project

The following Table 3 depicts the main relationship of this deliverable to other activities (or deliverables) developed within the SmartEnCity project and that should be considered along with this document for further understanding of its contents.

Deliverable Number	Contributions
D6.1	This deliverable provides the requirements identified for SmartEnCity
D6.2	This demonstrator presents the Reference Architecture for SmartEnCity
D6.3	This deliverable presents the Data Model architecture implementation
D6.4	This deliverable presents the Interoperability mechanisms implementation

Table 3: Relation to other activities in the project

2 State of the Art of existing Design Methodologies for Smart Cities

The objective of this section is to analyze existing frameworks and architectures on Internet of Things (IoT), city modelling, different vertical domains relevant for the project (e.g. energy, mobility or citizen engagement) and hardware and software architectures for the deployment of ICT solutions for Smart Cities.

Since the arrival of the digital systems, the computer, and the IoT in particular, the development of the software of such systems have been a topic of great concern and importance. From the time of its creation the ideologies or perspectives of its development have been changing, suffering an evolution caused significantly by the changes of the social and economic paradigm. That way there is a great amount of methodologies that pose their methods and approaches of how to develop software applications that ended as ICT solutions for Smart Cities.

Starting from this context, this section presents a review of the methodologies that have been developed during the last years. As a result, this review shows a chronological analysis of the methodologies, comparing them by their perspective, the approach, the phase of the process that they are aiming and their application for real cases.

2.1 Review of Existing design methodologies

This subsection provides a review on existing design methodologies for the deployment of ICT solutions for Smart Cities and similar. This analysis considers commercial as well as open and free alternatives and covers both hardware and software aspects.

The revision of methodologies for its subsequent comparative analysis is mainly in search of methodologies referring in recent years in the field of ICT solutions and software development. The criterion on which the selection was based has followed a process defined by Bhrel et al. (2015) in which the phases are defined in 4 stages (Fig. 1).

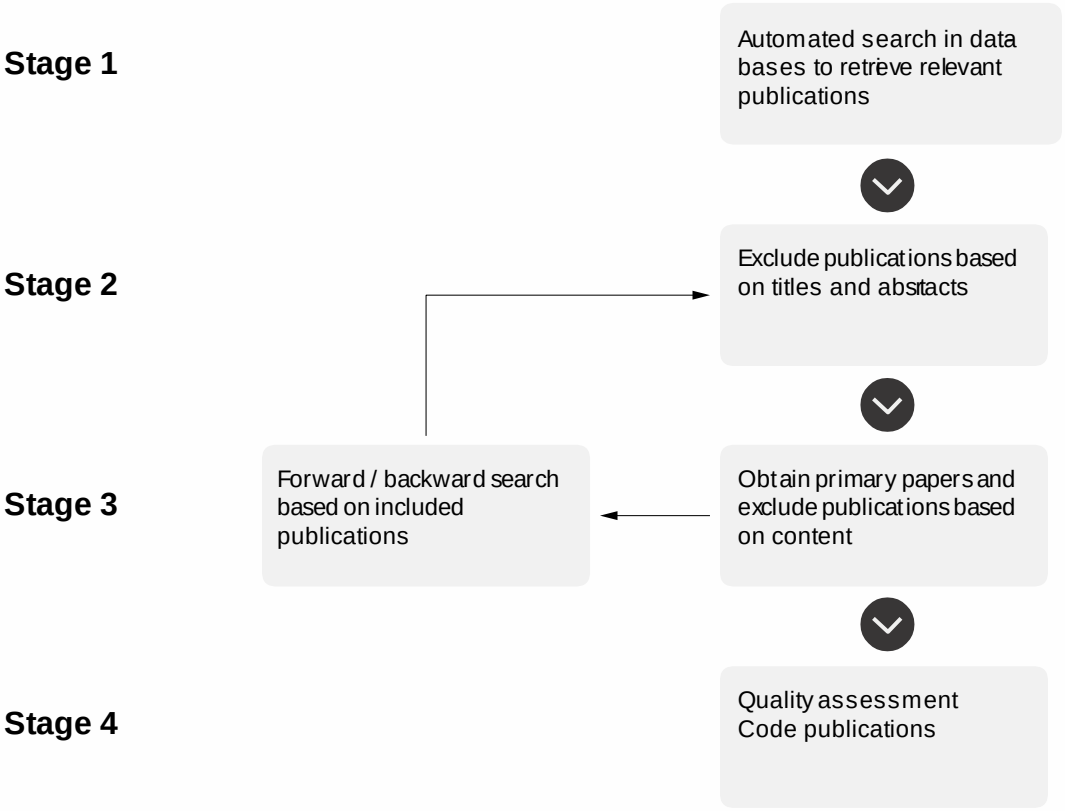

Figure 1:Selection methodology and process by Bhrel et al 2015

In the first phase, a search was made in the Google Scholar database (2017), using the search terms “ICT solutions”, “smart city methodology”, "software development methods" and similar. In the second phase, based on the titles and abstracts, possible publications of interest have been identified and those considered as not relevant have been excluded so that in a third phase, the content has been analyzed. In this stage, it was decided whether the content was suitable for inclusion in the review and if it was relevant in the area of ICT and software development. In the case of not considering it suitable for inclusion, it has moved back to the previous step. Finally, the review was carried out with the selected methodologies, which is shown in the comparative list.

The selected methodologies with a brief summary of their contents are the following ones:

- **Jackson System Development (JSD) method**. (M. A. Jackson, 1982)

The JSD methodology is based on a sequential and non-iterative process, defined with fixed steps in a tree structure. It is based on the following steps: entity and action analysis, entity structure analysis, initial model, system functions, system moments and implementation.

- **Structured Systems Analysis and Design Method (SSADM).** (Ashworth, C. M. 1988.)

It is a waterfall methodology based on a sequential (non-iterative) process, in which the following phases are not advanced before the previous phases are completed. It is divided into the following phases: feasibility analysis, environmental research, company system analysis, requirements definition, technical system options, logic design and physical design.

- **Agile software development: the business of innovation.** (Highsmith, J. Cockburn, A. 2001)

It is based on providing a dynamic process characterized by iterative cycles and timely participation of external agents, so that the Minimum Viable Product (MVP) is designed as soon as possible and is improved by continuous evaluations.

- **Key principles for user-centered systems design**. (Gulliksen, J. 2003).

User-Centered Systems Design (UCSD) is a process focused on usability throughout the entire development process and throughout the system life cycle. It is based on the following principles: focused on user needs and actions, active user participation, iterative and incremental development of the system, prototypes from the initial stages, evaluate use in the real context and a multidisciplinary team with a holistic view of process.

- **Usability engineering methods for software developers**. (Holzinger, A. 2005.)

The method focuses on working the usability of the product throughout the process. It defines usability as the usability and acceptability of the system for certain users carrying out specific tasks in a specific environment, focusing on both product quality and user satisfaction. In addition, it proposes different usability techniques for each phase of the project.

- **Towards a Framework for Integrating Agile Development and User-Centered Design**.

(Chamberlain, S.; Sharp, H. 2006.)

It is a theoretical framework where it proposes the union of the methodologies Agile Software Development (ASD) and User-Centered Design (UCD). After analyzing its similarities and differences, it defines the following principles as a basis for an adequate union: integrate the user into the development process, close collaboration between designers and developers, create prototypes as soon as possible to make ideas tangible, enhance exploration and manage the project in a cohesive way.

- **U-SCRUM: An agile methodology for promoting usability.** (Singh, M. 2008).

It proposes a methodology based on the integration of the usability and SCRUM principles. The so-called SCRUM methodology is based on agile development frameworks, with incremental and iterative processes. However, it does not pose the terms of usability, therefore, Singh proposes a methodology that unites the two parts and incorporates the concepts of usability throughout the process.

- **Aesthetics and experience-centered design**. (Wright, P.; Mccarthy, J. 2008).

A methodology focused on designing the user experience when interacting with the product. It proposes a theoretical framework where it emphasizes the initial phase of the process, where users are identified with their needs and motivations, so that user stories are designed with the experience they will have in their use.

- **User eXperience Design and Agile Development: From Theory to Practice**. (Silva, T. 2012.)

It proposes the union of the User Experience Design and Agile Software Development methodologies. To do this, it includes in the iterative agile process a previous step, called iteration 0, where the concepts of user experience and user-centered design are applied to

identify the needs of users. In addition to that, it presents different techniques for the integration of usability concepts throughout the process.

- **Context-aware systems**. (Fischer, G. 2012.)

It is a methodology that dictates as a transcendental factor the context that surrounds the system, and aims to offer a product that assists people to increase their knowledge, productivity and creativity. In this way, it is possible to provide a system that gives the correct information, at the right time, in the right place, in the right way and to the right person.

- **Agile Usability Patterns for UCD early stages**. (Betholdo, A. 2014.)

A methodology focused on the initial phases of the project, based on a process that combines the characteristics of ASD and UCD and introduces usability concepts during all phases. Define in particular three techniques for the initial phases: identify the needs of users using the tools of the UCD, specify and analyze the context of use of the system and define the technical requirements.

- **Metodologías ágiles centradas en personas para desarrollar software educativo**. (Gonzalez, C. 2015.)

It is a way of working that encompasses the methodologies of ASD, UCD, User Experience and Lean UX. It brings together concepts from all of them, so it raises a process based on "design thinking", where it proposes tools to analyze the user, the context and the actions, so that through an agile development, we can design a system that contributes a positive experience in the user.

- **Empowering user interfaces for industry 4.0**. (Pfeiffer, T. 2016).

It exposes a way of working focused on creating systems that, through concepts extracted from the UCD, ASD and usability, are able to empower and strengthen the user's capabilities to increase the control and productivity of ICT within the industry 4.0. It presents three concrete tools for this: People, Storyboards and the use of eye-tracking for observations of user behavior.

- **Creating people-aware IoT applications by combining design thinking and user-centered design methods.** (Goyal, S. 2016.)

A methodology based on the design thinking and user experience to create applications centered on the user. It claims that development is driven on the basis of user needs and not technical opportunities. This methodology encompasses all phases of the process and proposes tools for each of them. It defines the following phases: exploration, identification of requirements, analysis, design, prototype, evaluation and redefinition.

- **A conceptual UX-aware Model of Requirements**. (Kashfi, P. 2016.)

It exposes a way of working that unites the concepts of the user experience and the engineering of the software. It distinguishes the user experience from the concepts of usability only, since it also encompasses the emotional aspects of the users. It defines three main levels in the development process: user experience requirements (user's emotional needs), QR objectives (quality requirements) and FR objectives (functional requirements).

- **User experience methodology: from the physical to the emotional**. (Presley, O. 2016)

A design methodology to develop systems that create experiences. It proposes that the center of a software development process must be the person and works two main phases: generation and evaluation. In the first of them, it identifies the motivations of the user and their needs both functional and emotional. In the second, it presents an evaluation based on usability, but with a qualitative and non-quantitative character.

- **Spiral UX Design Model.** (Guo, H. 2016)

It is the adaptation of the user experience to the spiral model of software development. Defines an iterative process that begins with the identification of user needs. Then, for the next phases, it proposes a spiral model where each step of advance assumes an increase in the degree of fidelity of the concept, starting from design sketches to the final product.

2.2 Classification criteria

After the description of revised methodologies, the criteria used for the subsequent comparison are detailed on this section, which is divided in four elements: focus, perspective, process phase and application.

2.2.1 Focus

It refers to the way in which the author believes that a software development process should be addressed. Four different approaches have been divided:

- *Product*: focuses solely on the software itself, proposes a methodology to create more efficient and effective systems.
- *Product-user*: states that the process must start from the needs of the user, and that its compliance determines the success of the product. In addition to user-centered design concepts, it proposes software development techniques.
- *Product-user-context*: proposes a way of working in which it is necessary to contemplate the whole system environment, analyzing the product, the user who uses it and the context in which the interaction is performed.
- *Experience*: he argues that when developing new software, he must design his experience of use, fulfilling the motivations and functional and emotional needs of the users.

2.2.2 Perspective

The criterion of the perspective refers to the point of view of the team members that the author proposes, which is determined by the discipline (s) of the team members. Three perspectives have been differentiated:

- Software Development Process (SDP): the team composed of software developers.
- SDP-Design: multidisciplinary team composed of developers and designers.
- Design: the team formed by designers.

2.2.3 Process Phase

In this criterion have been defined the phases of the process in which the working method refers. The phases are defined based on the DBZ methodology of Mondragon Unibertsitatea (DBZ 2014):

- *Strategic*: identify opportunities for new products and services through the analysis of the internal and external context of the company.
- *Exploration*: based on the opportunity detected, define the specifications that the product / service has to consider meeting the needs of the client / user.
- *Design*: to generate new product / service concepts according to the specifications included in the Design Brief.
- *Development*: Design and develop in detail the concept of product / service selected to obtain a functional prototype.
- *Launch*: insert and promote the new product / service in the market and collect performance information for the identification of improvements.

2.2.4 Application

It tries to classify the methodologies based on the applicability of the same ones. Three applications have been defined:

- Framework: theoretical framework where it exposes concepts of a way of working.
- Method: proposes a concrete process with steps and phases to be performed to be able to apply the theoretical exposition.
- Tools: it presents concrete tools of detailed form for its direct application in real cases.

Thus, in this new classification, the mentioned methodologies have been gathered in a table to compare them with the defined criteria (table 1). In the vertical axis, the methodologies have been placed, chronologically in the order in which they are exposed in previous section 2.1. In the horizontal axis, the criteria have been located, in the same order defined in section 2.2.

Through the analysis of the methodologies based on the defined criteria, different conclusions have been obtained. On the one hand, it is observed that there is an evolution of software developments towards the user experience. At first, they focused solely on the development of the software itself, so the goal was to achieve an optimal product in functionality. However, over the years the idea of considering the user/citizen in the process was introduced, mainly identifying their needs with the product and designing them starting from them. As a consequence, many methodologies were created based on the combination of methods of software development, such as Agile Software Development (ASD) and design methods such as User Centered Design (UCD).

