



simpli-city

The Road User Information System Of The Future

WP7 – Use Case I: Meeting the Increased Mobility Demand

D7.3 Evaluation Report

Document Lead: FGM

Contributing Partners: TUV, ASC, TIE, TUDA, IBMIL, TALK, WORLD, SRM, CRF

Delivery Date: 10/2015

Dissemination Level: Public

Version 1.0

Short Abstract (Teaser)

This document provides an overview of the evaluation approach taken within SIMPLI-CITY, and describes the results of the WP7 related evaluation activities conducted within the project.



| Document Status | |
|----------------------------|---|
| Deliverable Lead | Michaela Kargl, FGM |
| Internal Reviewer 1 | Philipp Hoenisch, TUV |
| Internal Reviewer 2 | Daniel Burgstahler, TUDA |
| Type | Public Deliverable |
| Work Package | WP7 Use Case I: Meeting the Increased Mobility Demand |
| ID | D7.3 Evaluation Report |
| Due Date | 30.09.2015 |
| Delivery Date | 09.11.2015 |
| Status | For Approval |

| Document History | |
|----------------------|---|
| Contributions | <p>V0.1, Michaela Kargl (FGM), 02.09.2015, Document structure.</p> <p>V0.2, Michaela Kargl (FGM), 07.09.2015, added chapter 2, chapter 3 and related chapter 6</p> <p>V0.3, Markus Garger (FGM), 08.09.2015, added chapter 4 and related chapters 7,8,9,10</p> <p>V0.4, Michaela Kargl (FGM), 15.09.2015, added chapter 5.1 and related chapters 11, 12</p> <p>V0.5, Michaela Kargl (FGM), 30.09.2015, added chapter 5.2 and related chapters 13, 14, as well as the summary</p> <p>V1.0, Michaela Kargl (FGM), 15.10.2015, updated according to internal reviewers' comments</p> |
| Final Version | October 30, 2015 (M37) |

Note

This deliverable is subject to final acceptance by the European Commission.

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

| | | | | |
|---|-----------------------|--|----------------------|---------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 2 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Project Partners



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Vienna University of Technology (Coordinator),
Austria



Ascora GmbH, Germany



TIE Nederland B.V., The Netherlands



Technische Universität Darmstadt, Germany



IBM Research – Ireland
Smarter Cities Technology Centre



Forschungsgesellschaft Mobilität, Austria



Talkamatic AB, Sweden



Atos Worldline, Spain



Centro Ricerche FIAT, Italy



SRM – Reti e Mobilità, Italy

Executive Summary

The evaluation strategy followed within SIMPLI-CITY combined elements of technical evaluation, usability evaluation, and functional evaluation, that are all three combined within this report. The Technical Evaluation describes the grade of fulfilment of the requirements specified within deliverable D2.3 at the beginning of the project. Usability Evaluation and Functional Evaluation describe the usability and the functionality of the system from users' point of view. However, the SIMPLI-CITY system addresses two user groups: software developers and road users. Hence, the parts of SIMPLI-CITY that are relevant for software developers and the parts of SIMPLI-CITY that are relevant for road users were evaluated separately.

Within the SIMPLI-CITY project, evaluation was assigned to the two tasks T7.3 and T8.3 responsible for the evaluation of Use Case I and Use Case II related parts of the SIMPLI-CITY system respectively. While the evaluation of the Use Case I and Use Case II related SIMPLI-CITY Apps could easily be assigned to the relevant work packages, the evaluation activities related to the other parts of the SIMPLI-CITY system were relevant for both Use Cases. Therefore, the evaluation of SIMPLI-CITY's market places and SIMPLI-CITY's developer tools as well as the Technical Evaluation of the whole SIMPLI-CITY system were done in close cooperation of both evaluation tasks T7.3 and T8.3.

This document includes a detailed description and the results of the common evaluation and the evaluation of the Use Case I related SIMPLI-CITY App.

The Technical Evaluation of the SIMPLI-CITY prototypes showed that all 115 "Must Have" requirements, which were defined during the specification phase at the beginning of the project, were fulfilled, whereby over 86% of these requirements were completely fulfilled. In addition, of the 38 "Should Have" requirements, about 74% were completely fulfilled.

The Usability Evaluation was seen as formative evaluation: in order to allow the SIMPLI-CITY developers to implement further improvements to the system based on the findings of the evaluation, Usability Evaluation was conducted in phases starting from June 2015 to September 2015. For all components of the SIMPLI-CITY system several useful hints for improvement of their usability were obtained by the usability evaluation. Many of these improvements suggested by the results of the usability evaluation were implemented by the SIMPLI-CITY developers immediately. Some further improvements of the WP7 related App were even done in the first weeks of October 2015. However, of course not all of the usability issues could be solved during the runtime of the project, but the remaining usability issues are well documented and constitute valuable contributions for further developments of the SIMPLI-CITY prototypes.

Functional Evaluation confirmed that the SIMPLI-CITY prototypes are functioning in real usage. As an additional outcome of the Functional Evaluation several suggestions for practically useful enhancements of the functionality of the prototypes were obtained, which will be helpful for further development of the SIMPLI-CITY system.

Table of Contents

| | | |
|-------|--|----|
| 1 | Introduction | 8 |
| 1.1 | SIMPLI-CITY Project Overview | 8 |
| 1.2 | Deliverable Purpose, Scope and Context | 9 |
| 1.3 | Document Status and Target Audience | 9 |
| 1.4 | Abbreviations and Glossary | 9 |
| 1.5 | Document Structure | 9 |
| 2 | SIMPLI-CITY's Evaluation Approach | 10 |
| 3 | Results of the Technical Evaluation | 12 |
| 4 | Results of the Evaluation from Developers' Point of View | 16 |
| 4.1 | Results of the Usability Evaluation by Test-Users | 16 |
| 4.1.1 | Results of the Usability Evaluation of SIMPLI-CITY's Developer Tools | 16 |
| 4.1.2 | Results of the Usability Evaluation of the Developers' Part of the SIMPLI-CITY App Marketplace | 17 |
| 4.2 | Results of the Usability Inspection by Experts | 17 |
| 4.3 | Results of the Functional Evaluation from Developers' Point of View | 18 |
| 5 | Results of the Evaluation from Road Users' Point of View | 19 |
| 5.1 | Results of the Evaluation of the Road Users' Part of the App Marketplace | 19 |
| 5.1.1 | Results of the Usability Inspection by Experts | 19 |
| 5.1.2 | Results of the Usability Evaluation by Test-Users | 20 |
| 5.2 | Results of the Evaluation of the Use Case 7 related App | 20 |
| 5.2.1 | Results of the Usability Inspection by Experts | 20 |
| 5.2.2 | Results of the Usability Evaluation by Test-Users | 21 |
| 6 | Detailed Results of the Technical Evaluation | 22 |
| 7 | Usability Evaluation of SIMPLI-CITY's Developer Tools | 42 |
| 7.1 | Approach taken for the Usability Evaluation | 42 |
| 7.1.1 | Aim of the Usability Evaluation | 42 |
| 7.1.2 | Method Applied | 42 |
| 7.1.3 | Team of Facilitators | 42 |
| 7.1.4 | Technical Setup for the Usability Evaluation | 42 |
| 7.1.5 | Usability Evaluation Sessions | 43 |
| 7.1.6 | Test-Users | 43 |
| 7.1.7 | Evaluation Tasks | 45 |
| 7.2 | Summary of the Results of the Usability Evaluation | 45 |
| 7.3 | Detailed Documentation of the Findings | 46 |
| 7.3.1 | Usability Metrics | 46 |
| 7.3.2 | Main Positive Aspects Mentioned by the Test-Users | 49 |
| 7.3.3 | Main Usability Issues Revealed During the Evaluation | 50 |
| 7.4 | Task Sheets for Usability Evaluation of SIMPLI-CITY's Developer Tools | 56 |
| 7.5 | Transcription of the Debriefing Interviews | 59 |
| 8 | Usability Evaluation of the Developers' Part of the App Marketplace | 64 |
| 8.1 | Approach taken for the Usability Evaluation | 64 |
| 8.1.1 | Aim of the Usability Evaluation | 64 |
| 8.1.2 | Method Applied | 64 |
| 8.1.3 | Team of Facilitators | 64 |
| 8.1.4 | Usability Evaluation Sessions | 65 |
| 8.1.5 | Test-Users | 65 |

| | | |
|--------|--|-----|
| 8.1.6 | Evaluation Tasks..... | 66 |
| 8.2 | Summary of the Results of the Usability Evaluation | 67 |
| 8.3 | Detailed Documentation of the Findings..... | 68 |
| 8.3.1 | Usability Metrics..... | 68 |
| 8.3.2 | Main Positive Aspects Mentioned by the Test-Users | 70 |
| 8.3.3 | Main Usability Issues Revealed During the Evaluation | 70 |
| 8.4 | Task Sheets for Usability Evaluation of the App Marketplace from Developers' Point of View | 74 |
| 8.5 | Transcription of the Debriefing Interviews | 76 |
| 9 | Usability Inspection from Developers' Point of View | 78 |
| 9.1 | Approach taken for the Usability Inspection | 78 |
| 9.1.1 | Aim of the Usability Inspection | 78 |
| 9.1.2 | Method Applied | 78 |
| 9.1.3 | Team of Evaluators..... | 78 |
| 9.1.4 | Usability Inspection Sessions..... | 79 |
| 9.1.5 | Context Information..... | 79 |
| 9.2 | Summary of the Results of the Usability Inspection..... | 80 |
| 9.3 | Detailed Documentation of the Findings..... | 80 |
| 9.3.1 | Task 1: Develop an App Using the SIMPLI-CITY System..... | 81 |
| 9.3.2 | Task 2: Develop a Service Using the SIMPLI-CITY System | 85 |
| 9.3.3 | Task 3: Utilise a Service from the SIMPLI-CITY Service Marketplace for Creating an App..... | 89 |
| 9.3.4 | Task 4: Put an App on the SIMPLI-CITY App Marketplace..... | 91 |
| 9.3.5 | Task 5: Put a Service on the SIMPLI-CITY Service Marketplace..... | 93 |
| 10 | Functional Evaluation of SIMPLI-CITY from Developers' Point of View..... | 95 |
| 10.1 | Detailed Documentation of the Findings..... | 95 |
| 10.2 | Summary of the Results of the Functional Evaluation from Developers' Point of View | 100 |
| 11 | Usability Inspection of the Road Users' Part of the App Marketplace | 101 |
| 11.1 | Approach taken for the Usability Inspection | 101 |
| 11.1.1 | Aim of the Usability Inspection | 101 |
| 11.1.2 | Team of Evaluators..... | 101 |
| 11.1.3 | Context Information..... | 101 |
| 11.1.4 | Method Applied | 102 |
| 11.1.5 | List of Heuristics..... | 103 |
| 11.2 | Summary of the Results of the Usability Inspection..... | 105 |
| 11.3 | Detailed Documentation of the Findings..... | 105 |
| 11.4 | Screenshots Documenting Potential Usability Issues..... | 109 |
| 12 | Usability Evaluation of the Road Users' Part of the App Marketplace..... | 110 |
| 12.1 | Approach taken for the Usability Evaluation | 110 |
| 12.1.1 | Aim of the Usability Evaluation..... | 110 |
| 12.1.2 | Method Applied | 110 |
| 12.2 | Summary of the Results of the Usability Evaluation | 113 |
| 12.3 | Detailed Documentation of the Findings..... | 114 |
| 12.3.1 | Usability Metrics..... | 114 |
| 12.3.2 | Main Positive Aspects Mentioned by the Test-Users | 115 |
| 12.3.3 | Main Usability Issues Revealed During the Evaluation | 116 |
| 12.4 | Task Sheets for Usability Evaluation of the App Marketplace from road-users' point of view..... | 121 |