These user-centered methodologies have taken on great importance within the new digital era and today many companies have integrated it for their IoT developments. Despite this,

the digital products and services continue evolving and in recent years has begun to emerge the term user experience. It tries to design the experience that the user will have when using the product and is considered the space that generates the greatest link between the product and the human being, the place where human emotions are faced. Therefore, not only focuses on the pragmatic needs that the user may have, as in the UCD, but also focuses on the emotional needs and motivations that you want to meet in the use of the product, in order to achieve a system which provides positive emotions that increase people's involvement (Isen, 2001), facilitating learning processes (Kort et al., 2000). In addition, it takes into account everything that may influence the user experience, such as the context, the moment of use, the actions to be performed and the means that will be used to do so.

On the other hand, it can be observed that the design perspective is increasing popularity within the software development methodologies. This demonstrates that the IoT field and design are increasingly closer, so that multidisciplinary teams on projects are becoming more common.

Also, when analyzing the development phases of each methodology, it can be observed that the older ones, which focus only on the functionality of the product, speak mainly of the development phase. As far as user-centered methodologies and experience are concerned, most are focused only on the exploration phase, where they analyze the user and other factors that may influence later on to realize both the design and the development. So, there is a lack of methodologies that take into account the whole process.

Finally, the last criterion indicates that these methodologies have different applications, but it is observed that in the last years there are more methodologies focused on the user that show defined phases of work and tools that allow to realize what they propose.

2.3 Conclusions

After the review was carried out and as a main conclusion, a proposal has been made for the ideal characteristics that a methodology for the design and development of Smart City related IoT solutions should meet today. First, the methodology should focus on the user experience, since it works on aspects that favor the development of more active, critical, participative, motivated users with a greater affective bond with respect to what surrounds them. Secondly, it should visualize the project from a perspective of union between software and design, forming a multidisciplinary team that contributes expert opinions in both fields of knowledge and different and constructive points of view that avoid major redesigns and optimize the final duration of the project. Thirdly, it would be appropriate to consider all phases of the process, from an exploration phase of opportunities to the final phase of launch, in order to achieve a more coherent and complete product. Finally, it should be able to show its application for real cases, offering a structured guide that defines the method, the phases of the process and the steps to be taken together with the tools to be used in each one of them.

3 New designing guide for SmartEnCity

Being aware of the need to create a methodology that could be adapted to the advancement of technology, Highsmith and Cockburn (2001) summed up the challenges that the new methodologies should fulfill:

- Satisfactory to the client who approaches the mayor who complies with the original plans.
- Dealing with problems in the best way and reducing the costs of change during development.
- Make changes in previous phases to avoid the project to failure.
- Offering innovative, high quality software that meets the needs of the market and early

Starting from these challenges the Agile Software Design (ASD) was born. It focuses on providing a dynamic process characterized by iterative cycles and the timely involvement of external agents (Highsmith, 2001), so that the Minimum Viable Product (PMV) is designed as soon as possible and is improved by continuous evaluations.

However, the purpose of the ASD is to look for the functionality of the product, stating that customer satisfaction is met with a result that responds to its function, but does not focus on finding usability, and on meeting the needs of users (Blomkvist, 2005). The UCD, therefore, can offer a determinant contribution (Gulliksen et al., 2003), since it assures that the objectives and the necessities of the user are the main focus of the development and because it proposes a process of continuous evaluations of the user, and of iterations to redefine the concepts design and prototypes (Fox, Sillito and Maurer, 2008) . Therefore, as ISO 9241-210 (2010) says, it is a determining factor for the development of interactive systems (Fig. 2).

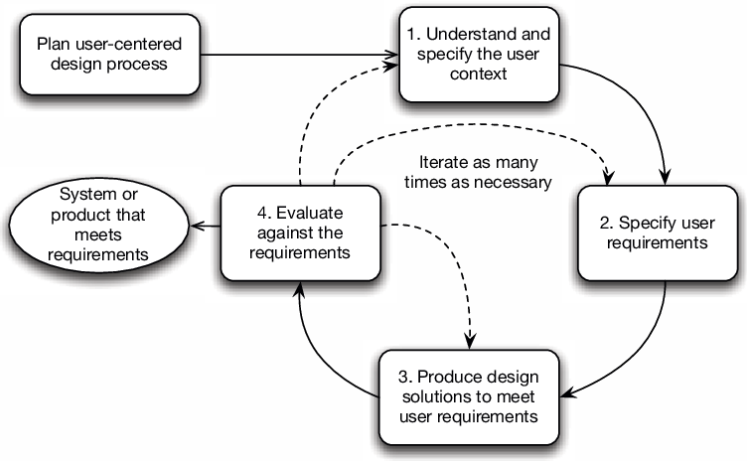

Figure 2: ISO 9241-210 User Centered Design

Given this situation, and within the SmartEnCity project, a new Smart Citizen Centered Digital Design (SCD) methodology has been created. It is an approximation of a structured designing guide based on these two approaches, defining in a more detailed way the phases, the actions and tools to be used and the way of implementation in each of them.

Also, unlike the analyzed works (Bhrel, 2015), SCD is aimed for ICT solutions in the context of Smart Cities. These characteristics make SCD unite a series of virtues that differentiate it from the current methodologies.

In addition, during the definition of this new design guide the philosophy of "Design Thinking" (Brown, 2008) has been taken into account, specifically the one proposed by the methodology of Innovation Design Center (DBZ) from Mondragon Unibertsitatea's (DBZ) (2015) (Fig.3).

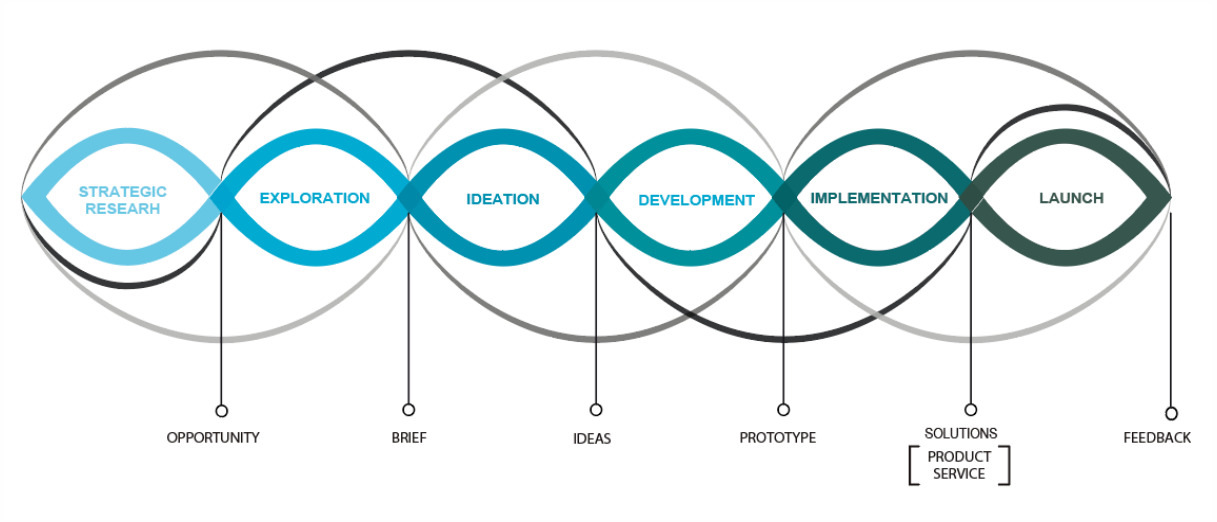

Figure 3. People Centered innovation methodology by DBZ of Mondragon Unibertsitatea.

Thus, the SCD methodology has been created. A structured procedure with the objective of serving as a design guide to develop ICT solutions within the Smart City context. It proposes the active participation of users during the whole design process, together with the multidisciplinary team formed by both designers and programmers, so that they can create solutions that meet the user needs.

3.1 SCD framework

SCD methodology is composed of three main phases: Understanding, Designing and Development (Fig. 4). The phases of the process, the actions that should be carried out and the proposed design tools that could be used are explained in next section.

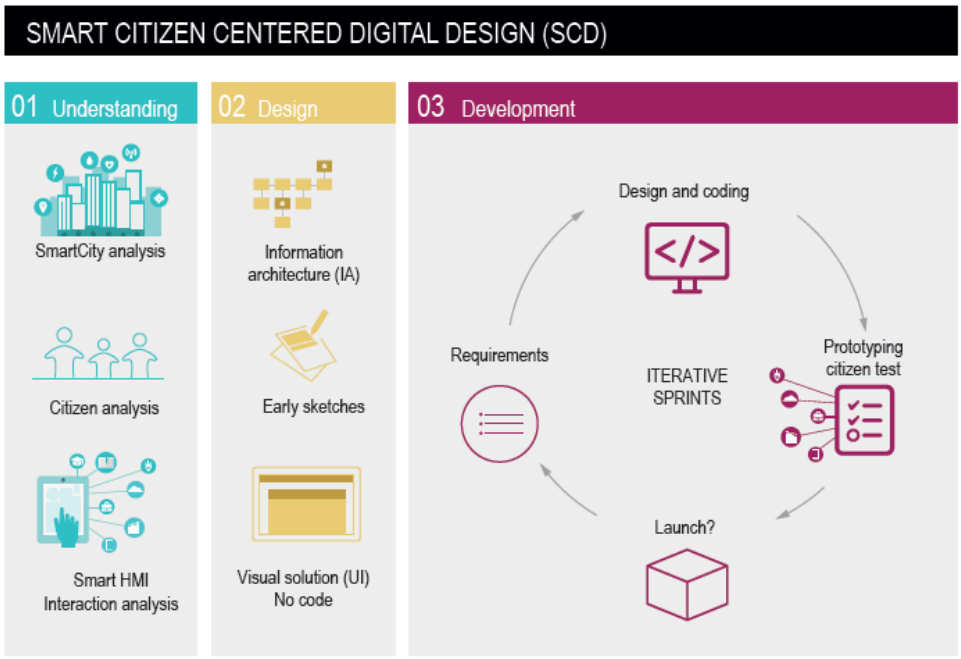

Figure 4: Understanding, Design and Development, SCD Methodology

3.2 Solution development phases

3.2.1 Understanding

In the Understanding phase three main elements should be analyzed: the Smart City, the citizen/user and the interaction between Smart solutions and citizen (Fig. 5).

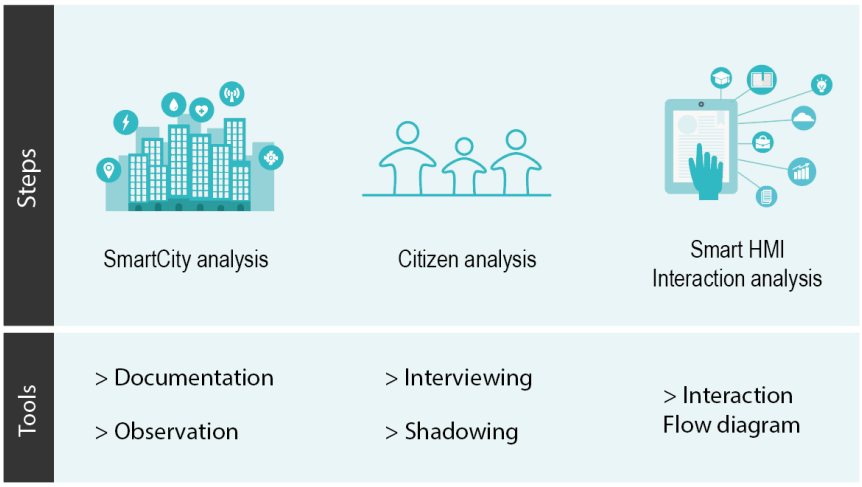

Figure 5. Understanding, steps and tools

During this first phase an analysis should be performed to understand the context of use. In the first place, an analysis of the Smart City key aspects should be performed, it is necessary to know its characteristics and main features. During the analysis of the city, observations should be made at smart-house, workplace, public spaces, etc. And the documentation received from council or data management agencies has to be examined. However, it is not only a question of understanding how the behavior of the citizen is, but also how is the

interaction between the citizen and the ICT smart solutions. Therefore, an analysis should be carried out with different citizens. And the design tool that is proposed for that purpose is the interviewing. Different stakeholders that are linked with the ICT solution should be interviewed. These interviews consisted in finding out the how is the use of the product/service, the behavior of the user and identify needs.

In addition to the interviews, it is proposed the tool Shadowing activity (Daae, 2014), which is based on the observation of citizens/users in real context or Smart City environment. The activity enables to identify what the needs and motivations of citizens are, understanding their lifestyle and habits, in order to create a new solution that is adapted to the citizen. Then, using the Personas tool (Daae, 2014), the citizen information can be displayed, showing features and motivations, which the new ICT solutions must comply with. Finally, the interaction with the ICT solutions should be considered, analyzing the interaction flow of the current architecture of the digital solutions. In order to visualize the architecture, the flow diagram visualization tool is proposed, showing the main screens and the interaction that is realized between them. The diagram enables to understand the behavior of the user when performing the different tasks, and the options provided by the ICT digital solution.

3.2.2 Design

In this phase the concept of the new digital solution is defined. For this, three steps are proposed: development of the new architecture, early sketches (paper prototypes) and visual proposals without coding (programing) (Fig. 6).

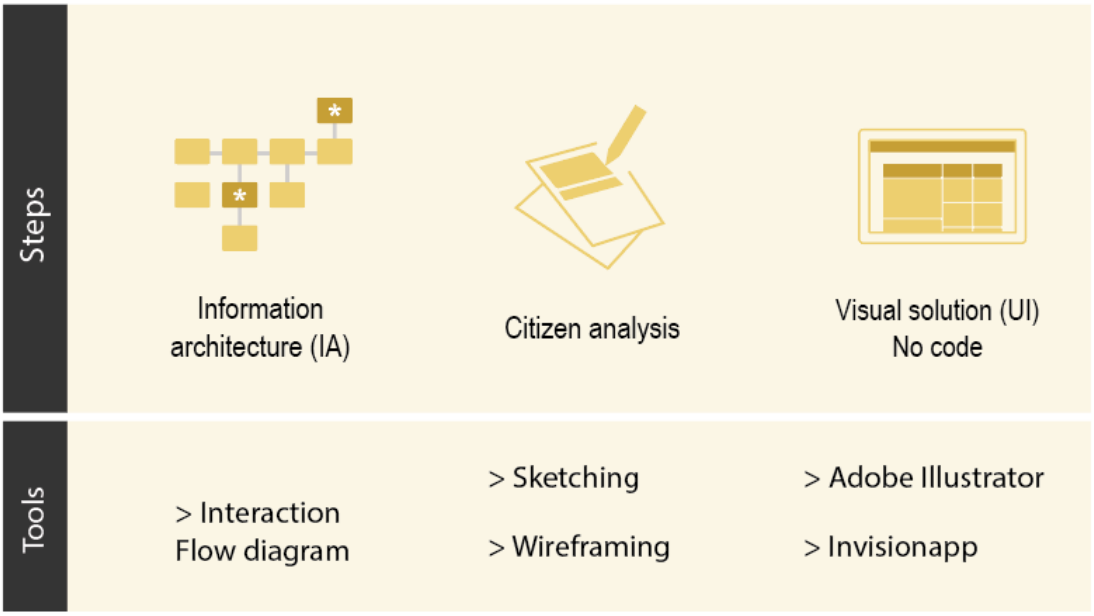

Figure 6. Design, steps and tools

First, based on the insights gathered from the previous phase, the new architecture should be developed, again performing an interaction flow diagram. In it, the navigability has been defined, the steps that must be taken for each action. The flow diagrams should be done by hand, creating not detailed images in order to show the whole architecture and encourage the participation of the citizens/users during the creation process. After defining the first ideas, a test has to be carried out to evaluate if the solution is desired, if the behavior is

natural and if all options are accessed easily and intuitively. Once the feedback is received, in a convergence phase, the new architecture should be completely defined.

Next, detailed sketches should be created by hand, where two tools are proposed: sketching and wireframing (Roberts, 2016). The first of them allows to visualize the idea in a very conceptual way, with few details but enough to be able to give an idea of what it is wanted to show and how. The second tool, enables to work on the concept in a more detailed way, making a scheme of the screens, to visualize how the information will be displayed or where to click to go from one screen to another. Carrying out these two steps by hand helps when getting citizens/users quick feedback, encourages participation and timing optimization (Roberts, 2016). To finish with this phase, visual proposals has to be developed using specific software, but still without code.

In order to get a more valuable and enriching evaluation is it proposed to create digital navigable prototypes, they offer the opportunity to assess the interaction between user and created ICT solution, but without coding. To do this, it is proposed the Invisionapp application (2018), a web tool where interactions between static screens (the designed proposals) are created. Users can access it from anywhere and can try to navigate through the different screens, which offers a great help in making a fairly close evaluation of the screens already programmed. In addition, they can write comments giving opinion.

In order to extract as much as possible from the evaluation, the test has to be performed with several users (n > 10). Thanks to the mentioned Invisionapp platform, the citizens/users could give their feedback easily through the comments.

3.2.3 Development

In the third phase, previously defined design should be developed. To do this, the design team goes through an iterative process, consisting of 4 steps. Within this iterative process, in firstly requirements are defined, then the interfaces are designed and coded so that they are finally evaluated with the users. It is then concluded whether the product is suitable for market launching or requires more time for development (improving on some specific requirements) (Fig. 7).

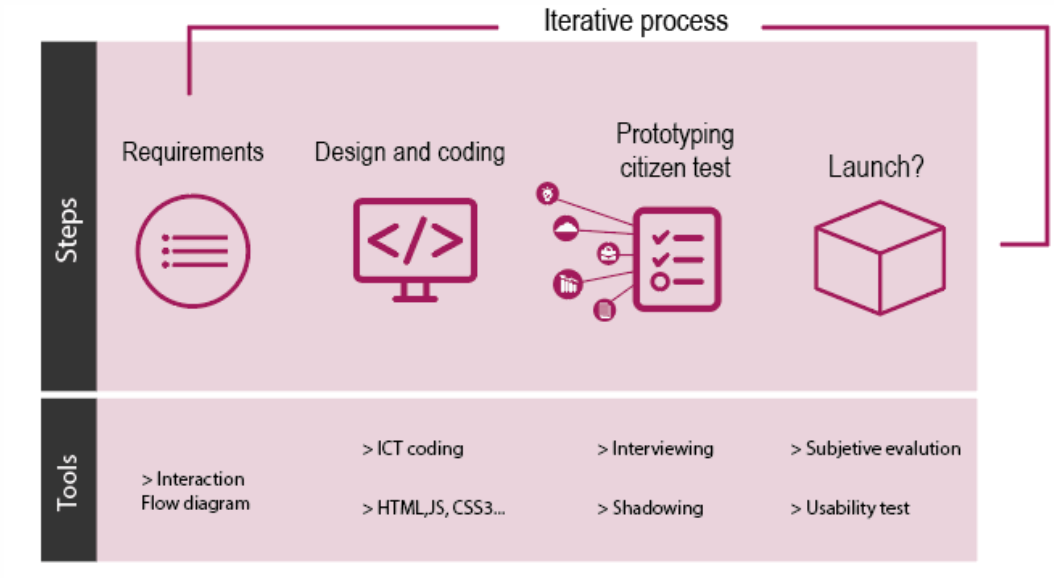

Figure 7. Development, steps and tools

Firstly, based on the feedback received from the citizens/users in the previous phases, the requirements that the new digital solutions have to fulfill are specified (IEEE-std, 1990). Within requirement specification process, in which special emphasis has to be placed on digital interface requirements and non-functional requirements, divided in three phases (Escalona and Koch, 2005). In the first phase (Obtaining Phase), interviewing different stakeholders, the needs of the ICT solution should be defined taking into account multiple perspectives (Kujala et al., 2005). In the second phase (definition phase), each requirement has been stored in a chart: requirement identifier, author, description, importance, urgency and comments. Finally, in the third phase (validation phase) all stakeholders should iteratively validate the new digital solution, verifying that the requirements have been satisfactorily fulfilled.

After that, once all the digital solution and non-functional requirements have been collected and defined, the design and coding of the different interfaces should be developed. In this step, the main objective is to lay out the visual proposal elaborated in the previous phase, using web technologies such as HTML5 for layout and structuring, JavaScript (JS) as a language to develop the interaction of different interfaces and Cascade Style Sheet (CSS) to provide the interfaces with aesthetic design. It should be noted, as previously mentioned, that these interfaces will be mainly based on Web technologies because of the advantages they offer (Lojka, 2015):

1) Multiplatform: Adaptability to different environments and systems.
2) Multi-device: Compatibility with different kind of devices.
3) Evolution and continuous improvement of web technologies that enable to be applied to last ICT solutions.

The third step is to evaluate the proposal with the citizens/users. The SCD methodology proposes to perform as far in advance as possible, in order to waste time and money, when perhaps a so detailed solution is not necessary to get a proper feedback. Therefore, the valuation must be performed with the Minimum Viable Product (MVP).

First tests should be focused on evaluating the specific elements that have formed the MVP, to see if they are understandable and intuitive for the citizens. In the beginning, to perform usability test is proposed, where the following concepts should be evaluated: trust and credibility, navigation and architecture, control and feedback, fault tolerance, content and writing and execution of tasks. For this assessment, shadowing tools and interviews are proposed again. The first one helps to see if they actually perform the supposed steps, and if the application actually has an optimal interactive flow. Afterwards, the interview is conducted to receive citizen's impressions, comment on interesting points identified in the observation and assess the usability.

After receiving feedback, it has to be considered whether to continue working with these elements or on the contrary the product is ready for market launching. In this way, it begins an iterative cycle where it is designed and evaluated until a more appropriate ICT solution is finally created.

4 Analysis of the technologies and tools for solution development

This section provides a collection of technologies and tools that have been identified for the development of HMI solutions for Smart City projects. The identification has been focused on those technologies and tools that fit into the Smart Citizen Centered Digital Design (SCD) defined in the previous sections. This digital design identifies three phases (i.e. Understanding, Design and Development) (see Figure 8) and the technologies and tools are classified in the three groups in the following sections:

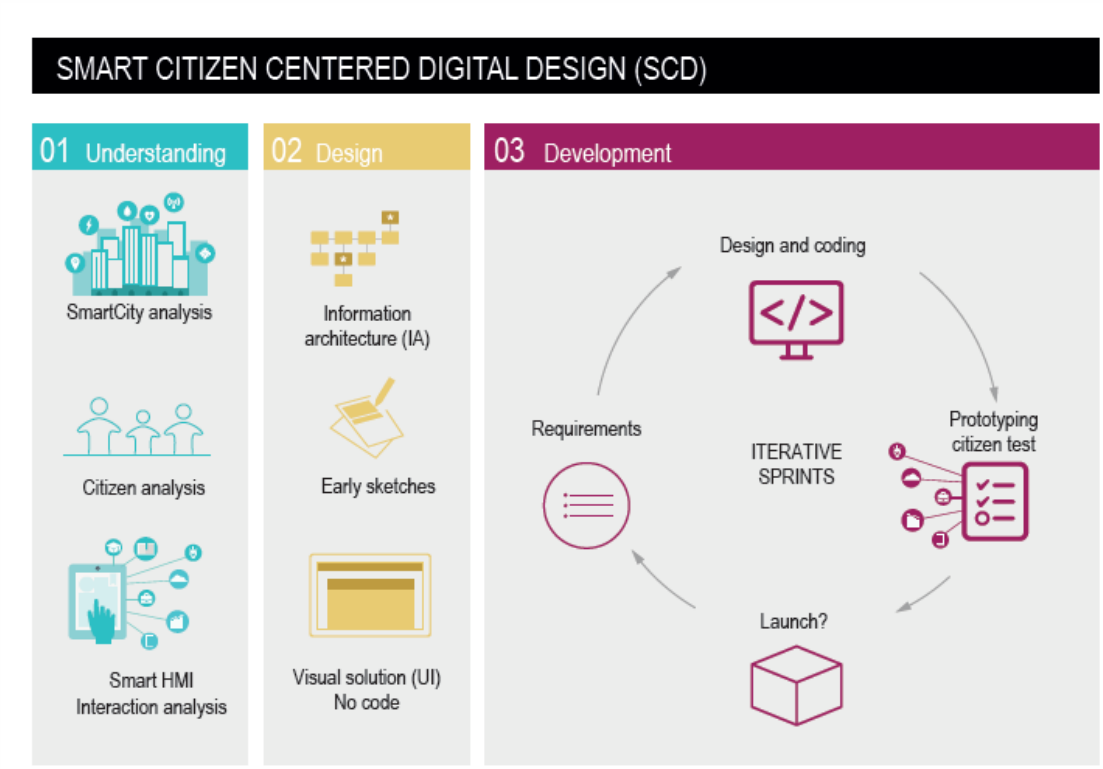

Figure 8. Smart citizen Centered Digital Design (SCD) framework

4.1 Tools and technologies for Understanding Phase

¡Error! No se encuentra el origen de la referencia. shows a list of identified tools and echnologies for development of solutions of Smart City projects to be used in the phase of problem understanding or analysis of user requirements. Most of them are methodologies described in research papers and some others are tools available for collecting requirements or user specifications and feedback. The table shows the name of the method or tool and a short description, including the reference where more information about the tool/technology can be obtained.

Name	Description
Trendwatching	Trendwatching.com is an independent and opinionated trend firm scanning the globe for promising consumer trends, insights and related.

	Reference: http://trendwatching.com/
Cultural probe	Cultural probes (or design probes) is a technique used to inspire ideas in a design process. It serves as a means of gathering inspirational data about people's lives, values and thoughts. **Reference:** Gaver, B., Dunne, T., & Pacenti, E. (1999). Design: cultural probes. interactions, 6(1), 21-29.
User interviews	User interviews can be a great way to extract information from users for user experience understanding, usability understanding and ideation. **Reference:** Wilson, C. (2013). Interview techniques for UX practitioners: A user-centered design method. Newnes.
User expert interviews	User expert interviews can be a great way to extract information from expert users for better understanding, usability understanding and ideation. **Reference:** Wilson, C. (2013). Interview techniques for UX practitioners: A user-centered design method. Newnes.
Focus groups	It is a form of qualitative research consisting of interviews in which a group of people are asked about their perceptions, opinions, beliefs, and attitudes towards a product, service, concept, etc. **Reference:** Kitzinger, J. (1995). Qualitative research. Introducing focus groups. BMJ: British medical journal, 311(7000), 299.
Empathy map	An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. **Reference:** Ferreira, B., Silva, W., Oliveira Jr, E. A., & Conte, T. (2015). Designing Personas with Empathy Map. In SEKE (pp. 501-505).
Shadowing	First-hand observation of daily behaviour. **Reference:** McDonald, S. (2005). Studying actions in context: A qualitative shadowing method for organizational research. Qualitative Research, 5(4), 455-473.
Questionnaires	A questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from respondents. **Reference:** https://surveymonkey.com
Experience map	Experience mapping is a strategic process of capturing and communicating complex customer interactions. The activity of mapping builds knowledge and consensus across your organization, and the map helps build seamless customer experiences. **Reference:** https://medium.com/@wnialloconnor/how-to-build-an-experience-map-5e55b7ee4f32#.o4dng9rwz
Personas	The purpose of personas is to create reliable and realistic representations of your key audience segments for reference. **Reference:** Mulder, S., & Yaar, Z. (2006). The user is always right: A practical guide to creating and using personas for the web. New Riders.
Actor map	Actor Map defines the relationships among the actors in the actor table in terms of how their roles are shared and disparate. **Reference:** Bechmann, A., & Lomborg, S. (2013). Mapping actor roles in social media: Different perspectives on value creation in theories of user participation. New media & society, 15(5), 765-781.