| | | |
|--------|--|-----|
| 12.5 | Transcription of the Debriefing Interviews | 123 |
| 12.6 | Observer Protocol..... | 126 |
| 13 | Usability Inspection of the Use Case I related SIMPLI-CITY App | 133 |
| 13.1 | Approach Taken for the Usability Inspection | 133 |
| 13.2 | List of Usability Issues Found | 133 |
| 13.2.1 | Screenshots Corresponding to the Revealed Usability Issues..... | 135 |
| 14 | Usability Evaluation of the Use Case I related SIMPLI-CITY App..... | 137 |
| 14.1 | Approach taken for the Usability Evaluation | 137 |
| 14.1.1 | Aim of the Usability Evaluation..... | 137 |
| 14.1.2 | Method Applied | 137 |
| 14.1.3 | Facilitator | 137 |
| 14.1.4 | Test-Users | 137 |
| 14.1.5 | Usability Evaluation Sessions | 138 |
| 14.2 | Main Issues Revealed During the Usability Evaluation..... | 140 |
| 14.2.1 | Issues Noted by the Facilitator | 140 |
| 14.2.2 | Usability Issues Revealed by the Test-Users..... | 140 |
| 14.3 | Detailed Documentation of the Findings..... | 141 |
| 14.3.1 | Usability Metrics..... | 141 |
| 14.4 | Task Sheets for Usability Evaluation of the WP7 related App | 144 |
| 15 | Summary and Conclusions | 146 |

1 Introduction

SIMPLI-CITY – The Road User Information System of the Future – was a project funded by the Seventh Framework Programme of the European Commission under Grant Agreement No. 318201. Its results provide the technological foundation for bringing the “App Revolution” to road users by facilitating data integration, service development, and end user interaction.

This deliverable describes the approach taken regarding evaluation in WP7. It outlines the overall evaluation strategy, gives an overview of the evaluation activities undertaken, and describes the results of all these evaluation activities in detail.

1.1 SIMPLI-CITY Project Overview

Analogously to the “App Revolution”, SIMPLI-CITY adds a “software layer” to the hardware-driven “product” mobility. SIMPLI-CITY takes advantage of the great success of mobile apps that are currently being provided for systems such as Android, iOS, or Windows Phone. These apps have created new opportunities and even business models by making it possible for developers to produce new apps on top of the mobile device infrastructure. Many of the most advanced and innovative apps have been developed by players formerly not involved in the mobile software market. Hence, SIMPLI-CITY supports third party developers to efficiently realise and sell their mobility-related service and app ideas by a range of methods and tools, including the Mobility Services and App Marketplaces.

In order to foster the wide usage of those services, a holistic framework is needed which structures and bundles potential services that could deliver data from various sources to road user information systems. SIMPLI-CITY provides such a framework by facilitating the following main project results:

- **Mobility Services Framework:** A next-generation European Wide Service Platform (EWSP) allowing the creation of mobility-related services as well as the creation of corresponding apps. This enables third party providers to produce a wide range of interoperable, value-added services, and apps for drivers and other road users.
- **Mobility-related Data as a Service:** The integration of various, heterogeneous data sources like sensors, cooperative systems, telematics, open data repositories, people-centric sensing, and media data streams, which can be modelled, accessed, and integrated in a unified way.
- **Personal Mobility Assistant:** An end user assistant that allows road users to make use of the information provided by apps and to interact with them in a non-distracting way – based on a speech recognition approach. New apps can be integrated into the Personal Mobility Assistant in order to extend its functionalities for individual needs.

To achieve its goals, SIMPLI-CITY conducted original research and applied technologies from the fields of Ubiquitous Computing, Big Data, Media Streaming, the Semantic Web, the Internet of Things, the Internet of Services, and Human-Computer Interaction. For more information, please refer to the project website at <http://www.simpli-city.eu>.

| | | | | |
|---|--------------------------|--|----------------------|------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 8 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

1.2 Deliverable Purpose, Scope and Context

The purpose of this deliverable is to outline the evaluation approach taken within WP7 of the project SIMPLI-CITY, to describe the evaluation related activities conducted by the consortium, and to provide a detailed overview of the results of these evaluation activities.

1.3 Document Status and Target Audience

This document is listed in the Description of Work (DoW) as “public”. The results of the evaluation activities conducted within WP7 of the SIMPLI-CITY project might be interesting for all parties, who want to use and exploit (parts of) the SIMPLI-CITY system, since these evaluation results give valuable hints for further improvement of the system.

1.4 Abbreviations and Glossary

A definition of common terms and roles related to the realization of SIMPLI-CITY as well as a list of abbreviations is available in the supplementary document “Supplement: Abbreviations and Glossary”, which is provided in addition to this deliverable.

Further information can be found at <http://www.simpli-city.eu>.

1.5 Document Structure

This deliverable is broken down into the following sections:

Section 1 provides an introduction for this deliverable including a general overview of the project, and outlines the purpose, scope, context, status, and target audience of this deliverable.

Section 2 outlines SIMPLI-CITY’s evaluation approach.

Sections 3, 4 and 5 provide summaries of the results of the Technical Evaluation, the Evaluation from developers’ point of view, and the evaluation from road-users’ point of view respectively.

Sections 6, 7, 8, 9, 10, 11, 12, 13 and 14 give a detailed description of the evaluation activities conducted and the results obtained during Technical Evaluation, Usability Evaluation of SIMPLI-CITY’s Developer Tools, Usability Evaluation of the developers’ part of the SIMPLI-CITY App Marketplace, Usability Inspection from developers’ point of view, Functional Evaluation from developers’ point of view, Usability Inspection of the road users’ part of the SIMPLI-CITY App Marketplace, Usability Evaluation of the road users’ part of the SIMPLI-CITY App Marketplace, Usability Inspection of the use case 1 related app, and Usability Evaluation of the use case 1 related app, respectively.

Section 15 briefly summarises the evaluation results.

| | | | | |
|---|--------------------------|--|----------------------|------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 9 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

2 SIMPLI-CITY's Evaluation Approach

The evaluation strategy followed within SIMPLI-CITY aimed to reach two main objectives:

- to evaluate the prototypes developed within SIMPLI-CITY against the requirements specified in deliverable D2.3 "Requirements Analyses Report"
- to validate the applicability of the SIMPLI-CITY prototypes in a real-world setting

In order to reach these objectives, the evaluation strategy combined elements of technical evaluation, elements of usability evaluation, and elements of functional evaluation.

Technical Evaluation looked at the grade of fulfilment of the requirements specified within deliverable D2.3. Since these requirements cover all parts of the SIMPLI-CITY system, Technical Evaluation was done for the whole SIMPLI-CITY system.

Usability Evaluation and Functional Evaluation looked at the usability and the functionality of the system from users' point of view. However, the SIMPLI-CITY system addresses two user groups: software developers and road users. Hence, the parts of SIMPLI-CITY that are relevant for software developers and the parts of SIMPLI-CITY that are relevant for road users were evaluated separately.

Table 1 gives an overview of the evaluation approach applied in SIMPLI-CITY.

Table 1: SIMPLI-CITY's Evaluation Approach

| Evaluation Element Applied | Objective |
|--|---|
| Technical Evaluation | Evaluate the SIMPLI-CITY system against the requirements specified within deliverable D2.3 |
| Usability Evaluation from Developers' Point of View | Evaluate the Usability of SIMPLI-CITY's "developer package" (comprising of Service Marketplace, the Application Design Studio, the Service Development API, and the part of the App Marketplace targeted to developers) |
| Functional Evaluation from Developers' Point of View | Evaluate the functionality of SIMPLI-CITY's tools targeted at developers in real-world usage |
| Usability Evaluation from Road Users' Point of View | Evaluate the Usability of SIMPLI-CITY's PMA with the Apps (developed for the Use Cases in WP7 and WP8), and the part of the App Marketplace targeted to road users |
| Functional Evaluation from Road Users' Point of View | Evaluate the functionality of SIMPLI-CITY's PMA with the Apps (developed for the Use Cases in WP7 and WP8) in real-world usage conditions |

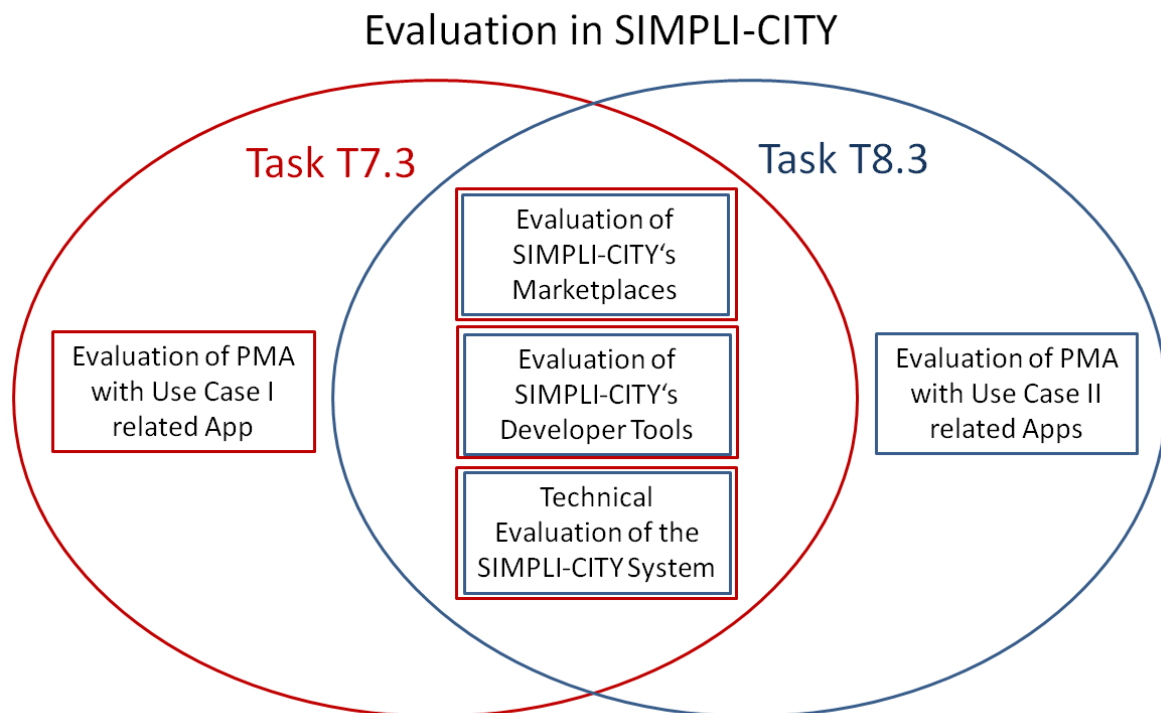


Figure 1: Evaluation within the SIMPLI-CITY Project

As depicted in Figure 1, within the SIMPLI-CITY project, evaluation was assigned to the two tasks T7.3 and T8.3 responsible for the evaluation of the Use Case I and Use Case II related parts of the SIMPLI-CITY system respectively. While the evaluation of the Use Case I and Use Case II related SIMPLI-CITY Apps could easily be assigned to the relevant work packages, the evaluation activities related to the other parts of the SIMPLI-CITY system were relevant for both Use Cases. Therefore, the evaluation of SIMPLI-CITY's market places and SIMPLI-CITY's developer tools as well as the Technical Evaluation of the whole SIMPLI-CITY system were done in close cooperation of both evaluation tasks T7.3 and T8.3.