Table 4 List of technologies and tools for problem understanding phase

4.2 Tools and technologies for Design Phase

¡Error! No se encuentra el origen de la referencia. shows a list of identified tools and echnologies for development of solutions of Smart City projects to be used in the phase of solution design. The first set of items represents the most common techniques and methods used in the field of design. The last set represents examples of tools used for the creation of mockups or prototypes of software applications GUIs. The table shows the name of the method or tool and a short description, including the reference where more information about the tool/technology can be obtained.

Name	Description
Brainstorming	Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members. **Reference:** Wilson, C. (2013). Brainstorming and beyond: a user-centered design method. Newnes.
Metaphors and analogues	Metaphors and analogues help designers to understand unfamiliar design problems by juxtaposing them with known situations. **Reference:** Hamilton, A. (2000). Interface metaphors and logical analogues: a question of terminology. Journal of the American society for information science, 51(2), 111-122.
Sketching	A sketch is a rapidly executed freehand drawing that is not usually intended as a finished work. **Reference:** Buxton, B. (2010). Sketching user experiences: getting the design right and the right design. Morgan Kaufmann.
Storyboard	A storyboard is a graphic organizer in the form of illustrations or images displayed in sequence for the purpose of pre-visualizing a motion picture, animation, motion graphic or interactive media sequence. **Reference:** Buxton, B. (2010). Sketching user experiences: getting the design right and the right design. Morgan Kaufmann.
Co-creation	A business strategy focusing on customer experience and interactive relationships. Co-creation allows and encourages a more active involvement from the customer to create a value rich experience. **Reference:** Sanders, E. B. N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. Co-design, 4(1), 5-18.
Early prototyping	Can be defined as a group of techniques used to quickly fabricate a scale model of a part or assembly. **Reference:** Baek, C. B. (2008). User-Centered Design and Development. In M. J. Spector, D. M. Merrill, J. Van Merrienboer, & M. P. Drescoll, Handbook of Research on Educational Communications and Technology (pp. 660-668).
Role-playing	The acting out of the part of a particular person or character (the potential user). **Reference:** Wilshire, B. (1982). Role playing and identity: The limits of theatre as metaphor.
Information architecture map	The information architecture (IA) defines the overarching structure and relationship between all areas of a site (or multiple sites) and informs the sitemap. **Reference:** Wodtke, C., Govella, A., & Christina, W. (2011). Information architecture: Blueprints for the Web. Pearson Education India.

Wireframes	Is a visual guide that represents the skeletal framework of a website. Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose. **Reference:** https://www.invisionapp.com/
Balsamiq Mockups	Rapid wireframing tool that reproduces the experience of sketching on a whiteboard. It allows the collaborative design of mobile applications, desktop applications and web applications. Payware tool. **Reference:** https://balsamiq.com/products/mockups/
Draw.io	Web based tool that supports UML, BPMN and flowcharts. Free tool. **Reference:** https://www.draw.io/
Bpmn.io	Web-based tooling for BPMN, DMN and CMMN. Free tool. **Reference:** http://bpmn.io/
Moqups	Web app that supports the creation and collaboration on the design of wireframes, mockups, diagrams and prototypes. Payware. **Reference:** https://moqups.com
justinmind	Web app that supports the creation of clickable UI prototypes. Payware. **Reference:** https://www.justinmind.com/

Table 5 List of technologies and tools for the design phase

4.3 Tools and technologies for Development Phase

¡Error! No se encuentra el origen de la referencia. shows a list of identified tools and echnologies for development of solutions of Smart City projects to be used in the implementation phase. Technologies and tools identified in the table are grouped according to the following taxonomy:

- **Usability and User Experience:** There are several tools and methods used to identify usability problems or difficulties related to the user interface. This category includes a diverse set of tools and methods ranging from questionnaires to sophisticated tools based on artificial intelligence. Some examples of tools and methods are listed in the table below.
- **Web Application Framework:** Web Frameworks are software frameworks specifically developed for creating web applications and web APIs, and websites. A large list of options is currently available based on the main web programming languages (e.g. Java, PHP, Python). Some of the most popular frameworks are included in the table below.
- **Visualization Library:** The way the data are visualized in the web application is one of the most important issue related with the User Interface. The number of library for producing dynamic, interactive data visualizations in web browsers is enormous. Some of them are focused on charts creation, while others on graph visualization, including 3D visualization, and some of them are of general purpose. A set of visualization libraries is included in the table below.
- **User Interface for IoT Framework:** IoT frameworks provide in general a way to present the data collected to the end user. Some of them are able to use only within the IoT frameworks, while most of them are modules or SDK that can be used externally to the IoT framework. Some examples of the User interface modules for IoT Frameworks are listed in the table below.

- **GIS Data Visualization:** Including georeferenced information is of great interest for several applications and users. Some of the tools included in the other categories include geomapping functionalities, however there are specific libraries for displaying information on top of a map. 3D visualization of maps is also included in some of the identified tools. Some examples of georeferenced data visualization libraries are listed in the table below.

The table shows the category, the name of the method or tool and a short description, including the reference where more information about the tool/technology can be obtained.

Category	Name	Description
Usability and User Experience	Usability test	Usability testing is a way to see how easy to use something is by testing it with real users. Users are asked to complete tasks, typically while they are being observed by a researcher, to see where they encounter problems and experience confusion. **Reference:** https://www.usability.gov/how-to-and-tools/methods/usability-testing.html
	Heuristic evaluation	A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface (UI) design. **Reference:** https://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html
	Eyetracking	*Eye tracking* is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. **Reference:** Lasa Erle, G., & Justel Lozano, D. 2016 Ene. Nuevo modelo de evaluación de ideas conceptuales para productos y servicios basados en la experiencia de usuario. DYNA Ingeniería e Industria. [En línea] 91:1
	Facereader	FaceReader is the premier professional software for automatic analysis of facial expressions. **Reference:** Den Uyl, M. J., & Van Kuilenburg, H. (2005, August). The FaceReader: Online facial expression recognition. In Proceedings of measuring behavior (Vol. 30, pp. 589-590).
	Questionnaire	A questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from respondents. **Reference:** https://surveymonkey.com
	Shadowing	First-hand observation of daily behaviour. **Reference:** McDonald, S. (2005). Studying actions in context: A qualitative shadowing method for organizational research. Qualitative Research, 5(4), 455-473.
Web Application Framework	Python Django	Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF). Django's primary goal is to ease the creation of complex, database-driven websites. **Reference:** Martinez, I., Reguera D. et al. 2015 Desarrollo y aplicación de una metodología para el diseño de soluciones de visualización en el proyecto europeo CITyFiED . www.djangoproject.com
	Ruby Rails	Ruby on Rails, or simply Rails, is a server-side web application

		framework written in Ruby under the MIT License. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing. **Reference:** https://rubyonrails.org/
	Java Spring	The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is open source. **Reference:** https://spring.io/
	PHP Symfony	Symfony is a PHP web application framework and a set of reusable PHP components/libraries. Symfony was published as free software on October 18, 2005 and released under the MIT license. **Reference:** https://symfony.com/
	Jaggeryjs	JS framework to build webapps. **Reference:** http://jaggeryjs.org /
	AngularJS	JavaScript-based open-source front-end web application framework to address many of the challenges encountered in developing single web page applications. **Reference:** https://angularjs.org/
Visualization Library	Datavisual	Datavisu.al is a simple, accessible tool to make data visualizations. **Reference:** http://datavisu.al/
	ChartBlocks	Chartblock s is a webtool to develop online datagraphics. **Reference:** http://www.chartblocks.com/
	DataWrapper	Datawrapper is an open source tool helping everyone to create simple, correct and embeddable charts in minutes. **Reference:** https://www.datawrapper.de/
	Bokeh	Bokeh is a Python interactive visualization library that targets modern web browsers for presentation. **Reference:** http://bokeh.pydata.org/en/latest/
	Seaborn	Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. **Reference:** https://seaborn.pydata.org/
	Google Chart	This tool lets you develop online charts in a useful way. **Reference:** https://developers.google.com/chart/
	D3.js	D3.js (or just D3 for Data-Driven Documents) is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely implemented SVG, HTML5, and CSS standards. It is the successor to the earlier Protovis framework. In contrast to many other libraries, D3.js allows great control over the final visual result. **Reference:** https://d3js.org/
	C3.js	C3 makes it easy to generate D3-based charts by wrapping the code

		required to construct the entire chart. C3 provides a variety of APIs and callbacks to access the state of the chart. **Reference:** http://c3js.org/
	Dc.js	dc.js is a javascript charting library with native crossfilter support, allowing highly efficient exploration on large multi-dimensional datasets (inspired by crossfilter's demo). It leverages d3 to render charts in CSS-friendly SVG format. Charts rendered using dc.js are data driven and reactive and therefore provide instant feedback to user interaction. **Reference:** https://dc-js.github.io/dc.js/
	Teechart	Library for charts development. **Reference:** https://www.steema.com/product/html5
	Polycharts	Polychart2.js is an easy-to-use yet powerful JavaScript graphing library. It takes many ideas from the Grammar of Graphics and the R library ggplot2, and adds interactive elements to take full advantage of the web. **Reference:** https://github.com/Polychart/polychart2
	Fusioncharts	Javascript library for chart creation. **Reference:** http://www.fusioncharts.com/
	Highcharts JS	Interactive javascript charts for the web. **Reference:** http://www.highcharts.com/
	Tableau Public	Software for datavisualization. **Reference:** https://www.tableau.com/
	ChartIO	Chartio is a cloud-based business analytics solution that allows everyone to explore their data and create business dashboards. **Reference:** https://chartio.com/
User Interface for IoT Framework	Dashboard Server	Web application (open source) that allows to: • Create a dashboard (visually) with data visualization and share it across different organizational roles • Create Microsites to present growing information of a focused domain (collection of dashboards) • Supported data sources: • Batch Data Source • Relational Database Source • Realtime Data Source • REST Data Source This application is the data visualization component of the WSO2 Data Analytics Server and uses Jaggeryjs. **Reference:** https://github.com/wso2/product-ds
	Geo Dashboard	Web application (open source) that generates real time information about geo special objects and a variety of alerts and warnings as follows: • Speed alert • Proximity alert • Within alert • Stationery alert • Congestion Alert This application is a component of WSO2 DAS that leverages Jaggeryjs. **Reference:**

		https://docs.wso2.com/display/DAS310/Geo+Dashboard
	Pentaho Internet of Things Analytics	Pentaho Analytics provides the ability to blend operational data with data from IT systems and deliver batch and real time analytics. **Reference:** http://www.pentaho.com/internet-of-things-analytics
	Power BI	Power BI is a suite of business analytics tools that deliver insights throughout an organization. It is capable to connect to hundreds of data sources and drive ad hoc analysis to generate and publish reports. This product is integrated with the Azure Analysis Services. **Reference:** https://powerbi.microsoft.com/
	Tableau	Business analytics and visualization tool **Reference:** https://www.tableau.com/
	Qlik	Business analytics and visualization tool **Reference:** http://www.qlik.com/us/
	SAS	Business analytics tool **Reference:** https://www.sas.com
	Information Builders	Business Intelligence, analytics and data management tool. **Reference:** http://www.informationbuilders.com/
	Amazon QuickSight	Business Intelligence tool. **Reference:** https://quicksight.aws/
	Spagobi	Business Intelligence tool. **Reference:** http://www.spagobi.org/
	FIWARE Advanced Web Based User Interface	FIWARE Enablers that support the building of Web Based User Interfaces for FIWARE IoT platform. **Reference:** https://catalogue.fiware.org/chapter/advanced-web-based-user-interface
	Cumulocity Web SDK	This SDK allows to build AngularJS based applications on top of Cumulocity IoT platform. **Reference:** https://www.cumulocity.com/guides/web/introduction/
	Kaa SDK	This SKD enables the generation of applications that run on top of Kaa IoT platform infrastructure. The Kaa server allows the generation of libraries for data access from target platforms (clients). **Reference:** http://docs.kaaproject.org/display/KAA/Your+first+Kaa+application
GIS data Visualization	Leaflet	Leaflet is a widely used open-source JavaScript library for creating mobile interactive maps. It is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has an easy to use and well-documented API and a simple, readable source code. **Reference:** http://leafletjs.com/
	OpenLayers	OpenLayers is a free, open-source JavaScript library for displaying map data in web and mobile browsers. It provides a widely used and well-documented API for building rich web-based geographic applications. Furthermore, it can display map tiles, vector data and markers loaded from any source and can be easily customized and extended thanks to its flexibility. **Reference:** https://openlayers.org/
	Google Maps API	The Google Maps API give developers several ways of embedding Google Maps into web pages or retrieving data from Google Maps,

		and allow for either simple use or extensive customization. There are several API offerings depending on the needs: Web APIs, Web Service APIs and Mobile APIs. **Reference:** https://developers.google.com/maps/
	ArcGIS JavaScript API	The ArcGIS JavaScript API is a lightweight way to embed maps and tasks in web applications. It combines modern web technology and powerful geospatial capabilities enabling to create high-performing apps and smarter visualizations of the data. Latest version allows developers to build full-featured 3D applications powered by web scenes than can include rich information layers such as terrain, basemaps, imagery, features, integrated mesh layers, and 3D objects than can be streamed in via tile, feature, image and scene services. **Reference:** https://developers.arcgis.com/javascript/
	Cesium	Cesium is an open-source JavaScript library for creating 3D globes, 2D maps and Columbus view (2.5D) in a web browser without a plugin. It uses WebGL for hardware-accelerated graphics, and is cross platform, cross-browser and tuned for dynamic-data visualization. **Reference:** https://cesiumjs.org/
	CARTO	CARTO (formerly CartoDB) is a Software as a Service (SaaS) cloud computing platform that provides GIS and web mapping tools for display in a web browser. The software is built on PostGIS and PostgreSQL. In addition, it offers an open source tool (CARTO Engine) that allows developers to use various APIs to create dynamic and advanced geospatial datasets and scalable maps for the development of their own applications. **Reference:** https://carto.com/
	HERE	HERE Platform gives access to robust location functionality for web, desktop and mobile. It offers RESTful web services (for integrating core functionalities and platform extensions), JavaScript API (for web development) and mobile SDKs (for native development). **Reference:** https://here.com

Table 6 List of technologies and tools for the development phase

5 Selection of technologies and tools

5.1 The methodology: technology selection process

One of the most critical aspects of the selection process is the methodology used to make the best selection. Regarding the selection methodologies, the most expanded ones are the Multi Criteria Decision Making models (MCDM).

Multi Criteria Decision Making (MCDM) is a discipline of the operational research, which concerns with structuring and solving decision and planning problems involving multiple criteria. The purpose is to support decision-makers facing decision problems where multiple conflicting criteria need to be considered. Typically, there does not exist a unique optimal solution for such problems and it is necessary to use decision-maker's preferences to differentiate between solutions. Different alternative methods have been defined in order to support decision making with conflicting criteria. These methods provide systematic approaches for evaluating and scoring alternatives with multiple criteria. However, they are not easy to implement.

Multicriteria methods can be classified mainly in two categories: Utility-based models (single score for every alternative) and Outranking methods (pairwise comparisons between alternatives). Examples of the first category are Multi-Objective Programming (MOP) and Simple Multi-Attribute Rating Technique (SMART). MOP is a method for multi-objective optimization which is based on providing a weight to each objective and then maximizing the weighted sum. This method has been applied to a wide array of problems, namely production planning, oil refinery scheduling, health care, portfolio selection, distribution system design, energy planning, water reservoir management, timber harvest scheduling, problems of wildlife management, etc. MOP requires decision makers to specify the exact values of the weights of the individual criteria (De Montis et al 2008). However, accurate weight values are difficult to obtain. Determining weight is a difficult task to implement the MOP approach. SMART is a method similar to MOP. In SMART the final score of a given alternative is calculated as the total sum of the value of each criterion multiplied with the weight of that criterion. This is done mathematically by the decision-maker by means of a Value Function. The simplest and most widely used form of a value function method is the additive model, which in the simplest cases can be applied using a linear scale (DTU Transport 2014)**¡Error! No se encuentra el origen de la referencia.**. The same problem presented for the MOP method is faced by SMART.

The most representative method of the second category (Outranking methods) is the Analytic Hierarchy Process (AHP) (Kornyshova et al. 2007). AHP is based on the definition of criteria and alternatives to achieve a goal (See **¡Error! No se encuentra el origen de la referencia.**). From a procedural point of view this method consists of three steps: (1) construct suitable hierarchies; (2) establish priorities between elements of the hierarchies by means of pairwise comparisons; (3) check logical consistency of pairwise comparisons.

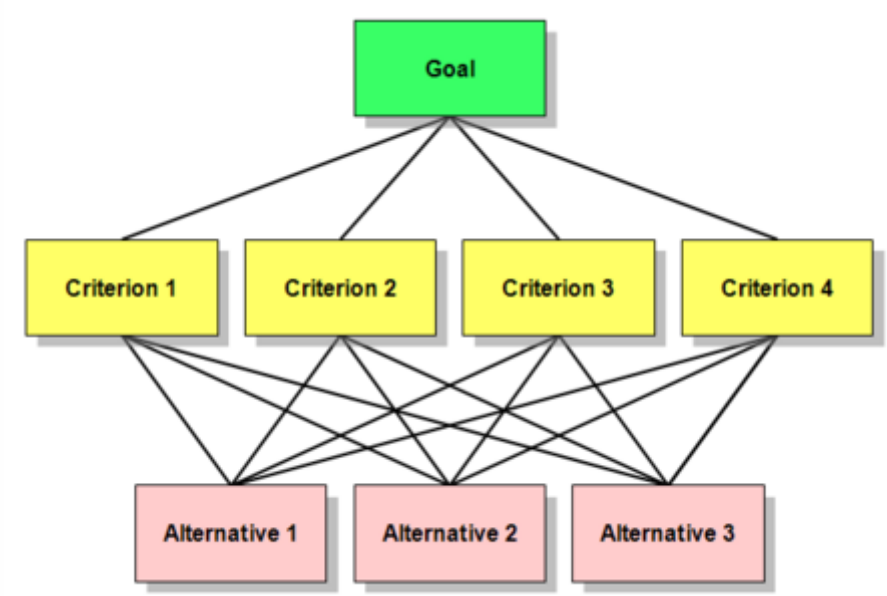

Figure 9. Simple AHP Hierarchy

AHP provides interactive comparisons for users to obtain weights. Decision makers are required to make comparisons of pairs between the criteria and the alternatives of the provider under a particular criterion. However, the results are highly dependent on the subjective judgments of the decision makers. Decision makers need to specify not only the direction of relative importance, but also the degree of relativity.

An alternative to the AHP method is the Data Envelopment Analysis (DEA) method. In this case the method automatically derives the optimal weights for each criterion as a function of the value of each alternative for each criterion. The DEA approach does not require the decision maker to pre-define the weights. The solutions of the DEA models require a linear optimizer, which is available to a decision maker (Wang 2008).

Beside this,, based on the main feature of the project, which are the technologies, it is considered that the methodology must be focused on this aspect. So, the methodology proposed for the selection of technologies and tools in order to build HMIs in Smart City projects is inspired on the “Integrated technology selection process” proposed by Shen, Y.C. et al (2010).

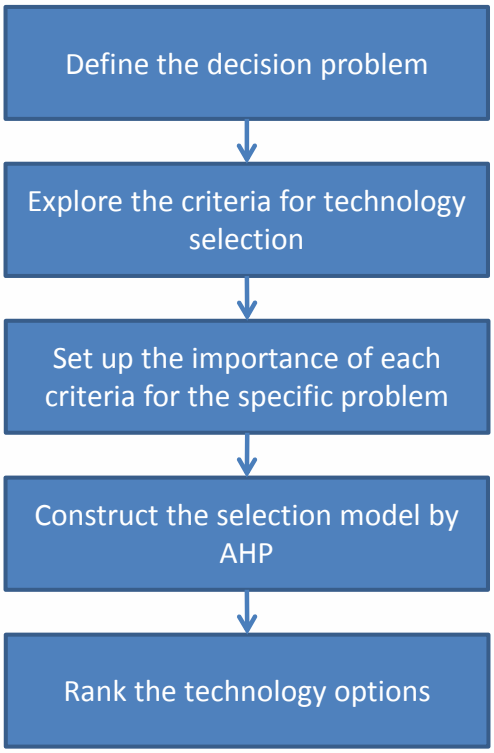

Figure 10. Technology Selection Process

The technology selection process that is used within this project is divided into 5 sequential steps (Figure 10). A detailed description of each of the steps is described below:

- **Define the Decision Problem:** in this stage the aim is to identify the scope for which technology will be analyzed and selected.
- **Explore the Criteria for Technology Selection:** the objective is to explore the different options for technology selection criteria, such as benefit, user participation, feedback capability, skills needed, cost, risk, and so on.
- **Set up the importance of each criteria for the specific problem:** in this stage the aim is to sift through the important criteria integrating the views of experts of different backgrounds. The importance rate should be widely defined for all the criteria factors, respecting the same scale (e.g. from 0 to 3).
- **Construct the selection model by AHP:** the aim is to obtain the weights of technology selection criteria by employing the Analytic Hierarchy Process (AHP) to construct the technology selection hierarchy.
- **Rank the technology options:** Finally, the objective is to rank the technology options. Experts evaluate every technology option based on the technology selection model.

This specific selection process combines different types of approaches in the hope of entailing a more objective and practical technology selection process, concerning from aspects that are specific for both, HMI development and Smart City context. This proposed model has not been used in other Smart City related projects yet.

5.2 Selection criteria

In this section are listed the criteria for the understanding, design and development phase. Each criterion has an associated weight from 0 to 3 according to the different levels of evaluation. After assigning the weights to the different criteria, these will be used to implement the evaluation following the method defined in previous sections.

5.2.1 Criteria for the Understanding phase

The criteria for the understanding phase are covered by 5 different aspects: User participation, ease of access/use, feedback capability, technical skills needed and cost. Being within the Smart citizen Centered digital Design (SCD) framework, one of the most valuable aspects would be the one that measures the user participation.

The definition for all criteria measurements is explained below:

User Participation

User Participation (Kaasbøll, J. 2007, He, J., King W. 2008) refers to feedback from target users during the development process of the system. It can be vital in order to design a system that not only works properly, but is also well-received by the target users.

How to evaluate:

- 0 = No participation: the user is considered without having to participate. 0
- 1 = Evaluates: the user adopts a critical attitude about a product or service with the purpose of enhance it.
- 2 = Creates: the user identifies problems and/or suggests solutions or improvements.
- 3 = Develops: the user develops the project by engaging with the team internally or externally.

Ease of access / use

Ease of access/use denotes the amount of effort the user requires to learn and use a given technology. Ease of use does not indicate how optimal the capabilities of the technology are to achieve objectives. However, it is a requirement for the user to fully exploit the capabilities the technology has. Accordingly, this is an important factor that must be considered when selecting a technology.

How to evaluate:

- 0 = Very difficult to use
- 1 = Difficult to access/use
- 2 = Easy to access/use
- 3 = Very easy to access/use

Feedback Capability

Another important aspect when selecting technologies is the feedback capability it has. The feedback is the information exchanged between user and system. It refers to both the information that the system returns to the user as a response to the user's actions, and to what the system collects from the user to improve its own future behavior. Better feedback capability means shorter development times, better error resolution, and an improved user satisfaction.

How to evaluate:

- 0 = No feedback
- 1 = Short feedback
- 2 = Reasonable feedback
- 3 = Extended feedback

Technical skills needed

Technical skills are the abilities and knowledge required to carry out specific tasks effectively, using the technology. Selecting a technology involves evaluating its technical skills requirements. Those will determine how fit the technology is for the user, and how fit the user is for working with it.

How to evaluate:

- 0 = No skills needed
- 1 = General or low skills needed
- 2 = Some specific skills needed
- 3 = High qualification needed

Cost

Cost can be seen as the sum of acquisition, deployment and maintenance or support cost (Nourse W. 2014). The acquisition cost covers the expenses of buying the necessary elements of hardware and software to complete the project. Deployment costs refer to those associated with setting up the software including installation of components or configure the working environment. Finally, the maintenance or support costs include the license, upgrading or guarantee costs. All of these costs also have impact when selecting a certain technology and should be taken into account.

How to evaluate:

- 0 = Low cost to implement
- 1 = Reasonable cost to implement
- 2 = High cost to implement
- 3 = Very high cost to implement

5.2.2 Criteria for the design phase

License (price)

Software tools are currently licensed in a great variety of ways. In the proprietary software the owner company request money for providing access to use a software application. In open source applications different kinds of licenses allow software to be used, modified and shared.

The way that is paid for most common types for licensing are:

- 0 = Payware
- 1 = Trial version for a period of time Free for non-commercial purposes
- 2 = Free for non-commercial purposes
- 3 = Freeware

Desktop vs Web

Desktop applications require to be installed in the user computer. The main advantage is the speed of use, mainly for tasks which required much processing or the management of a huge amount of data. On the contrary, portability is an issue as well as the updating of the software versions. Web applications are the most suitable solution for applications which requires to be used by different users or when the user can be located in different places. The requirement of an internet connection and a high speed connection in same cases is the mayor drawback of this alternative.

How to evaluate:

- 1 = Only Desktop application / Only Web application
- 3 = Desktop and online versions
- 0 = Not available / Not applicable

Life preview

One important aspect of the design phase is the possibility to visualize the final output during the design phase. This aspect is linked to the concept WYSIWYG (What you see is what you get).

How to evaluate:

- 1 = No
- 3 = Yes (Life preview)

Multiplatform

The platform to run an application is a combination of hardware and software, including mainly the computer architecture and operating system. The concept of multiplatform allows software code to be interpreted by different platforms without having to modify the source code of the application. This is an implicit aspect of web applications, but is very relevant for desktop ones.

How to evaluate:

- 3 = Multiplatform (Windows, Mac OS, Linux, Android…)
- 2 = Just for some PC platforms
- 1 = Only one platform
- 0 = Not available / Not applicable

5.2.3 Criteria for the development phase

The development phase can be divided in three areas: administration, technical, results and data visualization.

ADMINISTRATION

License (Price)

Evaluation of software options is expected to occur prior to selection and implementation of a production urban platform, and the cost of the licenses is a determining factor in that

evaluation. Obviously, the choice of the software cannot depend solely on it, but the budget limitations are an aspect to take into account. Free software licenses are royalty-free and permit the copying, distribution and modification for any use, even commercial, so it could assess whether some Operational Support System can accomplish our needs.

How to evaluate:

- 0 = Payware
- 1 = Trial version for a period of time Free for non-commercial purposes
- 2 = Free for non-commercial purposes
  3 = Freeware

Installation Requirements

Most software defines some system requirements to be used efficiently. These requirements are often used as a guideline as opposed to an absolute rule. However, a server that meets only the minimum requirements may not function properly, especially when server runs certain high demand functions. These requirements are appropriate just for testing, so knowing installation requirements is fundamental.

How to evaluate:

- 3 = Meet
- 0 = Not Meet / Not Applicable

Multilingual

At present, we live in a totally globalized society, so people from around the world can use the same applications and computer programs (Muratza, M., Shwan, A. 2009). In order to gain competitive advantage, offering a multilingual platform is crucial, as well as having a comprehensive infrastructure for multilingual support. This guarantees a greater diffusion of our work and possibility of success.