The following 3 sections of this document include the results of all these common evaluation activities done within the SIMPLI-CITY project, as well as the results of the evaluation of the Use Case I related SIMPLI-CITY App. More detailed information regarding these evaluation activities and their results can be found in Sections 6 to 14 of this document.

For more information regarding the evaluation of the Use Case II related SIMPLI-CITY Apps, refer to deliverable D8.3.

3 Results of the Technical Evaluation

At the beginning of the project, during the specification phase of the SIMPLI-CITY system, in total 212 requirements were defined in deliverable D2.3 Requirements Analysis Report. Of these, 115 were defined as “Must Have” requirements, 38 as “Should Have”, 37 “Could Have”, and 22 “Will Not Have For Now”.

In order to be able to assess to which grade the prototypes developed within the SIMPLI-CITY project fulfil these requirements, a Technical Evaluation Questionnaire was prepared at the end of the project. In this questionnaire the developers of the SIMPLI-CITY prototypes were asked to state for all these requirements, whether they were fulfilled completely, partially, or not at all. Furthermore, the developers were asked to explain for all “Must Have” or “Should Have” requirements, which were not or only partially fulfilled, the reason why these requirements were not implemented completely. The detailed results of the Technical Evaluation Questionnaire can be found in the Section 6 of this document.

The SIMPLI-CITY developers stated that 143 of the 212 requirements were completely fulfilled by the prototypes developed in the project.

Of the 115 “Must Have” requirements, 99 were completely fulfilled. Further 16 of the 115 “Must Have” requirements were fulfilled partially. For these requirements Table 2 provides explanations regarding the state of fulfilment, as stated by the SIMPLI-CITY developers.

Of the 38 “Should Have” requirements defined in deliverable D2.3, only one requirement, namely U188 “Composition of services”, was not fulfilled. According to the developers, this requirement was not covered within the final prototype, since the composition of services happened in apps, not on the level of services. Further 10 of the “Should Have” requirements were fulfilled partially. For these requirements Table 3 provides explanations regarding the state of fulfilment, as stated by the SIMPLI-CITY developers.

Table 2: Partially fulfilled Requirements with Priority “Must Have”

| | Requirement | Explanation Regarding Partial Fulfilment |
|------------|---|---|
| | User goals - Tactical | |
| U15 | Help to balance the traffic in a particular area | SIMPLI-CITY allows balancing the traffic in a particular area by sophisticated traffic management. For this, the PMA acts as both a data source (for the traffic manager) as well as the end user device telling the driver how to behave. However, SIMPLI-CITY was not a traffic management project. Hence, the focus was only partially on this requirement. However, an arbitrary traffic management algorithm could be integrated into the SIMPLI-CITY Mobility Services Framework and used in the PMA. |
| U16 | Provide ordered driving in the city | Same explanation as for requirement U15 |

| | Requirement | Explanation Regarding Partial Fulfilment |
|------|---|---|
| | User Experience – Behaviour – System reaction to context | |
| U27 | Proactive behaviour | To a particular degree the system recognizes which information is relevant to the user, and services are enabled to forward information to the end user App. However, the Apps are responsible of showing such notifications needed for end users |
| | Marketplace | |
| U69 | App download into the device | This is in fact executed by the PMA, not the marketplace. The marketplace sends a push message to notify a request for a new installation |
| U70 | App installation | This is in fact executed by the PMA, not the marketplace |
| U71 | App uninstallation | This is in fact executed by the PMA, not the marketplace |
| | Security | |
| U93 | Third party access to the system | Usage of https has been foreseen at the design phase and the platform is ready to use it. Currently the plain http protocol is used for simplicity reason. |
| | Access to components | |
| U105 | Access to the dialog system | Full Java libraries are provided for PMA side of the App. For MMDI side, some examples are provided and the IDE is capable of highlight properly the source code but not against any library, due to restrictions when creating projects on IntelliJ, that only allow to select one SDK |
| | Business model - Sales | |
| U138 | Provision of apps statistics | Only number of installations available, information from crashes is not available |
| | Developer studio - SDK | |
| U166 | Identification of the developer / signature | The Service Registry permits to store and provide information about the developer of a service. However, it is not possible to integrate with App Marketplace due to client restriction on API |
| | Developer studio – SDK - documentation | |
| U169 | Provision of UI templates | Developers can create the dialogs, but the GUI elements are provided dynamically through the dialog between the User and the MMDI |
| | Developer studio - Statistics | |
| U182 | Provision of statistics, e.g. usage, traffic | This is a manual process by requesting data from the database |

| | Requirement | Explanation Regarding Partial Fulfilment |
|------|---|---|
| | Development | |
| U195 | Proactive user notifications | Personal Mobility Assistant (PMA) can show notifications, when asked by Apps |
| U196 | Prioritization of notifications | This functionality has not been covered within T5.2 as prioritization of notifications is part of the PMA. PMA can show notifications, when asked by Apps |
| | WORLD Scenario (Use Case Topic II.2) | |
| U210 | Reproduction of multimedia information | Only images of the route and audio feedback are provided as other streams are not supported |
| U212 | Notification to end user about the proximity of Points of Interest | No updates are sent to the PMA as this is not supported by PMA. User needs to request updates actively |

Table 3: Partially fulfilled Requirements with Priority “Should Have”

| | Requirement | Explanation Regarding Partial Fulfilment |
|-----|---|---|
| | User Experience – UI and usability of Apps | |
| U18 | Reasonable response time | In most cases, the response from the system is quick. Also, the microphone icon in the PMA shows that the system is listening to the user's voice. However, some visual feedback for showing that the system is processing information (e.g. via the cloud) could help in cases where such processing is slow. |
| | User Experience – Look&Feel of Apps | |
| U23 | Unified Look&Feel within the project | During the project it turned out that design changes should be in the hand of the corresponding app developer in order to be successful in the market. The reason is twofold: Firstly, app developers need to integrate their apps into their companywide look & feel and secondly, apps need to have the possibility to react to new trends without being bound to a UI restriction of the underlying framework. For example, Google has released the Material Design guideline which quickly led to several changes for most apps. As such, SIMPLI-CITY recommends the use of the Material Design but does not enforce a unified look & feel. |
| U24 | Unified Look&Feel for third party developers | see requirement U23 above. |
| | User Experience – Behaviour – System reaction to context | |
| U30 | Reaction to who is in the car, via sensors | The vehicle sensors would be able to recognize the users. However, the current use cases are independent from the driver identity |

| | Requirement | Explanation Regarding Partial Fulfilment |
|------|---|--|
| | User Experience – Behaviour – Interaction with the system | |
| U45 | Automotive quality voice recognition (automotive acoustic models for in car use) | The built-in speech recognizer in Android is used. The recognition quality in the automotive environment depends on the device, connectivity and on the speech recognition provider (either built-in, provided by Google, or from a 3rd party). |
| U46 | High speech recognition rate | The built-in speech recognizer in Android is used. The recognition quality in the automotive environment depends on the device, connectivity and on the speech recognition provider (either built-in, provided by Google, or from a 3rd party). No evaluation of word error rate (WER) has been performed. |
| | User Experience - Feedback | |
| U55 | Rating of apps | Available only from mobile marketplace app, not from the marketplace web interface |
| | Backwards compatibility | |
| U100 | Backwards compatibility of apps | Some checks on the version and classes used by the App are done during the installation process to prevent crashes, but a full list of older libraries with versions is not maintained on PMA side |
| U102 | Backwards compatibility of API | The API version can be changed when editing the App Manifest, but App Design Studio will provide the last library available |
| | System scalability | |
| U124 | Scalability of the service platform | This functionality as such was not covered within the final prototype, however the overall Service Runtime Environment has been designed in a way that allows provisioning of scalability features in the future. |

4 Results of the Evaluation from Developers' Point of View

Evaluation of the SIMPLI-CITY system from developers' point of view included 3 activities:

1. Usability Evaluation by test-users
2. Usability Inspection by experts
3. Functional Evaluation by SIMPLI-CITY developers

The following sections summarise the results of these evaluation activities. More detailed information can be found in the respective Annexes of this document.

4.1 Results of the Usability Evaluation by Test-Users

As a first step in the evaluation of that part of the SIMPLI-CITY system, which is targeted at software developers, usability evaluation sessions with test-users were conducted in Vienna and in Barcelona. The results of these evaluation sessions were communicated to SIMPLI-CITY's developers, so that they could immediately start with improving the developed software.

4.1.1 Results of the Usability Evaluation of SIMPLI-CITY's Developer Tools

The Usability Evaluation Sessions took place at the Technical University of Vienna, Austria. The pre-test was done on Tuesday, 23.06.2015, and 6 usability sessions were conducted on Wednesday, 24.06.2015.

The usability evaluation demonstrated that the developed tools already provide a reasonable support for SIMPLI-CITY software developers. Nevertheless, a number of important usability issues were discovered during the evaluation sequence. In general the evaluated developer plugin was accepted quite well by every user. However, due to different levels of experiences and background knowledge from the users, such as a non-existing familiarity with Windows 8, IntelliJ (the used developer IDE), and the pre-installed unpack tool, high variations of the users' task completion times were noticed.

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY system, which they've especially liked:

- The good integration with IntelliJ IDEA.
- The easy way of publishing services in the Service Marketplace.

However, the test-users mentioned also some aspects of the SIMPLI-CITY system, which should be improved:

- The developer guidance materials, shipped along with developed software, have to be improved.
- The provided user interface should be made more intuitive.
- The software development sequence needs to be simplified.