How to evaluate:

- 1 = Monolingual
- 2 = Bilingual
- 3 = Multilingual
- 0 = Not available / Not applicable

TECHNICAL

Documentation

Special attention needs to be given to developer's documentation. Documentation should include requirements of the system, identifying attributes, capabilities and characteristics and describing what software do or shall do, technical documentation associated with the source code and the server environments, installation and configuration documents, manuals for the end-user, system administrators and support staff, etc. Having a reliable documentation requires time and effort, but it is helpful for development, maintenance and knowledge transfer.

How to evaluate:

- 3 = Considerable amount of documentation
- 2 = Average amount of documentation
- 1 = Small amount of documentation
- 0 = No documentation

Learning curve

A learning curve is a concept that graphically depicts the relationship between learning and experience over a defined period of time. All solutions, platforms or software generally require a learning curve. For this reason, it is important to lessen the learning curve, that is, the time lost in learning to utilize the platform created, making it as simple as possible, including documentation (FAQs, tutorials, manuals, etc.) and displaying an intuitive design, without complex interfaces.

How to evaluate:

- 3 = Very Easy to learn
- 2 = Easy and fast to learn
- 1 = Difficult or slow learning
- 0 = Very difficult to learn

Scalability

Scalability is an attribute that describes the ability of a software or system to grow and manage increased demand. A software or system that it is described as scalable has an advantage because it is more adaptable to the changing needs or demands of it users or clients. Scalability typically involves adding resources to the system but should not require changes to the deployment architecture. The amount of users, data, and services of a Smart City platform is massive, and can increase over time, so scalability is relevant to many functional requirements, such as data management or service management, as it means the platform is ready to handle the influx of demand, increased productivity, trends, changing needs, and even presence or introduction of new competitors.

How to evaluate:

- 3 = High scalability
- 2 = Reasonable scalability
- 1 = Low scalability
- 0 = No scalability

Maintainability

The maintainability is defined as the degree to which an application is understood, repaired, or enhanced. In other words, it is the ease with which a software product can be modified. Software maintainability is important because it is a relevant part of the cost related to a project. Moreover, understanding software maintainability allows identifying improvement areas as well as determining the value supplied by current applications or during development changes. In a Smart City platform, the maintaining can be a challenge due to its size as well as the very large number of devices connected to the platform.

How to evaluate:

- 3 = Easy to maintain
- 0 = Difficult to maintain

Security and data privacy

Malicious users can make fraudulent use of services and data provided by a platform. Since a Smart City platform collects and manipulates several citizen-sensitive data, the challenge is to use these data while hiding, or to avoid saving identifiable information. Some of the strategies used to achieve this requirement are cryptography, tokens to control the access to the data that users can manipulate, and anonymization, avoiding this way attacks to the developed system and information thefts.

How to evaluate:

- 3 = High
- 2 = Moderate
- 1 = Limited
- 0 = None

Multiplatform

The usability of the same software or platform in different environments is an essential aspect today. A Multi-platform or cross-platform software allows reach more customers and it is easier and faster to maintain and deploy changes if there is a single code. The portability is the key issue for reducing the development cost when software with the same functionality is produced for several computing platforms. It may involve building executable programs for each platform that it supports, but not necessarily.

How to evaluate:

- 3 = Multiplatform (Windows, Mac OS, Linux, Android…)
- 2 = Just for some PC platforms
- 1 = Only one platform
- 0 = Not available / Not applicable

Level of interactivity

The level of interactivity refers to how the platform accepts and responds to input from users (Pappas 2015). For purposes of participation in designing a Smart City, the platform has to enable interested citizens to participate in the different processes of design and planning of the communities, using data, models and scenarios informed by contemporary ICT. The aim is that citizens can play an active role. In this way, it is also interesting that users can interact with software that enables them to learn more by engaging with other uses online, as well as they can manipulate or upload information.

How to evaluate:

- 3 = High interaction / full immersion
- 2 = Moderate interaction
- 1 = Limited interaction

- 0 = Passive. No interaction

RESULTS AND VISUALIZATION

Real time data visualization

Nowadays, there is an increment on the need of real-time data. Operators and managers need to respond to urban problems, such as traffic jams, accidents or floods, in short time. Visualizations with real-time data allow extracting useful information, in such a way that users can make decisions on the fly and in an efficient way. The fields of application are manifold: observing in real time becomes a mean to understanding the present and anticipating the future. In addition, wireless communications devices have created new dimensions of interconnectedness between people, places and urban infrastructures, in such a way that people can record positional information using their GPS.

How to evaluate:

- 3 = Real time
- 2 = Near-Real time
- 1 = Batch processing

3D Visualization

Creating an interactive 3D visualization for Smart City projects allows having a wide overview of how those projects will impact the surrounded area, but also it will display a large range of geo-referenced data (simulation outcomes, socio-economic data, urban indicators, etc.). In addition, the developers of services for Smart Cities will be able to validate them without the need for previous deployment of real sensors in the city, through the virtual deployment of sensors. It may be said with certainty that 3D visualization makes easier and more collaborative the city management and the seeking of solutions.

How to evaluate:

- 3 = Yes
- 0 = No

Geolocation

The concept of Smart Cities refers to a city model where resources are optimized, improving the citizen life's quality through technological innovation. In this area, the geolocation or the identification of the real-word geographic location of an object, play an essential role in the management, processing and advanced analysis of spatial information in the city. Thanks to geolocation tools it is possible to integrate, store, edit, share or display geographically referenced information. In the environment of the Smart City, these are suitable for performing various operations, such as creating interactive queries, analyze spatial information or edit data and maps. It is important when creating or adding geolocation tools, of course, that they have an easily and intuitive use.

How to evaluate:

- 3 = Yes
- 0 = No

Dashboard Creation

A dashboard transforms data into information and facilitates the decision making. The creation of Smart City dashboards aims to furnish a universal instrument to local and central government, able to support strategic decisions, to drive investments, to measure reached goals and to compare different smart solution each other's. In short, it is meant to display real-time data in a highly accessible manner, in such a way that city officials can immediately respond to incidents and improve services. Furthermore, displaying such information together in one place may be a way to use Smart City data to empower people to make more informed and lower-impact decisions, to connect citizens with the data being collected and to enable them to understand what is happening in a city system at any point in time. This task is made easier by the dashboards potential to be customized according to user preferences.

How to evaluate:

- 3 = Yes
- 0 = No

Reports Creation

The ability to create reports in the format needed is a critical aspect in many applications. In the area of the Smart City, a high-quality reporting tool is necessary for integrating data from different sources and exporting data, keeping citizens informed that way. The reports generated should look beautiful digitally and printed and be easily created without external support.

How to evaluate:

- 3 = Yes
- 0 = No

Scenario Analysis

Scenario analysis is a method to ascertain and analyze possible events that can take place in the future, a way of structuring thinking about the future. For this reason this tool can be very useful in Smart Cities to make projections and identify potential problems and increase preparedness to handle them. Outcomes are visible given the different scenarios envisaged and so are the paths that lead to them from current situation, giving city officials more scope to refine and adjust plans accordingly.

How to evaluate:

- 3 = Yes
- 0 = No

5.3 Interesting technologies for SCD framework = case study

5.3.1. Case Study

As seen in previous sections, it is possible to identify different tools and technologies for each of the three main phases described for the SCD methodology: understanding, design and development.

SmartEnCity platform

Within the scope of smart cities, the SmartEnCity platform mainly provides tools for citizenship, in which the participation of individual citizens is crucial to turn these type of platforms in a successful case. In addition, it provides tools for institutions to improve the management and monitoring of urban assets. Enabling citizens to monitor energy consumption, or access real-time traffic information, increases the total number of participant citizens and their engagement.

The selection of the proper tools for all phases for the development of this platform is essential. The selected tools should fit together in order to allow the interconnectivity between all the involved elements. In this way it is possible to get the whole system solution together with all functionalities and required services.

5.3.2 Evaluation of the selected technologies with the proposed approach

The following section details the method used to evaluate the technologies selected for the SmartEnCity use case described in the previous sections.

The process followed was as follows:

Weight Category: Each phase has been divided into one or more categories. In principle, since there were not categories under the Understanding and Design phases, the method has been adapted, creating a single category, *General,* for these two phases. The sum of the scores of the values of these categories has been fixed in 6 points as follows:

Understanding:

- Cat General = 6

Design

- Cat General = 6

Development

- Cat Administration = 1
- Cat. Design = 2
- Cat. Results and Visualization = 3

Importance of the criteria: Once the weights of the categories have been set, the weights of criteria are defined from 1 to 4 like so:

1 = Desirable to have

2= Necessary

3= Important

4= Essential

Tool evaluation: Then, for each tool, the degree in which the tool meets a criterion is evaluated from 0 to 3. As each criteria has different evaluation levels and definitions, in the section 5.2, it is indicated the codification of each level for criteria evaluation.

However, all codifications follow the same rule, so generally it is possible to define the codifications as:

0 = Not available/Minimum

1 = Limited/Minimum

2 = Approved

3 = Excellent

There are also criteria with only two possible evaluations, this is the case for instance of Reports Creation, Dashboard Creation or Geolocation. In those cases, in order to maintain the codification from 0 to 3 as indicated previously, it has been mapped as:

0 = Not Meet/No

3 = Meet / Yes

Once all the values of categories, criteria and tools have been assigned, the totals are calculated:

```
Partial Evaluation = Importance of the criteria * Tool evaluation
Total Evaluation = Partial Evaluation Average
Section Cat = Total Evaluation * Weight Category
Total = ∑ Section Cat
```

As we have seen in the previous sections, for our use case, numerous tools have been included to evaluate in the three phases. These tools cover a wide range of different typologies and include libraries, techniques, applications, frameworks, processes and models.

Due to the different nature of the selected tools, the evaluation has taken into account not only the use of the tool itself but also the results that can be obtained when using the tool.

5.3.3 Results

The results obtained using the methodology proposed for the particular case of SmartEnCity are described below. It should be noted that the results and calculations now presented and included in detail in the annex are those that have been applied for the case of SmartEnCity and will therefore change depending on the use case to which they are applied.

Understanding Phase:
Focus groups **54**

Shadowing	**54**
Questionnaries	**54**
User interviews	50,4
User expert interviews	50,4
Empaty maps	44,4
Personas	44,4
Experience maps	39,6
Actor maps	39,6
Cultural probe	36
"Trendwatching Evaluation"	24

Design Phase:

Sketching	**49,5**
Storyboard	**49,5**
Draw.io	**49,5**
Early prototyping	45
Information architecture map	45
Wireframes	43,5
Bpmn.io	43,5
Brainstorming	43,5
Balsamiq Mockups	31,5
Moqups	31,5
justinmind	31,5
Metaphors and analogues	21
Role-playing	16,5
Co-creation	12

Development Usability Category

Questionnaire	**18,67**
Usability Test	16,49
Shadowing	16,49
Heuristic evaluation	15,17
Facereader	14,99
Eyetracking	13,49

Development Web Application Framework Category

Java Spring	**26,99**
AngularJS	**26,99**
Ruby Rails	26,13
PHP Symfony	26,13
Jaggery.js	26,13
Python Django	24,56

Development Visualization Libraries Category

D3.js	**29,20**
Google Chart	**28,63**
Seaborn	27,63
Bokeh	26,77
Fusioncharts	26,85
Dc.js	26,49
ChartBlocks	26,27

Teechart	26,27
Highcharts JS	26,27
C3.js	25,92
Polycharts	25,92
DataWrapper	24,42
Tableau Public	24,13
ChartIO	24,13
Datavisual	23,45

Development User Interface for IoT Framework Category

Geo Dashboard	**36,27**
FIWARE Advanced Web Based User Interface	34,20
Kaa SDK	33,63
Spagobi	33,77
SAS	32,27
Pentaho Internet of Things Analytics	31,99
Cumulocity Web SDK	31,99
Information Builders	31,42
Tableau	31,13
Power BI	31,02
Amazon QuickSight	30,42
Dashboard Server	29,42
Qlik	27,85

Development GIS Data Visualization Category

Leaflet	**31,92**
Open Layers	**31,92**
Google Maps API	**31,92**
Cesium	**31,92**
ArcGIS Javascript JS	**31,92**
HERE	23,88
CARTO	24,88

This is the summary of the obtained results after going through all the process of applying the proposed methodology for the different tools and criteria. After having conducted this exercise of analysis, some comments regarding the experience should be mentioned:

- ✓ The evaluation process of all tools has got difficulties as a wide range of tools has been reviewed, even though in many cases they do not even fit with the proposed criteria. For instance, when it comes to libraries, maintenance or security criteria are not applied since these do depend on the final product that is being developed, or they are achieved by including external packages.

- ✓ The criteria that can be crucial at the moment of selecting a tool is to take into account the previous knowledge, background and experience that the user has on it. This criterion is not considered in the proposed methodology, and in that case, the learning curve should not be applied in any case.

- ✓ Another issue is related to the licensing cost, as many times, even though the tool needs to be licensed, it may be not be applied since it is already considered in other projects' phases. In this case, the cost criteria should not been taken into account.

- ✓ There are criteria that it has been complex to assign the punctuations due to there are not good or bad options. The multiplatform criteria could be a valid example of that. In this case it is not better to have only Windows support, or Linux, or Mac. These options are related to the requirements of the final solution.

- ✓ In relation to the previous comment, regarding the criteria which evaluations are not good or bad, but different, some transformations have to be done when defining how to evaluate. The output of these transformations should be criteria that can be evaluated in terms of "levels of compliance".

- ✓ For YES/NO criteria, instead of transforming it into levels (in order to punctuate the technologies according to it) it has been mapped a value of 3 for YES and 0 for NO. In this case there are no intermediate evaluations. For these criteria, like reports creation, It is only considered that a technology meets the criteria or not.

- ✓ Regarding the obtained results, there are categories with a low variation in the final scores. This is due to the maturity level of all the tools identified and evaluated on a category. This causes that every reviewed tool accomplishes almost every criterion in the same way and there exists very small variations among them.