4.1.2 Results of the Usability Evaluation of the Developers' Part of the SIMPLI-CITY App Marketplace

The Usability Evaluation Sessions for the part of the SIMPLI-CITY App Marketplace, which is dedicated to app developers, took place at Worldline's office in Barcelona, Spain. The 4 usability sessions were conducted on Friday, 17.07.2015 and on Monday, 20.07.2015.

The usability evaluation demonstrated that a developer can publish an application successfully with a minimum of effort and knowledge. Nevertheless, a number of important usability issues were discovered during the evaluation sequence. In general the evaluated Application Marketplace was accepted quite well by every user. Also we can see that the platform is more useful and easy to use on each interaction.

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY system, which they've especially liked:

- The single screen form to create an application.
- The simplicity to approve an application.

However, the test-users mentioned also some aspects of the SIMPLI-CITY system, which should be improved:

- "+ Create App" button functionality and position.
- Confusion between "Add binary" button and "Upload binary" button.
- The place of the "Submit" button on the publication of an application.
- The drag and drop in "Tests Management" tab is not intuitive.
- Differences between test and group test are not stated/clear.
- "Publish" button was hard to find.

4.2 Results of the Usability Inspection by Experts

As a second step in the usability evaluation of SIMPLI-CITY from developers' point of view, the Usability Inspection of the SIMPLI-CITY system from software developers' point of view took place at the TIE headquarters in the Netherlands on 26th of August 2015.

As the usability inspection involved evaluation of development tools (Application Design Studio and Service Development API plus SIMPLI-CITY marketplaces), it was taken care that all evaluators invited to the session already had an ample experience with different development tools, IDEs (integrated development environments) and so on, which they use in their daily work. Thus it was ensured that they could subjectively and partly objectively judge whether they could use the tools being tested in their real life work. They could compare the tools being discussed with alternatives they have used in the past, and they were able to compare the implementation of standard features, ways of interface organisation, responsiveness, developer friendliness, and other aspects.

The unanimous opinion of the evaluators was positive. They liked what the SIMPLI-CITY development tools allow them to do for development of SIMPLI-CITY services and apps, and how these services and apps can be later marketed on the SIMPLI-CITY marketplaces.

Specifically, the evaluators noted positively, that the development tools are based on one of the best freely available IDEs: IntelliJ IDEA, which together with Eclipse is one of the most widely used IDEs in the industry.

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 17 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Furthermore, the evaluators noted that the extensions provided for SIMPLI-CITY specific work are logical and conscious, the layout is straight forward, and this makes it easy for developers to use the tools.

The evaluation went smoothly as all features being evaluated worked as supposed. Based on their experience, the evaluators suggested several improvements, which are listed in detail in the subsequent sections of this document. However these suggestions were mostly cosmetic touches and user interface improvements. According to the evaluators, the overall logic of tooling is absolutely adequate and makes it easier for an experienced service or application developer to achieve her/his goals.

4.3 Results of the Functional Evaluation from Developers' Point of View

In order to get a good picture of the functionality of the SIMPLI-CITY system from developers' point of view, in September 2015 the software developers among the SIMPLI-CITY partners, who had used (parts of) the SIMPLI-CITY system in their work, were asked to express their subjective opinions about the practical usefulness and applicability of the SIMPLI-CITY components, and make suggestions for further improvement.

In general, partners were positive about the SIMPLI-CITY components they had to use in the course of development. This comes to no surprise as the Functional and Technical specifications produced in the beginning of the project in cooperation with all partners were thought out in the smallest details, laying out specifics of interfaces and underlying technologies that partners wanted to be used. The majority of comments, which were received from partners during this functional evaluation of SIMPLI-CITY's developer tools, is related to possible future enhancements, some of which were considered as optional "nice to have" features in the earlier stages of the project and were skipped during the implementation phase, because the development of other more important "must have" features occupied time and resources.

To summarise, it can be said that apart from the fact that there's strong demand for good documentation and guidance, overall SIMPLI-CITY partners' software developers were quite satisfied with the way they had to work to develop within the SIMPLI-CITY framework. The organisation of components and the overall architecture had proven to be up to the expected industry standards.

5 Results of the Evaluation from Road Users' Point of View

Within SIMPLI-CITY evaluation from road users' point of view included

- Evaluation of the SIMPLI-CITY PMA with the Use Case 7 related App
- Evaluation of the SIMPLI-CITY PMA with the Use Case 8 related Apps
- Evaluation of the part of the SIMPLI-CITY App Marketplace that is targeted at road users

The following sections provide a summary of the results of the evaluation of the road-users' part of the SIMPLI-CITY App Marketplace and of the SIMPLI-CITY PMA with the Use Case 7 related App. More information about these evaluation activities can be found in the respective Annexes of this document.

Note: Information regarding the Evaluation of the SIMPLI-CITY PMA with the Use Case 8 related Apps is not included in this document, but can be found in the dedicated deliverable D8.3 Evaluation Report.

5.1 Results of the Evaluation of the Road Users' Part of the App Marketplace

This section gives an overview of the results of the evaluation of the road users' part of the App Marketplace. For detailed information about the usability inspection by experts, please refer to chapter 11 of this document. For detailed information about the usability evaluation by test-users, please refer to chapter 12 of this document.

5.1.1 Results of the Usability Inspection by Experts

The Usability Inspection of the SIMPLI-CITY App Marketplace from road-users' point of view was done by 4 evaluators applying the usability inspection method "Heuristic Evaluation" in Bologna and Graz in July/August 2015.

The version of the SIMPLI-CITY App Marketplace, which was delivered to the evaluators for Usability Inspection, still needed some improvements before it could be considered as usable and functioning as expected. The potential usability problems, which were revealed by the Usability Inspection, were forwarded to the respective developer partners, so that the SIMPLI-CITY App Marketplace can be further improved.

Several potential usability problems were revealed by the evaluators. The main issues were:

- The App Marketplace seemed to be not designed as a "responsive website" --> the page-view did not adjust automatically to the size of the display, and thus the page was only partially displayed on the small screen of the device. (e.g., the "search button" was not visible alongside with the respective communication field where the user was supposed to enter the search term, and the "install button" was not visible when reading the App description and vice versa).
- It was not possible to install the App by clicking on the "install button", and no feedback was provided to the user in case the App could not be installed.
- The user was frequently thrown out from the marketplace and had to re-login.
- There was no information about the marketplace (not even an introductory text), no description of the functionality, no help supporting the user.

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 19 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

However, the Usability Inspection revealed also some positive findings. The main positive aspects noted by the evaluators were:

- The graphic approach and the navigation were good and easy.
- The search function was case insensitive and worked also with fragments of words.

5.1.2 Results of the Usability Evaluation by Test-Users

The Usability Evaluation Sessions for the SIMPLI-CITY App Marketplace from road-users' point of view took place at FGM's premises in Graz, Austria. The pre-test was done on Friday, 4.9.2015, and 5 usability sessions were conducted on Wednesday, 9.9.2015.

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY system, which they've especially liked:

- The website is very clear and simple.
- The users liked the design and the logo of the App marketplace.

However, the test-users mentioned also some aspects of the SIMPLI-CITY App Marketplace, which should be improved:

- There should be some introductory text at the top of the page describing what this website is about (some sort of "Welcome text", which explains to the user the features of the marketplace)
- The search function should be better (not only full text search of the app descriptions, but also finding suitable apps when typing in more general keywords such as e.g. navigation or routing.
- There should be used graphics and not only text in the list of the Apps.
- There should be more information about costs and size of the Apps.
- The website should be optimized for smartphones.

5.2 Results of the Evaluation of the Use Case 7 related App

This section gives an overview of the results of the evaluation of the SIMPLI-CITY App covering the Use Cases "Routing to a big event" and "Personalised Traffic Restrictions". Detailed information about the Usability Inspection and the Usability Evaluation of the PMA with the Use Case 7 related App can be found in Sections 13 and 14 of this document respectively.

5.2.1 Results of the Usability Inspection by Experts

Starting from July 2015, a Usability Inspection was conducted by SRM each time when a new release of the WP7 related SIMPLI-CITY app "Increased Mobility" was available. The results of each of these Usability Inspections were communicated to the developers immediately after each inspection, and the app could be improved accordingly.

Usability Inspection in September 2015 revealed usability issues mainly related to account creation / user registration, specifying origin and destination of a trip, and displaying the actual position.

More information about the Usability Inspection of the "Increased Mobility" app, as well as a detailed list of the usability issues found is provided in Section 13 of this document.

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 20 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

5.2.2 Results of the Usability Evaluation by Test-Users

The Usability Evaluation of the “Increased Mobility” app took place in SRM’s office in Bologna, Italy. Several Usability Issues were found during these sessions with test-users, and the most frequently observed reason for confusion of the test-users was lack of confirmation of the user’s input by the system. All test-users stated that they would like to get more feedback from the system.

In general it can be said that the usability evaluation confirmed the prototype stage of the “Increased Mobility” app, and revealed a lot of useful hints for further improvement of the system.

More detailed information about the Usability Evaluation of the WP7 related SIMPLI-CITY app can be found in Section 14 of this document.

6 Detailed Results of the Technical Evaluation

The Technical Evaluation, which was conducted by the SIMPLI-CITY partners at the end of the project, looked at the fulfilment of the requirements, which were specified in deliverable D2.3 in the first project year. The developers of the SIMPLI-CITY system were asked to provide information, whether these requirements were fulfilled by the prototypes developed within the project, and furthermore the developers were asked to explain for any not completely fulfilled “Must Have” and “Should Have” requirements, why it was not or only partially fulfilled.

The SIMPLI-CITY developers stated that 143 of the 212 requirements were completely fulfilled by the prototypes developed in the project.

Of the 115 “Must Have” requirements, 99 were completely fulfilled, and the remaining 16 were fulfilled partially.

Of the 38 “Should Have” requirements defined in deliverable D2.3, only one requirement, namely U188 “Composition of services”, was not fulfilled. According to the developers, this requirement was not covered within the final prototype, since the composition of services happened in Apps, not on the level of services. Further 10 of the “Should Have” requirements were fulfilled only partially.

The following tables, which include all requirements as specified in the deliverable D2.3, provide the detailed results of the Technical Evaluation of the SIMPLI-CITY prototypes.

Table 4: End-User Requirements

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|----|--|-----------|------|----|---|-------------|
| | | yes | part | no | | |
| | Usage types | | | | | |
| U1 | Support of car drivers as end users | x | | | | Must Have |
| U2 | Support of cyclists as end users | x | | | | Should Have |
| U3 | Support of private transport passengers as end users | x | | | | Should Have |
| U4 | Support of public transport users as end users | x | | | | Could Have |
| U5 | Support of pedestrians as end users | x | | | | Could Have |
| U6 | Support of truck drivers as end users | x | | | | Could Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------------------------|---|-----------|------|----|--|------------------------------|
| | | yes | part | no | | |
| U7 | Support for disabled drivers. It should take into account the interaction between the user and the road information system | | x | | The Multimodal User Interface allows the support for disabled drivers. However, the according functionalities need to be supported by app developers. | <i>Will Not Have For Now</i> |
| U8 | Multiuser support. It should support multiple drivers and different users with their specific basic data and user preferences | | | x | | <i>Will Not Have For Now</i> |
| User goals - Strategic | | | | | | |
| U9 | Allow users to become green. | x | | | | <i>Must Have</i> |
| U10 | Reduction of the number of accidents | x | | | Through specialized apps (i.e., the blackspot warning service) | <i>Must Have</i> |
| U11 | Provision of useful information to drivers | x | | | | <i>Must Have</i> |
| U12 | No distraction of the driver | x | | | | <i>Must Have</i> |
| U13 | No SIMPLI-CITY "Big Brother" | x | | | | <i>Should Have</i> |
| User goals - Tactical | | | | | | |
| U14 | Show real time carbon print | x | | | | <i>Must Have</i> |
| U15 | Help to balance the traffic in a particular area | | x | | SIMPLI-CITY allows to balance the traffic in a particular area by sophisticated traffic management. For this, the PMA acts as both a data source (for the traffic manager) as well as the end user device telling the driver how to behave. However, SIMPLI-CITY was not a traffic management project. Hence, the focus was only partially on this requirement. However, an arbitrary traffic management algorithm could be integrated into the SIMPLI-CITY Mobility Services Framework and used in the PMA. | <i>Must Have</i> |
| U16 | Provide ordered driving in the city. | | x | | Same explanation as above | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|---|---|-----------|------|----|--|------------------------------|
| | | yes | part | no | | |
| U17 | Allow local authorities to connect easily with users. | x | | | | <i>Must Have</i> |
| User Experience - UI and usability of Apps | | | | | | |
| U18 | Reasonable response time | | x | | In most cases, the response from the system is quick. Also, the microphone icon in the PMA shows that the system is listening to the user's voice. However, some visual feedback for showing that the system is processing information (e.g. via the cloud) could help in cases where such processing is slow. | <i>Should Have</i> |
| U19 | Minimum manual configuration | | | | | <i>Could Have</i> |
| U22 | Personalization - Incremental configuration | | x | | No configuration by the end user is needed | <i>Could Have</i> |
| U20 | Multilingual | | x | | The development environment is capable of handling strings and grammar components for different languages | <i>Will Not Have For Now</i> |
| U21 | Link voice commands to apps | | | x | | <i>Will Not Have For Now</i> |
| User Experience - Look&Feel of Apps | | | | | | |
| U23 | Unified Look&Feel within the project | | x | | During the project it turned out that design changes should be in the hand of the corresponding app developer in order to be successful in the market. The reason is twofold: Firstly, app developers need to integrate their apps into their company-wide look & feel and secondly, apps need to have the possibility to react to new trends without being bound to a UI restriction of the underlying framework. For example, Google has released the Material Design guideline which quickly led to several changes for most apps. As such, SIMPLI-CITY recommends the use of the Material Design but does not enforce a unified look & feel. | <i>Should Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|---|--|-----------|------|----|--|------------------------------|
| | | yes | part | no | | |
| U24 | Unified Look&Feel for third party developers | | x | | During the project it turned out that design changes should be in the hand of the corresponding app developer in order to be successful in the market. The reason is twofold: Firstly, app developers need to integrate their apps into their company wide look & feel and secondly, apps need to have the possibility to react to new trends without being bound to a UI restriction of the underlying framework. For example, Google has released the Material Design guideline which quickly led to several changes for most apps. As such, SIMPLI-CITY recommends the use of the Material Design but does not enforce a unified look & feel. | <i>Should Have</i> |
| U25 | Usage of UI guidelines within the project | x | | | | <i>Should Have</i> |
| U26 | Intuitive Usability | x | | | | <i>Should Have</i> |
| User Experience - Behaviour - System reaction to context | | | | | | |
| U27 | Proactive behaviour | | x | | To a particular degree the system recognizes which information is relevant to the user, and services are enabled to forward information to the end user app. However, the Apps are responsible of showing such notifications needed for end users | <i>Must Have</i> |
| U28 | The system learns from feedback | | | x | | <i>Will Not Have For Now</i> |
| U29 | Reaction to who is in the car, via simple dialog | | | x | The current use cases are independent from the driver identity | <i>Could Have</i> |
| U30 | Reaction to who is in the car, via sensors | | x | | The vehicle sensors would be able to recognize the users. However, the current use cases are independent from the driver identity | <i>Should Have</i> |
| U31 | Reaction to time of the day | x | | | | <i>Must Have</i> |
| U32 | Reaction to sensors | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|--|---|-----------|------|----|---|-------------------|
| | | yes | part | no | | |
| U33 | Reaction to KPIs from the car, like level of fuel and kilometres done | x | | | | <i>Must Have</i> |
| U34 | Reaction to history of usage | x | | | | <i>Must Have</i> |
| U35 | Reaction to traffic information, like traffic jams, train schedules, road works, accidents, and strikes | x | | | | <i>Must Have</i> |
| User Experience - Behaviour - Interaction with the system | | | | | | |
| U36 | Natural speech recognition | x | | | | <i>Must Have</i> |
| U37 | Results oriented instead of service/app recognition oriented | x | | | | <i>Could Have</i> |
| U38 | Disambiguation | x | | | | <i>Could Have</i> |
| U39 | Provision of a limited number of alternatives | x | | | | <i>Could Have</i> |
| U40 | Friendly voice | x | | | | <i>Could Have</i> |
| U44 | Voice quality | x | | | | <i>Could Have</i> |
| U41 | Input interactions with system via multimodal UI: on screen, voice control, and non-voice control | x | | | | <i>Must Have</i> |
| U42 | Output interaction from system through UI | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|--|--|-----------|------|----|--|--------------------|
| | | yes | part | no | | |
| U43 | Non-distracting interaction | x | | | | <i>Should Have</i> |
| U45 | Automotive quality voice recognition (automotive acoustic models for in car use) | | x | | The built-in speech recognizer in Android is used. The recognition quality in the automotive environment depends on the device, connectivity and on the speech recognition provider (either built-in, provided by Google, or from a 3rd party). | <i>Should Have</i> |
| U46 | High speech recognition rate | | x | | The built-in speech recognizer in Android is used. The recognition quality in the automotive environment depends on the device, connectivity and on the speech recognition provider (either built-in, provided by Google, or from a 3rd party). No evaluation of word error rate (WER) has been performed. | <i>Should Have</i> |
| U47 | Voice interaction through the car audio system (microphone & loudspeakers) for hands free in car use | | | x | | <i>Could Have</i> |
| User Experience - Behaviour - Interaction with the system | | | | | | |
| U48 | App crashes are minimized | x | | | | <i>Should Have</i> |
| U49 | A crash should not impact other apps | x | | | | <i>Should Have</i> |
| User Experience - Offline access | | | | | | |
| U50 | Prefetching of media data & offline access | x | | | | <i>Must Have</i> |
| U51 | Avoid the download of data from 3G | x | | | | <i>Should Have</i> |
| U52 | Offline access of data used within apps | x | | | | <i>Should Have</i> |
| U53 | App recommendations | | x | | Automatically pre-fetch data from users / cars but no functionalities to detect apps that could make use of such data. The number of services / apps was limited in the first iterations of the project, and recommendation could not be properly tested | <i>Could Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|---|---|-----------|------|----|--|------------------------------|
| | | yes | part | no | | |
| U54 | Expiration of data | x | | | | <i>Should Have</i> |
| User Experience - Feedback | | | | | | |
| U55 | Rating of apps | | | x | Available only from mobile marketplace app, not from the marketplace web interface | <i>Should Have</i> |
| U56 | Feedback to developers through the marketplace | x | | | | <i>Should Have</i> |
| U57 | Social network functionality with the objective of spreading the word | | | x | | <i>Could Have</i> |
| U58 | User recommendations | | | x | | <i>Could Have</i> |
| User Experience - Generic Services | | | | | | |
| U59 | User centric data services, e.g., Facebook events, calendar | x | | | | <i>Must Have</i> |
| Marketplace | | | | | | |
| U60 | Payment for apps and services – Mock-up | | | x | | <i>Could Have</i> |
| U61 | Payment for apps and services - Fully deployed | | | x | | <i>Will Not Have For Now</i> |
| U62 | Versioning of apps | x | | | | <i>Must Have</i> |
| U63 | Upgrading of apps | x | | | | <i>Must Have</i> |
| U64 | Quality assurance – Mock-up | x | | | | <i>Must Have</i> |
| U65 | Quality assurance - Fully deployed | x | | | | <i>Could Have</i> |
| U66 | Marketplace easy to use | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|-----------------------------|--|-----------|------|----|---|------------------------------|
| | | yes | part | no | | |
| U67 | Apps and services are easy to buy | x | | | | <i>Must Have</i> |
| U68 | Discovery of apps and services | x | | | | <i>Must Have</i> |
| U69 | App download into the device | | x | | This is in fact executed by the PMA, not the marketplace. The marketplace sends a push message to notify a request for a new installation | <i>Must Have</i> |
| U70 | App installation | | x | | This is in fact executed by the PMA, not the marketplace | <i>Must Have</i> |
| U71 | App uninstallation | | x | | This is in fact executed by the PMA, not the marketplace | <i>Must Have</i> |
| Environment - Device | | | | | | |
| U72 | Android support | x | | | | <i>Must Have</i> |
| U73 | iOS support | | | x | | <i>Will Not Have For Now</i> |
| U74 | Blackberry support | | | x | | <i>Will Not Have For Now</i> |
| U75 | Windows phone support | | | x | | <i>Will Not Have For Now</i> |
| U76 | Tablet support | x | | | | <i>Could Have</i> |
| U77 | The PMA fits into the pocket | x | | | | <i>Will Not Have For Now</i> |
| U78 | The battery lifetime of the PMA is longer than 1 day | | x | | Usually resources are cached as far as possible. However, in general the battery lifetime depends on the used device. | <i>Will Not Have For Now</i> |
| Environment | | | | | | |
| U79 | Support to exchange devices | | | x | | <i>Will Not Have For Now</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|-----------------------------------|---|-----------|------|----|--|------------------------------|
| | | yes | part | no | | |
| U80 | Profile in the cloud | x | | | | <i>Must Have</i> |
| U81 | Storage of data in the cloud | x | | | | <i>Must Have</i> |
| Environment - Connectivity | | | | | | |
| U82 | Gateway to data sources | | | x | Has not been implemented since this requirement was of low importance. | <i>Will Not Have For Now</i> |
| U83 | Interaction with head up display | | | x | Out of scope for the project as no use case was supporting such requirements | <i>Will Not Have For Now</i> |
| U84 | Interaction with user's devices, e.g. contacts exchange | | | x | Out of scope for the project as no use case was supporting such requirements | <i>Could Have</i> |
| U85 | Interaction with car sensors, e.g. car sensors | x | | | | <i>Must Have</i> |
| Privacy | | | | | | |
| U86 | Transparency | x | | | | <i>Must Have</i> |
| U87 | Confidentiality. Does not give away data to third parties | x | | | | <i>Must Have</i> |
| U88 | Data encryption | x | | | | <i>Must Have</i> |
| U89 | Certification. Only certified apps are allowed to access users data | x | | | | <i>Must Have</i> |
| Security | | | | | | |
| U90 | Availability | x | | | | <i>Must Have</i> |
| U91 | Integrity | x | | | | <i>Must Have</i> |
| U92 | Secure access to system | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|-----|----------------------------------|-----------|------|----|--|------------------|
| | | yes | part | no | | |
| U93 | Third party access to the system | | x | | Usage of https has been foreseen at the design phase and the platform is ready to use it. Currently the plain http protocol is used for simplicity reason. | <i>Must Have</i> |