As it is possible to note in these previous comments, the casuistry is huge and every tool and criteria can be read into many ways. All of this, adding the tools wide typology makes scoring them much harder.

6 Conclusions, deviations and outputs for other WPs

The document presents a new guide for designing applications and services for SmartEnCity based on ICT technologies. This design guide focuses on different aspects such us user experience, technology features and characteristics.

In addition, a catalog of specific tools has been identified to implement the described methodology. This catalog is structured in the three phases defined by the methodology. A new list of technologies including methodologies, libraries and tools has been identified. The list is not intended to be exhaustive of all existing, but a significant sample of the main types found.

Finally, a methodological proposal has been defined for the selection and evaluation of technologies for the design and development of Smart City solutions based on IoT. This proposal is based on an existing methodology, already used in the processes of technology selection, adapted to the scenario proposed in this task. The selection process, the selection criteria for each of the phases, as well as guidelines for its application are proposed in this deliverable. Information included in this deliverable represents the framework for the selection of technologies and tools for a specific project. The number of technologies identified, as well as the different interpretations that can be made of the chosen criteria makes it necessary to contextualize the results to a specific project. The presented use case represents an example of application of the methodology that should be particularized to a reduced number of technologies and an interpretation of the criteria that will be defined by the context of each project.

No deviations have been produced according to the dates and content of the deliverable with respect to the proposed plan.

The outputs produced in this deliverable will have effects mainly on other activities of the WP6 and on activities related with the deployment of the CIOP platform in the three lighthouse cities (Vitoria-Gasteiz WP3, Tartu WP4 and Sonderborg WP5)

7 References

Ahmed Seffah, Jan Gulliksen, Michael C. Desmarais. Human-centered software engineering -integrating usability in the software development lifecycle . s.l. : Springer. Vol. 8.

An efficient and simple model for multiple criteria supplier selection problem. Wan Lung Ng. European Journal of Operational Research 186 (2008).

Ashworth, C. M. (1988). Structured systems analysis and design method (SSADM). Information and Software Technology, 30(3), 153-163.

Assessing the quality of different MCDA methods. Andrea de Montis, Pasquale De Toro, Bert Droste-Franke, Ines Omann, Sigrid Stagl.
http://people.unica.it/adm/files/2008/11/05_de_monti_et_al.pdf

Bertholdo, A. P. O., da Silva, T. S., Melo, C. D. O., Kon, F., & Silveira, M. S. (2014, June). Agile usability patterns for UCD early stages. In International Conference of Design, User Experience, and Usability (pp. 33-44). Springer, Cham.

Bhrel, Manuel. Exploring principles of user-centered agile software development: A literature review. s.l. : Elsevier, 2015, Vol. 61, págs. 63-181. doi: http://dx.doi.org/10.1016/j.infsof.2015.01.004

Blomkvist, Stefan. Towards a model for bridging agile development and user-centered design. s.l. : Human-centered software engineering—integrating usability in the software development lifecycle. Springer Netherlands, 2005. p. 219-244. doi: http://dx.doi.org/10.1007/1-4020-4113-6_12

Brown, Tim. Design Thinking. s.l. : Harvard Business Review, 2008.

Chamberlain, S., Sharp, H., & Maiden, N. (2006). Towards a framework for integrating agile development and user-centred design. Extreme programming and agile processes in software engineering, 143-153.

Cohen, David; Lindvall, Mikael; Costa, Patricia. Agile software development. DACS Soar Report, 2003, vol. 11.

Da Silva, T. S., Silveira, M. S., Maurer, F., & Hellmann, T. (2012). User experience design and agile development: from theory to practice. Journal of Software Engineering and Applications, 5(10), 743.

Daae, Johannes. A classification of user research methods for design for sustainable behaviour. s.l. : Elsevier, 2014, Journal of Cleaner Production journal, Vol. 106, págs. 680-689. doi: http://dx.doi.org/10.1016/j.jclepro.2014.04.056

DIS, ISO. 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. International Standardization Organization (ISO). Switzerland, 2009.

Escalona, M. J., y Koch, N. Ingeniería de Requisitos en Aplicaciones para la Web–Un estudio comparativo. Universidad de Sevilla. 2002

Fauquex, M., Goyal, S., Evequoz, F., & Bocchi, Y. (2015, December). Creating people-aware IoT applications by combining design thinking and user-centered design methods. In Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on (pp. 57-62). IEEE.

Fauquex, Milène, et al. Creating people-aware IoT applications by combining design thinking and user-centered design methods.. En Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on. IEEE, 2015. p. 57-62. doi: http://dx.doi.org/10.1109/WF-IoT.2015.7389027

Fischer, G. (2012, May). Context-aware systems: the'right'information, at the'right'time, in the'right'place, in the'right'way, to the'right'person. In Proceedings of the international working conference on advanced visual interfaces (pp. 287-294). ACM.

Fischer, Gerhard. Context-Aware Systems: The 'Right' Information, at the 'Right' Time, in the 'Right' Place, in the 'Right' Way, to the 'Right' Person. s.l. : ACM, 2012. International Working Conference on Advanced Visual Interfaces 2012. págs. 287-294. doi: http://dx.doi.org/10.1145/2254556.2254611

Fox, David; Sillito, Jonathan; Maurer, Frank. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. En Agile, 2008. AGILE'08. Conference. IEEE, 2008. p. 63-72. doi: 10.1109/Agile.2008.78

Gorecky, Dominic. Human-machine-interaction in the industry 4.0 era. s.l. : IEEE, 2014. doi: http://dx.doi.org/10.1109/INDIN.2014.6945523

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., & Cajander, Å. (2003). Key principles for user-centred systems design. Behaviour and Information Technology, 22(6), 397-409.

Gulliksen, Jan et al. Key principles for user-centred systems design. Behaviour and Information Technology, 2003, vol. 22, no 6, p. 397-409. doi: http://dx.doi.org/10.1080/01449290310001624329

Guo, H. (2016, July). Lean but not Mean UX: Towards a Spiral UX Design Model. In International Conference of Design, User Experience, and Usability (pp. 25-33). Springer International Publishing.

He, J., King W. (2008). The Role of User Participation in Information Systems Development: Implications from a Meta-Analysis. Journal of Management Information Systems archive Volume 25 Issue 1, Number 1. Pages 301-331

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. Computer, 34(9), 120-127.

Highsmith, Jim. Agile software development: the business of innovation. 9, s.l. : IEEE, 2001, Vol. 34, págs. 120-127. doi: http://dx.doi.org/10.1109/2.947100

Holzinger, A. (2005). Usability engineering methods for software developers. Communications of the ACM, 48(1), 71-74.

IEEE-Std.'610' (1990) IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society Press.

InVision. Invisionapp.. [En línea] 2017. www.invisionapp.com

Jackson, M. A. (1982). A system development method. Tools and notions for program construction: An advanced course, 1-25.

Jäger, Andreas. Industry 4.0 : challenges for the human factor in future production scenarios. Stockholm : 4th Conference on Learning Factories, 2014.

Kaasbøll, J. (2007) Explaining the role of user participation in information system use. Department of Informatics, University Oslo. Available at: http://heim.ifi.uio.no/~jensj/INF3270/Involvement.pdf

Kashfi, P., Feldt, R., Nilsson, A., & Svensson, R. B. (2016, August). A Conceptual UX-Aware Model of Requirements. In International Conference on Human-Centred Software Engineering (pp. 234-245). Springer International Publishing.

Kujala, S., Kauppinen, M., Lehtola, L., & Kojo, T. (2005, August). The role of user involvement in requirements quality and project success. In Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on (pp. 75-84). IEEE.

Lee, Jay. Service innovation and smart analytics for Industry 4.0 and big data environment. Cincinnati : Elsevier, 2014, Vol. 16. doi: http://dx.doi.org/10.1016/j.procir.2014.02.001

Lojka, T. Web technologies in industry HMI. s.l. : IEEE, 2015. Intelligent Engineering Systems (INES), 2015 IEEE 19th International Conference on. doi: 10.1109/INES.2015.7329647

MCDM Techniques Selection Approaches: State of the Art. E. Kornyshova, C. Salinesi. Proceedings of the 2007 IEEE Symposium on Computational. Intelligence in Multicriteria Decision Making (MCDM 2007).

Multi-criteria decision analysis for use in transport decision making, DTU Transport. Compendium Series part 2, 2014.
http://miroslawdabrowski.com/downloads/MoV/The%20Simple%20Multi%20Attribute%20Rating%20Technique%20%28SMART%29.pdf

Muratza, M., Shwan, A. (2009) Guidelines for Multilingual Software Development. Chalmers University of Technology. University of Gothenburg. Department of Computer Science and Engineering. Available at: http://publications.lib.chalmers.se/records/fulltext/161183.pdf

Nourse, W. (2014) Technology: Understanding the Total Cost of Ownership. Available at: https://www.salesforce.com/blog/2014/04/technology-cost-ownership-gp.html

Oliveira, Ana Paula. Agile Usability Patterns for User-Centered Design Final Stages. [aut. libro] Masaaki Kurosu. Human-Computer Interaction. Theory, Design, Development and Practice. s.l. : Springer, 2016. doi: http://dx.doi.org/10.1007/978-3-319-39510-4_40

Pappas, C. (2015). eLearning Interactivity: The Ultimate Guide For eLearning Professionals. Available at: https://elearningindustry.com/elearning-interactivity-the-ultimate-guide-for-elearning-professionals

Pfeiffer, T., Hellmers, J., Schön, E. M., & Thomaschewski, J. (2016). Empowering user interfaces for Industrie 4.0. Proceedings of the IEEE, 104(5), 986-996.

Pfeiffer, Thies. Empowering User Interfaces for Industrie 4.0. 5, s.l. : IEEE, 2016, Proceedings of the IEEE, Vol. 104, págs. 986-996. doi: http://dx.doi.org/10.1109/JPROC.2015.2508640

Priestley, O. (2015). User experience methodology: from the physical to the emotional. Learned Publishing, 28(4), 317-320.

Roberts, Jonathan C. Sketching Designs Using the Five Design-Sheet Methodology. 1, s.l. : IEEE, 2016, Vol. 22, págs. 419-428. doi: http://dx.doi.org/10.1109/TVCG.2015.2467271

Ross, L. (2010), “Virtual 3D City Models in Urban Land Management Technologies and Applications”, Master thesis, the University of Berlin, Germany, 2010

Saffer, Dan. Microinteractions: Designing with details. s.l. : O'Reilly, 2013.

Shen, Y. C., Chang, S. H., Lin, G. T., & Yu, H. C. (2010). A hybrid selection model for emerging technology. Technological Forecasting and Social Change, 77(1), 151-166.

Singh, M. (2008, August). U-SCRUM: An agile methodology for promoting usability. In Agile, 2008. AGILE'08. Conference (pp. 555-560). IEEE.

Unibertsitatea, Mondragon. Metodología de Innovación Centrada en las Personas. 2014.

Wehle, Hans-Dieter; IBM. Augmented Reality and the Internet of Things (IoT) / Industry 4.0. 2016.

Wright, P., Wallace, J., & McCarthy, J. (2008). Aesthetics and experience-centered design. ACM Transactions on Computer-Human Interaction (TOCHI), 15(4), 18.

8 ANNEX

This Annex provides all the detailed information that has been created within the Task 6.5 of the project SmartEnCity. The main content of this Annex is focused on the selection of tools and technologies, and its evaluation tables, which has been fulfilled by experts.

8.1 Understanding phase criteria evaluation

At this section the evaluation for the Understanding phase tools are detailed (table x).

Phase	Criteria	Importance	Tool	Evaluation
Understanding	User participation	4	Trendwatching	1
			Cultural probe	2
			User interviews	3
			User expert interviews	3
			Focus groups	3
			Empathy Map	3
			Shadowing	3
			Questionnaires	3
			Experience map	2
			Personas	3
			Actor map	2
	Ease of access / use	3	Trendwatching	2
			Cultural probe	2
			User interviews	3
			User expert interviews	3
			Focus groups	3
			Empathy Map	2
			Shadowing	3
			Questionnaires	3
			Experience map	2
			Personas	2
			Actor map	2
	Feedback capability	3	Trendwatching	1
			Cultural probe	2
			User interviews	2
			User expert interviews	3
			Focus groups	3
			Empathy Map	3
			Shadowing	3
			Questionnaires	3

			Experience map	3
			Personas	3
			Actor map	3
	Technical skills needed	2	Trendwatching	2
			Cultural probe	2
			User interviews	3
			User expert interviews	3
			Focus groups	3
			Empathy Map	2
			Shadowing	3
			Questionnaires	3
			Experience map	2
			Personas	2
			Actor map	2
	Cost	3	Trendwatching	1
			Cultural probe	2
			User interviews	3
			User expert interviews	2
			Focus groups	3
			Empathy Map	2
			Shadowing	3
			Questionnaires	3
			Experience map	2
			Personas	2
			Actor map	2

Table 7. Understanding phase

8.2 Design phase criteria evaluation

At this section the evaluation for the design phase tools and technologies are detailed (table x).

Phase	Criteria	Importance	Tool	Evaluation
Design	**License (price)**	3	Brainstorming	3
			Metaphors and analogues	3
			Sketching	3
			Storyboard	3

			Co-creation	2
			Early prototyping	2
			Role-playing	3
			Information architecture map	2
			Wireframes	3
			Balsamiq Mockups	2
			Draw.io	3
			Bpmn.io	3
			Moqups	2
			justinmind	2
	Desktop vs Web	3	Brainstorming	1
			Metaphors and analogues	1
			Sketching	1
			Storyboard	1
			Co-creation	1
			Early prototyping	0
			Role-playing	0
			Information architecture map	1
			Wireframes	1
			Balsamiq Mockups	1
			Draw.io	1
			Bpmn.io	1
			Moqups	1
			justinmind	1
	Life preview	2	Brainstorming	0
			Metaphors and analogues	0
			Sketching	0
			Storyboard	0
			Co-creation	0
			Early prototyping	2
			Role-playing	0
			Information architecture map	0
			Wireframes	2
			Balsamiq Mockups	2
			Draw.io	1
			Bpmn.io	1
			Moqups	2
			justinmind	2
	Multiplatform	3	Brainstorming	1
			Metaphors and analogues	1
			Sketching	1

			Storyboard	1
			Co-creation	1
			Early prototyping	1
			Role-playing	1
			Information architecture map	1
			Wireframes	1
			Balsamiq Mockups	1
			Draw.io	1
			Bpmn.io	1
			Moqups	1
			justinmind	1

Table 8. Design phase

8.3 Development phase criteria evaluation

At this section, the evaluation for the development phase tools and technologies is detailed. Which is divided into 3 main criteria sections: administration (table x), technical (table x) and results and visualization criteria (table x).