Table 5: Developer Requirements

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|-------------------------------------|-----------|------|----|---|-------------|
| | | yes | part | no | | |
| | Backwards compatibility | | | | | |
| U100 | Backwards compatibility of apps | | x | | Some checks on the version and classes used by the App are done during the installation process to prevent crashes, but a full list of older libraries with versions is not maintained on PMA side | Should Have |
| U101 | Backwards compatibility of services | x | | | | Could Have |
| U102 | Backwards compatibility of API | | x | | The API version can be changed when editing the App Manifest, but App Design Studio will provide the last library available | Should Have |
| | Robustness | | | | | |
| U103 | Fault tolerance | x | | | | Must Have |
| U104 | Stability | x | | | | Must Have |
| | Access to components | | | | | |
| U105 | Access to the dialog system | | x | | Full Java libraries are provided for PMA side of the App. For MMDI side, some examples are provided and the IDE is capable of highlight properly the source code but not against any library, due to restrictions when creating projects on IntelliJ, that only allow to select one SDK | Must Have |
| U106 | Access to cloud services | x | | | | Must Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|---|-----------|------|----|---|-----------------------|
| | | yes | part | no | | |
| | Access to sensors | | | | | |
| U107 | Access to sensors of the vehicle (via car APIs) when the functionality is available. Types: fuel consumption, speed, acceleration | x | | | | Should Have |
| U108 | Access to smart device sensors | x | | | | Must Have |
| U109 | Access to remote sensors, e.g. traffic sensors | x | | | | Must Have |
| | Access to car components | | | | | |
| U110 | Remote control of car components when the functionality is available, e.g. air conditioning, heating, battery charge timing | | | x | Out of scope for the project as no use case was supporting such requirements | Will Not Have For Now |
| U111 | Provision of web site of the car | | | x | | Could Have |
| | Data sources | | | | | |
| U112 | Support of open data | x | | | | Must Have |
| U113 | Handling of multimedia data | x | | | | Must Have |
| | Data sources - Data services | | | | | |
| U114 | Configuration of the frequency of update of the data from data sources | x | | | This requirement was superseded by the requirement to have push functionality, which makes it unnecessary to update or poll background services regularly | Must Have |
| U115 | Transformation support | x | | | | Must Have |
| U117 | Data filtering | x | | | | Must Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------------------------------------|---|-----------|------|----|---|--------------------|
| | | yes | part | no | | |
| U118 | Data correlation | x | | | | <i>Must Have</i> |
| U116 | Unified data model | x | | | | <i>Must Have</i> |
| U119 | Data summarisation | x | | | | <i>Must Have</i> |
| U120 | Handle of data streams | x | | | | <i>Must Have</i> |
| Data sources - Storage type | | | | | | |
| U121 | Local storage of data | x | | | | <i>Should Have</i> |
| U122 | Storage of data in the cloud: binary, semantic and structured | x | | | | <i>Must Have</i> |
| System scalability | | | | | | |
| U123 | Scalability of the cloud infrastructure | x | | | | <i>Should Have</i> |
| U124 | Scalability of the service platform | | x | | This functionality as such was not covered within the final prototype, however the overall Service Runtime Environment has been designed in a way that allows provisioning of scalability features in the future. The REST Proxy being the single entry point for all method calls can be retrofitted to serve as a load balancer and the built-in monitoring and SLA watching modules can spawn new Service Runtime Environment instances should load increase and response times degrade. | <i>Should Have</i> |
| General standard approach | | | | | | |
| U125 | Open interfaces | x | | | | <i>Should Have</i> |
| U126 | Openness of the system | x | | | | <i>Should Have</i> |
| U127 | Extensibility | x | | | | <i>Should Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|--|-----------|------|----|--|-------------|
| | | yes | part | no | | |
| | Business model - Finance - Apps | | | | | |
| U128 | The system allows app developers to make money | x | | | | Should Have |
| U129 | The system allows the consortium to make money from apps | x | | | | Should Have |
| U130 | Apps are free but supported, i.e. red hat model | | | | | Could Have |
| U131 | Free and premium versions of apps | | | | | Could Have |
| U132 | Guarantee and minimum service: life time, support time | | | | | Could Have |
| | Business model - Finance - Services | | | | | |
| U133 | The system allows service developers to make money | x | | | | Could Have |
| U134 | The system allows the consortium to make money from services | x | | | | Could Have |
| U135 | Services are free but supported, i.e. red hat model | | | x | | Could Have |
| U136 | Free and premium versions of services | | | x | | Could Have |
| U137 | Guarantee and minimum service: life time, support time | | x | | Storage is possible but logic needs to be performed by an external payment provider during commercialization phase | Could Have |
| | Business model - Sales | | | | | |
| U138 | Provision of apps statistics | | x | | Only number of installations available, information from crashes is not available | Must Have |
| U139 | Provision of services statistics | x | | | | Must Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|--------------------------------------|------------------------------------|-----------|------|----|---|------------------------------|
| | | yes | part | no | | |
| U140 | Promotional aspects | | | x | | <i>Could Have</i> |
| U141 | Permit to spread the word | | x | | | <i>Could Have</i> |
| App Marketplace | | | | | | |
| U142 | Quality/checking certification | x | | | | <i>Could Have</i> |
| U143 | App publication | x | | | | <i>Must Have</i> |
| U145 | App removal | x | | | | <i>Must Have</i> |
| U147 | Easy to publish | x | | | | <i>Must Have</i> |
| U144 | App versioning | x | | | | <i>Must Have</i> |
| U146 | Call for developers | | | x | | <i>Will Not Have For Now</i> |
| Service Marketplace | | | | | | |
| U148 | Service registration | x | | | | <i>Must Have</i> |
| U149 | Service extension and modification | x | | | | <i>Must Have</i> |
| U150 | Service management | x | | | | <i>Must Have</i> |
| U151 | Service versioning | x | | | | <i>Must Have</i> |
| Service Level Agreement (SLA) | | | | | | |
| U152 | SLA support | x | | | | <i>Must Have</i> |
| U153 | Usage of an official SLA standard | x | | | | <i>Must Have</i> |
| U154 | Simple SLA description standard | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|--|-----------|------|----|--|-----------------------|
| | | yes | part | no | | |
| | Developer studio - SDK - API | | | | | |
| U155 | Provision of JAVA API | x | | | | Must Have |
| U156 | Provision of C/C++ API | | | x | | Will Not Have For Now |
| U157 | Standard programming interface | x | | | | Must Have |
| U158 | Easy access to API through the developer studio | x | | | | Must Have |
| | Developer studio - SDK - Use case driven development | | | | | |
| U159 | Support for dialogue | x | | | | Must Have |
| U160 | Support for apps | x | | | | Must Have |
| U161 | Support for data services | x | | | | Must Have |
| U162 | Support for backend services | x | | | | Must Have |
| | Developer studio - SDK | | | | | |
| U163 | Easy to use environment | x | | | | Should Have |
| U164 | Low investment required | x | | | | Should Have |
| U165 | User friendly developer studio interface | x | | | | Could Have |
| U166 | Identification of the developer / signature | | x | | The Service Registry permits to store and provide information about the developer of a service. However, it is not possible to integrate with App Marketplace due to client restriction on API | Must Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|---|---|-----------|------|----|---|------------------------------|
| | | yes | part | no | | |
| U167 | Hot updates | x | | | | <i>Will Not Have For Now</i> |
| Developer studio - SDK - documentation | | | | | | |
| U168 | Provision of source code examples | x | | | | <i>Must Have</i> |
| U169 | Provision of UI templates | | x | | Developers can create the dialogs, but the GUI elements are provided dynamically through the dialog between the User and the MMDI | <i>Must Have</i> |
| U170 | Provision of best practices | x | | | | <i>Must Have</i> |
| U171 | Provision of tutorials | x | | | | <i>Must Have</i> |
| U172 | Provision of guidelines | x | | | | <i>Must Have</i> |
| U173 | Provision of examples | x | | | | <i>Must Have</i> |
| Developer studio - Mind-set | | | | | | |
| U174 | Permit to develop an app in less than a day | x | | | | <i>Should Have</i> |
| U175 | Hide complexity from developer | x | | | | <i>Should Have</i> |
| U176 | Developer should not need to be skilled on spoken dialogue design | x | | | | <i>Will Not Have For Now</i> |
| Developer studio - Simulator | | | | | | |
| U177 | Provision of a PMA emulator | | x | | Developers can make use of an Android Virtual Device to emulate the PMA | <i>Will Not Have For Now</i> |
| Developer studio - Functionality description | | | | | | |
| U178 | Definition of minimum hardware requirements of apps | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|--|---|-----------|------|----|---|------------------------------|
| | | yes | part | no | | |
| U179 | Description of services: semantic description, keywords, phrasing, ontology | x | | | | <i>Must Have</i> |
| Developer studio - Debugging | | | | | | |
| U180 | Easy debugging of services | x | | | | <i>Could Have</i> |
| U181 | App and service crash reports | | x | | | <i>Could Have</i> |
| Developer studio - Statistics | | | | | | |
| U182 | Provision of statistics, e.g. usage, traffic | | x | | This is a manual process by requesting data from the database | <i>Must Have</i> |
| U183 | Provision of bug reports | | x | | | <i>Could Have</i> |
| U184 | Provision of crash reports | | x | | Logs can be provided but not step by step debugging, as Apps are installed. Bug and statistic information is not delivered to the cloud. | <i>Could Have</i> |
| Development - Interaction between apps and services | | | | | | |
| U185 | Standardised messages between apps | x | | | | <i>Must Have</i> |
| U186 | Registry of installed SIMPLI-CITY apps | x | | | | <i>Must Have</i> |
| U187 | Composition of apps | | | x | | <i>Will Not Have For Now</i> |
| U188 | Composition of services | | | x | This requirement was not covered within the final prototype, since the composition of services happens in apps, not on the level of services. | <i>Should Have</i> |
| Development - Interface | | | | | | |
| U189 | Unified interface for accessing sensors | x | | | | <i>Must Have</i> |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|-------------------------------|---|-----------|------|----|---|------------------------------|
| | | yes | part | no | | |
| U190 | Unified interface for accessing user centric data | x | | | | <i>Must Have</i> |
| U191 | Simulation of sensors | x | | | | <i>Must Have</i> |
| Development | | | | | | |
| U192 | Service deployment | x | | | | <i>Must Have</i> |
| U193 | Exchange of information from apps to server | x | | | | <i>Must Have</i> |
| U194 | Exchange of information from server to apps | x | | | | <i>Must Have</i> |
| U195 | Proactive user notifications | | x | | Personal Mobility Assistant (PMA) can show notifications, when asked by Apps | <i>Must Have</i> |
| U196 | Prioritization of notifications | | x | | This functionality has not been covered within T5.2 as prioritization of notifications should be done on a user's Personal Mobility Assistant (PMA). PMA can show notifications, when asked by Apps | <i>Must Have</i> |
| U197 | Get support from community | | | x | | <i>Will Not Have For Now</i> |
| Cross-cutting concerns | | | | | | |
| U198 | Small number of device to support | x | | | | <i>Must Have</i> |
| U199 | Compliance to regulations | x | | | | <i>Should Have</i> |

Table 6: Use Cases Requirements

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|---|-----------|------|----|---|-----------|
| | | yes | part | no | | |
| | Generic | | | | | |
| U201 | Provision of personalized info, e.g. travel, costs | x | | | The info is personalized based on user context on some factors: position, sensors, etc. | Must Have |
| | IBM Scenario (Use Case Topic I.1) | | | | | |
| U202 | Diagnosis of abnormal traffic condition in real-time | x | | | | Must Have |
| U203 | Prediction of abnormal traffic condition | x | | | | Must Have |
| U204 | Support for querying diagnosis historic | x | | | | Must Have |
| U205 | Support for querying impact factor on traffic condition | x | | | | Must Have |
| | SRM Scenario (Use Case Topic I.2) | | | | | |
| U206 | Optimization of financial resources, e.g. avoiding tolls, cheaper parking | x | | | | Must Have |
| U207 | Support suggestions for road/trip optimization if conditions change | x | | | | Must Have |
| U208 | Possibility to continue a previously started trip planning on the same device or different device | x | | | | Must Have |

| | Requirement | fulfilled | | | Explanation why not or only partially implemented | Priority |
|------|---|-----------|------|----|---|-------------|
| | | yes | part | no | | |
| | WORLD Scenario (Use Case Topic II.2) | | | | | |
| U209 | Provision of real-time information about the current route | x | | | The information about traffic is taken from STAR system and the sensors provided by PMA, and it's real-time as far as the information provided by this system is real-time. However, no updates are sent to the PMA and no real time information is notified to the app without a manual request, as this is not supported by PMA | Must Have |
| U210 | Reproduction of multimedia information | | x | | Only images of the route and audio feedback are provided as other streams are not supported | Must Have |
| U211 | Reproduction of streaming audio | x | | | | Should Have |
| U212 | Notification to end user about the proximity of Points of Interest | | x | | No updates are sent to the PMA as this is not supported by PMA. User needs to request update actively | Must Have |
| U213 | Social network integration | x | | | | Must Have |
| | CRF Scenario (Use Case Topic II.1) | | | | | |
| U214 | Reporting to the end user about eco-driving information | x | | | | Must Have |
| U215 | Vehicle information available to the system | x | | | | Must Have |
| U216 | Provision of real time feedback to the user in order to improve his/her performance | x | | | | Must Have |
| U217 | Access to a journey-related eco-driving data using the specific web portal | x | | | Information related to history trip eco-driving data included are available with the Eco Live assistant app choosing the option Trip History not through a web portal | Must Have |
| U218 | Comparing (eco-)performances of different drivers | x | | | | Must Have |
| U219 | Reward drivers through the eco-driving contest | x | | | | Should Have |

7 Usability Evaluation of SIMPLI-CITY's Developer Tools

This chapter presents the results of the Usability Evaluation of the SIMPLI-CITY system from software developers' point of view.

In the following Section 7.1 the Usability Evaluation approach is described briefly. Section 7.2 gives a summary of the results of the Usability Evaluation, and Section 7.3 outlines the findings in detail. Furthermore, Section 7.4 includes the Task Sheets, which were used during the Usability Evaluation sessions, and Section 7.5 provides a transcription of the Debriefing Interviews conducted at the end of each evaluation session.

7.1 Approach taken for the Usability Evaluation

7.1.1 Aim of the Usability Evaluation

This Usability Evaluation was done in order to find out how easy it is for developers to use the SIMPLI-CITY tools targeted at software developers. The results of this Usability Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

7.1.2 Method Applied

For the Usability Evaluation of the SIMPLI-CITY "developer package", the well-established Usability Evaluation method "*Thinking Aloud Test*" was applied:

A test-user is asked to speak out loud all her/his thoughts while using the system to complete a given test-task. By observing the test-user's way of working on the task, and by analysing the reasons for confusion, hesitation, or mistakes of the test-user, valuable hints for improvement of the usability of the system can be obtained.

7.1.3 Team of Facilitators

Three persons formed the team, which organised and conducted the usability evaluation sessions of SIMPLI-CITY's "developer package".

Table 7: Team of Facilitators of the Usability Evaluation Sessions

| Name (Organisation) | Role within the Facilitators' Team |
|---------------------------------|--|
| Michaela Kargl (FGM) | "Moderator" (guided the test-users through the session) |
| Philipp Hönisch (TUV) | "Observer" (took notes of revealed usability issues) |
| User Rostyslav Zabolotnyi (TUV) | "Technician" (ensured proper functioning of the technical equipment, software, and recordings) |

7.1.4 Technical Setup for the Usability Evaluation

The test-user worked on a Laptop running Windows 8.1 as operating system. On this computer the IntelliJ IDEA 14.1.3 Community Edition was installed as IDE, and it was part of the task of the test-user to install the respective SIMPLI-CITY IntelliJ Plugins. Furthermore, also Mozilla Firefox, Foxit Reader, WinRAR, and 7-Zip were installed on this

computer, which the test-user utilised for related tasks such as e.g., checking websites or extracting archives.

TeamViewer 10 was used for displaying the test-computer's screen on the observer's computer, and BlueBerry FlashBack Express 5 (5.8.0.3644) was used for screen recording. A Logitech 920 HD Pro webcam and the related software were utilised for general video and audio recording of the session, especially for recording of the feedback interview.

7.1.5 Usability Evaluation Sessions

The Usability Evaluation Sessions took place at the Technische Universität Wien, Vienna, Austria. The pre-test was done on Tuesday, 23.06.2015, and 6 usability sessions were conducted on Wednesday, 24.6.2015. The average duration of the Usability Evaluation Sessions was 46 minutes, with the longest session lasting for 61 minute and the shortest session taking 36 minutes.

All Usability Evaluation Sessions followed the same process:

First the moderator welcomed the test-user, introduced the team, and explained the purpose of the session. The moderator gave an overview of the test procedure, and asked the test-user for her/his permission that the session can be recorded (video, audio and screen recording), and these records can be used for later analysis and reporting purposes. After the test-user signed the consent form, the moderator encouraged the test-user to ask questions, whenever anything is unclear. In a short introductory interview, the moderator obtained from the test-user some demographic information (gender and age) and some information regarding relevant skills and knowledge of the test-user, such as e.g., familiarity with Windows, Android, Java and IntelliJ IDEA, knowledge of SIMPLI-CITY, and expertise with Thinking Aloud tests.

After this introductory part, the moderator explained to the test-user, how the "Thinking Aloud" method works, and asked the test-user to do one test-task after the other and answer the Single Ease Question (SEQ) immediately after completion of each task. While the test-user was working on the task, she/he spoke out loud all her/his thoughts and considerations, the team of facilitators watched the test-user silently, and the observer took notes of the test-user's activities and any usability issues that became apparent.

After finishing the last test-task, the moderator thanked the test-user for her/his cooperation and asked the following three questions in the debriefing interview: "How was it?" "Was there anything that you found to be especially good?" "Was there anything that you found to be especially bad?" When the test-user finished answering these questions, the moderator asked her/him to fill-in the System Usability Scale (SUS), a standardised questionnaire to assess a user's perception of the overall usability of a system. In case the test-user had further questions these were answered by the team, and after that the session was finished.

After each session, the recordings were saved, and the materials and setup was prepared for the next session.

7.1.6 Test-Users

In total 7 test-users participated in the Usability Evaluation of the SIMPLI-CITY "developer package". (In addition, one test-user (User #0) participated in the dry-run of the evaluation

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 43 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

session.) All of them answered the introductory questionnaire, did the 4 test-tasks, and provided their feedback in the debriefing interview and feedback questionnaire. However, the evaluation session with the first test-user was handled as pre-test, and after this session the material, specifically the task descriptions and the introductory questionnaire, was slightly modified according to the lessons learnt from this pre-test.

All 7 test-users were students of Computer Science at the Technical University of Vienna. All of them were between 25 and 30 years old, and there was only one woman among them. These demographic facts resemble the target user group for SIMPLI-CITY's "developer package" quite well: According to a recent survey¹ the average age of software developers in Europe is between 26 and 30 years, and only about 6% of software developers are female.

Table 14 gives an overview of the relevant skills and knowledge of the test-users as stated by them in the introduction questionnaire.

Table 8: Overview of the Test-Users' relevant Skills and Knowledge

| Test-User | User #2 | User #1 | User #3 | User #4 | User #5 | User #6 | User #7 |
|---|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Usability Eval. Session | Pre Test | Session 1 | Session 2 | Session 3 | Session 4 | Session 5 | Session 6 |
| Expertise in Usage of App Stores | high | medium | high | high | high | medium | medium |
| Expertise in Software Development | medium | high | medium | medium | medium | high | high |
| Expertise in Java Coding | high | high | medium | medium | medium | high | high |
| Expertise in Working with Windows | medium | high | high | low | high | low | low |
| Expertise in Developing for Android | low | low | no | low | no | low | no |
| Familiar with IntelliJ IDEA | no | yes | yes | yes | no | yes | yes |
| Preferred Keyboard | English | German | German | German | English | German | English |
| Knowledge about SIMPLI-CITY | heard of | nothing | nothing | used | heard of | heard of | nothing |
| Participated in Thinking Aloud Test Ever Before | no | no | no | yes | no | no | yes |

Four of the test-users self-assessed their software development expertise as being "medium", three of them stated that they do have "high" expertise in software development. Four of the test-users stated that their expertise in coding with Java is "high", the other three test-users self-assessed their expertise with Java as being "medium". Three of the test-users did not have much experience in working with Windows, and two of the test-users were not familiar with the IntelliJ IDEA. None of the test-users had much expertise in developing Android Apps or Services.

¹ <http://stackoverflow.com/research/developer-survey-2015>

None of the test-users was involved in the development of the SIMPLI-CITY “developer package”. One of the test users was working for other parts of SIMPLI-CITY, three of the test-users had heard of SIMPLI-CITY before, and the remaining three test-users did not know anything about SIMPLI-CITY.

Two of the test-users had participated in a Thinking Aloud Test before.