Phase	Criteria		Importance	Tool		Evaluation
	Administration	License (price)	3	Usability and User Experience	Usability Test	2
					Heuristic evaluation	2
					Eyetracking	1
					Facereader	1
					Questionnaire	3
					Shadowing	3
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	2
					ChartBlocks	1
					DataWrapper	1
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	1
					Polycharts	3
					Fusioncharts	1
					Highcharts JS	1
					Tableau Public	1
					ChartIO	1
				User interface for IoT framework	Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	1
					Power BI	2
					Tableau	1
					Qlik	1
					SAS	1
					Information Builders	1
					Amazon QuickSight	1
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	1
					Kaa SDK	3
				Tools and libraries	Leaflet	3
					Open Layers	3
					Google Maps API	3

					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2
		Installation Requirements	1	Usability and User Experience	Usability Test	0
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	0
					Shadowing	0
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	3
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	3
					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	3
					ChartIO	3
				User interface for IoT framework	Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3

					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	3
					HERE	3
				Usability and User Experience	Usability Test	3
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	3
					Shadowing	0
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
		Multilingual **0 – N/A** **1 – Monolingual** **3 - Multilingual**	1	Visualization library	Datavisual	3
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	3
					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	3
					ChartIO	3
				ace for IoT fram	Dashboard Server	3
					Geo Dashboard	3

					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	3
					HERE	3

Table 9. Administration

Phase	Criteria		Importance	Tool		Evaluation
	Technical	Documentation	3	Usability and User Experience	Usability Test	2
					Heuristic evaluation	2
					Eyetracking	2
					Facereader	2
					Questionnaire	3
					Shadowing	2
				Web Application Framework	Python Django	3
					Ruby Rails	2
					Java Spring	3
					PHP Symfony	2
					Jaggery.js	2
					AngularJS	3
				Visualization library	Datavisual	1
					ChartBlocks	2
					DataWrapper	1
					Bokeh	2
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	2
					Dc.js	2
					Teechart	3
					Polycharts	2
					Fusioncharts	3
					Highcharts JS	2
					Tableau Public	1
					ChartIO	2
				User interface for IoT framework	Dashboard Server	1
					Geo Dashboard	2
					Pentaho Internet of Things Analytics	1
					Power BI	1
					Tableau	1
					Qlik	1
					SAS	2
					Information Builders	1
					Amazon QuickSight	1
					Spagobi	2
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	2
					Kaa SDK	3
				Tools and libraries	Leaflet	3
					Open Layers	3
					Google Maps API	3

					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2
		Learning curve	2	Usability and User Experience	Usability Test	2
					Heuristic evaluation	1
					Eyetracking	2
					Facereader	2
					Questionnaire	3
					Shadowing	2
				Web Application Framework	Python Django	3
					Ruby Rails	2
					Java Spring	2
					PHP Symfony	2
					Jaggery.js	2
					AngularJS	2
				Visualization library	Datavisual	2
					ChartBlocks	2
					DataWrapper	2
					Bokeh	2
					Seaborn	2
					Google Chart	2
					D3.js	3
					C3.js	2
					Dc.js	3
					Teechart	2
					Polycharts	2
					Fusioncharts	3
					Highcharts JS	2
					Tableau Public	3
					ChartIO	3
				User interface for IoT framework	Dashboard Server	2
					Geo Dashboard	2
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	2
					SAS	2
					Information Builders	2
					Amazon QuickSight	2

					Spagobi	2
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	2
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2
		Scalability	3	Usability and User Experience	Usability Test	0
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	0
					Shadowing	0
				Web Application Framework	Python Django	2
					Ruby Rails	2
					Java Spring	2
					PHP Symfony	2
					Jaggery.js	2
					AngularJS	2
				Visualization library	Datavisual	1
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	2
					Dc.js	2
					Teechart	2
					Polycharts	2
					Fusioncharts	2
					Highcharts JS	3
					Tableau Public	3
					ChartIO	2
				ace for IoT fram	Dashboard Server	3
					Geo Dashboard	3

					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	2
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	3
					HERE	3
		Maintainability	2	Usability and User Experience	Usability Test	3
					Heuristic evaluation	3
					Eyetracking	3
					Facereader	3
					Questionnaire	3
					Shadowing	3
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	3
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	3

					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	3
					ChartIO	3
				User interface for IoT framework	Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	3
					HERE	3
					Leaflet	3
				Usability and User Experience	Usability Test	3
					Heuristic evaluation	3
					Eyetracking	3
					Facereader	3
					Questionnaire	3
					Shadowing	3
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	3
					ChartBlocks	3

		Security and data privacy	3			
				User interface for IoT framework	DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	3
					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	3
					ChartIO	3
					Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	3
					HERE	3
		Multiplatform		Usability and User Experience	Usability Test	3
					Heuristic evaluation	3
					Eyetracking	3
					Facereader	3
					Questionnaire	3
					Shadowing	3
				a t i o n	Python Django	3

					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	3
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
			3		Dc.js	3
					Teechart	3
					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	2
					ChartIO	2
				User interface for IoT framework	Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	3
					Power BI	1
					Tableau	2
					Qlik	1
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2

		Level of interactivity	1	Usability and User Experience	Usability Test	3
					Heuristic evaluation	3
					Eyetracking	3
					Facereader	3
					Questionnaire	3
					Shadowing	3
				Web Application Framework	Python Django	1
					Ruby Rails	1
					Java Spring	1
					PHP Symfony	1
					Jaggery.js	1
					AngularJS	1
				Visualization library	Datavisual	2
					ChartBlocks	2
					DataWrapper	2
					Bokeh	2
					Seaborn	2
					Google Chart	2
					D3.js	2
					C3.js	2
					Dc.js	2
					Teechart	2
					Polycharts	2
					Fusioncharts	2
					Highcharts JS	2
					Tableau Public	2
					ChartIO	2
				User interface for IoT framework	Dashboard Server	2
					Geo Dashboard	2
					Pentaho Internet of Things Analytics	2
					Power BI	2
					Tableau	2
					Qlik	2
					SAS	2
					Information Builders	2
					Amazon QuickSight	2
					Spagobi	2
					FIWARE Advanced Web Based User Interface	2
					Cumulocity Web SDK	2
					Kaa SDK	2

				Tools and libraries for GIS data visualization	Leaflet	1
					Open Layers	1
					Google Maps API	1
					Cesium	1
					ArcGIS Javascript JS	1
					CARTO	1
					HERE	1

Table 10. Technical issues criteria

Phase	Criteria		Importance	Tool		Evaluation
	Results and visualization	Real time data visualization	3	Usability and User Experience	Usability Test	1
					Heuristic evaluation	1
					Eyetracking	1
					Facereader	1
					Questionnaire	1
					Shadowing	1
				Web Application Framework	Python Django	3
					Ruby Rails	3
					Java Spring	3
					PHP Symfony	3
					Jaggery.js	3
					AngularJS	3
				Visualization library	Datavisual	3
					ChartBlocks	3
					DataWrapper	3
					Bokeh	3
					Seaborn	3
					Google Chart	3
					D3.js	3
					C3.js	3
					Dc.js	3
					Teechart	3
					Polycharts	3
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	3
					ChartIO	3
				...ace for IoT fram...	Dashboard Server	3
					Geo Dashboard	3

					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2
		3D visualization	1	Usability and User Experience	Usability Test	0
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	0
					Shadowing	0
				Web Application Framework	Python Django	2
					Ruby Rails	2
					Java Spring	2
					PHP Symfony	2
					Jaggery.js	2
					AngularJS	2
				Visualization library	Datavisual	1
					ChartBlocks	3
					DataWrapper	1
					Bokeh	1
					Seaborn	1
					Google Chart	3
					D3.js	3
					C3.js	1
					Dc.js	1
					Teechart	3

					Polycharts	1
					Fusioncharts	3
					Highcharts JS	3
					Tableau Public	1
					ChartIO	1
				User interface for IoT framework	Dashboard Server	
					Geo Dashboard	
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	1
					SAS	3
					Information Builders	3
					Amazon QuickSight	
					Spagobi	3
					FIWARE Advanced Web Based User Interface	1
					Cumulocity Web SDK	3
					Kaa SDK	1
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	2
					HERE	2
		Geolocation	3	Usability and User Experience	Usability Test	0
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	0
					Shadowing	0
				Web Application Framework	Python Django	1
					Ruby Rails	1
					Java Spring	1
					PHP Symfony	1
					Jaggery.js	1
					AngularJS	1
				Visualization library	Datavisual	1
					ChartBlocks	1
					DataWrapper	1

					Bokeh	1
					Seaborn	1
					Google Chart	1
					D3.js	1
					C3.js	1
					Dc.js	1
					Teechart	1
					Polycharts	1
					Fusioncharts	1
					Highcharts JS	1
					Tableau Public	1
					ChartIO	1
				User interface for IoT framework	Dashboard Server	1
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	1
					Power BI	1
					Tableau	1
					Qlik	1
					SAS	1
					Information Builders	1
					Amazon QuickSight	1
					Spagobi	1
					FIWARE Advanced Web Based User Interface	1
					Cumulocity Web SDK	1
					Kaa SDK	1
				Tools and libraries for GIS data visualization	Leaflet	3
					Open Layers	3
					Google Maps API	3
					Cesium	3
					ArcGIS Javascript JS	3
					CARTO	1
					HERE	1
		Dashboard creation		Usability and User Experience	Usability Test	0
					Heuristic evaluation	0
					Eyetracking	0
					Facereader	0
					Questionnaire	0
					Shadowing	0
				Application Fram	Python Django	1
					Ruby Rails	1

			3		Java Spring	1
					PHP Symfony	1
					Jaggery.js	1
					AngularJS	1
				Visualization library	Datavisual	1
					ChartBlocks	1
					DataWrapper	1
					Bokeh	1
					Seaborn	1
					Google Chart	1
					D3.js	1
					C3.js	1
					Dc.js	1
					Teechart	1
					Polycharts	1
					Fusioncharts	1
					Highcharts JS	1
					Tableau Public	1
					ChartIO	1
				User interface for IoT framework	Dashboard Server	3
					Geo Dashboard	3
					Pentaho Internet of Things Analytics	3
					Power BI	3
					Tableau	3
					Qlik	3
					SAS	3
					Information Builders	3
					Amazon QuickSight	3
					Spagobi	3
					FIWARE Advanced Web Based User Interface	3
					Cumulocity Web SDK	3
					Kaa SDK	3
				Tools and libraries for GIS data visualization	Leaflet	1
					Open Layers	1
					Google Maps API	1
					Cesium	1
					ArcGIS Javascript JS	1
					CARTO	1
					HERE	1
		Reports creation		d U s e	Usability Test	1

			3		
				Heuristic evaluation	1
				Eyetracking	1
				Facereader	1
				Questionnaire	1
				Shadowing	1
				Python Django	1
				Ruby Rails	1
				Java Spring	1
				PHP Symfony	1
				Jaggery.js	1
				AngularJS	1
				Datavisual	1
				ChartBlocks	1
				DataWrapper	1
				Bokeh	1
				Seaborn	1
				Google Chart	1
				D3.js	1
				C3.js	1
				Dc.js	1
				Teechart	1
				Polycharts	1
				Fusioncharts	1
				Highcharts JS	1
				Tableau Public	1
				ChartIO	1
				Dashboard Server	3
				Geo Dashboard	3
				Pentaho Internet of Things Analytics	3
				Power BI	3
				Tableau	3
				Qlik	3
				SAS	3
				Information Builders	3
				Amazon QuickSight	3
				Spagobi	3
				FIWARE Advanced Web Based User Interface	3
				Cumulocity Web SDK	3
				Kaa SDK	3
				Leaflet	1

Web Application Framework

Visualization library

User interface for IoT framework

Tools

					Open Layers	1
					Google Maps API	1
					Cesium	1
					ArcGIS Javascript JS	1
					CARTO	1
					HERE	1
		Scenario analysis		Usability and User Experience	Usability Test	1
					Heuristic evaluation	1
					Eyetracking	1
					Facereader	1
					Questionnaire	1
					Shadowing	1
				Web Application Framework	Python Django	1
					Ruby Rails	1
					Java Spring	1
					PHP Symfony	1
					Jaggery.js	1
					AngularJS	1
				Visualization library	Datavisual	1
					ChartBlocks	1
					DataWrapper	1
					Bokeh	1
					Seaborn	1
					Google Chart	1
					D3.js	1
					C3.js	1
					Dc.js	1
					Teechart	1
					Polycharts	1
					Fusioncharts	1
					Highcharts JS	1
					Tableau Public	1
					ChartIO	1
				User interface for IoT framework	Dashboard Server	1
					Geo Dashboard	1
					Pentaho Internet of Things Analytics	1
					Power BI	1
					Tableau	1
					Qlik	1
					SAS	1

					Information Builders	1
					Amazon QuickSight	1
					Spagobi	1
					FIWARE Advanced Web Based User Interface	1
					Cumulocity Web SDK	1
					Kaa SDK	1
				Tools and libraries for GIS data visualization	Leaflet	1
					Open Layers	1
					Google Maps API	1
					Cesium	1
					ArcGIS Javascript JS	1
					CARTO	1
					HERE	1

Table 11. Results and visualization criteria