7.1.7 Evaluation Tasks

The test-users were asked to complete the following tasks:

- Task 0: Install the SIMPLI-CITY plugin for IntelliJ IDEA
- Task 1: Try to develop a “Hello World Service” by using the SIMPLI-CITY system.
- Task 2: Find the created Backend Service in the Service Marketplace.
- Task 3: Delete the created service.

Annex A of this document includes the task-sheets, which the test-users got during the usability evaluation session.

7.2 Summary of the Results of the Usability Evaluation

The usability evaluation demonstrated that the developed tools already provide a reasonable support for SIMPLI-CITY software developers. Nevertheless, a number of important usability issues were discovered during the evaluation sequence. In general the evaluated developer plugin was accepted quite well by every user. However, due to different levels of experiences and background knowledge from the users, such as a non-existing familiarity with Windows 8, IntelliJ (the used developer IDE), and the pre-installed unpack tool, we noticed high variations of the users’ task completion times. For example, in the task 0, in which the users had to install a plugin for the IDE, some users had major problems, as they were simply not familiar with the security settings of Windows 8, i.e., it is not allowed to unpack a zip file directly into the program installation folder. The average task completion time (or “Time on Task”, TOT) for the task 0 was about 8 minutes with a standard deviation of 3.3. Further, while some users were able to complete that task on their own (57%), others needed minor help from the technician.

The first task was to develop a simple hello-world method, integrate it into a backend service and deploy that service in the SRE. We expected a task completion time of maximal 20 minutes. However, 2 out of the 7 users took longer than 20 minutes. Even though most of the users claimed to have high expertise in software development, only 43% of the users were able to finish this task on their own. Again, this can be reduced to the fact that the users were not familiar with the used IDE or SIMPLI-CITY software that they had to use. The average completion time was 16 minutes with a standard deviation of 4.8 which is still below the allowed maximum time.

As expected, compared to the first 2 tasks, the 3rd and 4th task were quite simple and straight forward. We achieved a completion rate of 100%, i.e., every user was totally able to finish these tasks on their own without any help. In both cases, the average completion time was about 2 minutes with a minor standard deviation of 1.

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY system, which they've especially liked:

1. The good integration with IntelliJ IDEA.
2. The easy way of publishing services in the Service Marketplace.

However, the test-users mentioned also some aspects of the SIMPLI-CITY system, which should be improved:

1. The plugin installation sequence is not straight forward.
2. The developer guidance materials, shipped along with developed software, have to be improved.
3. The provided user interface is sometimes counterintuitive and misleading.
4. The software development sequence needs to be simplified.

After the Usability Evaluation, the results of the evaluation sessions were communicated to SIMPLI-CITY's developers. During the remaining time of the project, their work was focused on solving these issues and improving the developed software.

7.3 Detailed Documentation of the Findings

7.3.1 Usability Metrics

7.3.1.1 Task Completion Success Rate

Overall performance of each participant was evaluated depending on his/her ability to finish the task on his/her own. The task success rate score is the number of participants that finished the task successfully divided by the total number of participants.

We believe that Task 2 and Task 3 have the highest completion rate due to appeared user familiarity with the evaluated system. While participants still had difficulties with these tasks, they managed to resolve them on their own. Task 0 was planned as the introductory one that required only basic operating system usage skills (i.e., Windows 8). However, some users did not manage to complete this task either due to lacking Windows usage experience or due to poor documentation. Finally, Task 1 required users to leverage their experience as developers, what caused some of them to yield for help, mainly due to the unknown IDE and development environment configuration.

Table 9: Task Completion Success Rate

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|---------------------|------------|------------|-------------|-------------|
| User #1 | ✓ | - | ✓ | ✓ |
| User #2 | - | - | ✓ | ✓ |
| User #3 | - | - | ✓ | ✓ |
| User #4 | ✓ | ✓ | ✓ | ✓ |
| User #5 | ✓ | - | ✓ | ✓ |
| User #6 | - | ✓ | ✓ | ✓ |
| User #7 | ✓ | ✓ | ✓ | ✓ |
| Success Rate | 57% | 43% | 100% | 100% |

✓ = successfully completed;

- = not completed / help needed for completion

7.3.1.2 Ease-of-Task Ratings

After each task, the participants rated the ease of completing the task, by answering the “Single Ease Question” (SEQ)² “Overall this task was?” on a 7-point rating scale ranging from “Very Difficult”(1) to “Very Easy”(7).

Table 10 shows the ratings given to the single tasks by the test-users, and provides an average of these ratings for each task.

Table 10: Ease-of-Task Ratings

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|---------------------------------------|------------|------------|------------|------------|
| User #1 | 6 | 4 | 7 | 4 |
| User #2 | 5 | 5 | 7 | 7 |
| User #3 | 3 | 4 | 7 | 5 |
| User #4 | 6 | 5 | 6 | 6 |
| User #5 | 7 | 4 | 7 | 6 |
| User #6 | 4 | 5 | 6 | 5 |
| User #7 | 6 | 6 | 7 | 5 |
| Average Rating | 5.1 | 4.5 | 6.5 | 5.6 |
| Percent “Easy” (Rating 6 or 7) | 50% | 13% | 88% | 50% |

All participants perceived Task 2 (“Find the created Backend Service in the Service Marketplace”) to be the most easiest one, and Task 1 (“Try to develop a “Hello World Service by using the SIMPLI-CITY system”) to be the most difficult test-task. Some of the test-users also perceived Task 0 (“Install the SIMPLI-CITY plugin for IntelliJ IDEA”) as being rather difficult.

² Jeff Sauro: “10 Things to know about the Single Ease Question (SEQ)”, online at <https://www.measuringu.com/blog/seq10.php>, visited 12.06.2015

7.3.1.3 Time on Task (TOT)

The amount of time needed by each participant to finish their tasks depended on their background knowledge about used tools and environment, task description length and quality of user interface.

Task 0, which required participants to install SIMPLI-CITY plugin for IntelliJ IDEA required only basic operating system knowledge (i.e., Windows 8) and the ability to follow the manual. On average this task took 8 minutes, while the fastest participant required only 5 minutes and slowest required 15 minutes.

Task 1 asked participants to develop a “Hello World Service” using the SIMPLI-CITY system. This task required basic programming skills and ability to fulfil critical SIMPLI-CITY software development steps. On average this task was completed in 16 minutes, while the fastest participant needed 10 minutes and the slowest required 22 minutes.

Task 3 required the users to find the created Backend Service in the Service Marketplace. The participants had to locate the Service Marketplace, get familiar with it and locate the service, which was deployed there while performing the previous task. On average this task took 2 minutes, while the shortest time was 1 minute and longest time was 4 minutes.

Task 4 asked the participants to delete the created service in the Service Marketplace. The developers had to locate the functionality to delete the service and verify that the service is deleted. On average this task was completed in 2 minutes, with a fastest time of 1 minute and a longest time of 4 minutes.

Table 11: Participants' Time-On-Task

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|---------------------------------|----------|-----------|----------|----------|
| User #1 | 6 | 15 | 2 | 2 |
| User #2 | 15 | 22 | 3 | 2 |
| User #3 | 9 | 18 | 2 | 4 |
| User #4 | 5 | 11 | 4 | 1 |
| User #5 | 7 | 21 | 2 | 2 |
| User #6 | 8 | 12 | 1 | 2 |
| User #7 | 6 | 10 | 2 | 1 |
| Average TOT (in minutes) | 8 | 16 | 2 | 2 |

7.3.1.4 Rating of the Overall System-Usability

After the completion of all tasks, the participants were asked to fill in a standardised feedback questionnaire (SUS System Usability Scale), answering the following 10 questions on a 5-point scale ranging from “strongly disagree” to “strongly agree”:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I would need the support of an experienced person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most developers would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

SUS yields a single number³ representing a composite measure of the overall usability of the system. SUS scores have a range of 0 ("not usable at all") to 100 ("perfectly usable system").

Table 12: Participants' Rating of the Overall System-Usability (SUS Score)

| Test-User | Test-User's Rating of the 10 Questions in the SUS questionnaire | | | | | | | | | | SUS-Score ⁴ |
|-----------|---|-----|-----|-----|-----|-----|-----|-----|-----|------|------------------------|
| | Q 1 | Q 2 | Q 3 | Q 4 | Q 5 | Q 6 | Q 7 | Q 8 | Q 9 | Q 10 | |
| User #0 | 3 | 2 | 2 | 1 | 2 | 3 | 4 | 3 | 2 | 1 | 57.5 |
| User #1 | 3 | 2 | 4 | 1 | 4 | 3 | 4 | 2 | 3 | 4 | 65.0 |
| User #2 | 5 | 1 | 4 | 2 | 5 | 2 | 5 | 1 | 3 | 1 | 87.5 |
| User #3 | 4 | 1 | 2 | 2 | 4 | 5 | 4 | 2 | 2 | 3 | 57.5 |
| User #4 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 75.0 |
| User #5 | 4 | 1 | 5 | 3 | 5 | 1 | 5 | 1 | 2 | 1 | 85.0 |
| User #6 | 4 | 2 | 3 | 2 | 3 | 2 | 4 | 2 | 5 | 2 | 72.5 |
| User #7 | 5 | 2 | 4 | 1 | 3 | 3 | 5 | 2 | 4 | 2 | 77.5 |

Analysis of the answers of the test-users to the feedback questionnaire reveals that the SIMPLI-CITY "developer package" has got a SUS score of 72.2. According to Bangor A. et al⁵, this SUS score can be translated to users' perceiving that the usability of the system is "acceptable / o.k." This is quite good for a prototype system, but indicates that there's still some improvement of the system's usability necessary.

7.3.2 Main Positive Aspects Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY system, which they've especially liked:

1. the good integration with IntelliJ IDEA

User #3: *"I liked the service creation in IntelliJ with a separate plugin and menu structure so that you can access and find the needed methods in an easy way."*

User #7: *"It's easy. The buttons, the menu is very structured and the icon-integration is good, so that the icons are prominent and you can see the new plugin and the functionality of the plugin."*

³ According to User #0 Brooke „SUS – A quick and dirty usability scale” ([http://hell.meiert.org/core/pdf/sus.pdf/?/](http://hell.meiert.org/core/pdf/sus.pdf?/))

⁴ According to User #0 Brooke „SUS – A quick and dirty usability scale” ([http://hell.meiert.org/core/pdf/sus.pdf/?/](http://hell.meiert.org/core/pdf/sus.pdf?/)) :

"To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10 the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of the SUS score. SUS scores have a range of 0 to 100."

⁵ Adjective ratings related to SUS scores: 25 = "worst imaginable", 38 = "poor", 52 = "acceptable / o.k.", 74 = "good", 85 = "Excellent", 100 = "best imaginable" (Source: Bangor A., Kortum P., Miller J.: "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale", in Journal of Usability Studies JUS, Vol. 4, Issue 3, May 2009, pp. 114-123)

2. the easy way of publishing services in the Service Marketplace

User #1: *"It seemed that the publishing worked pretty straight forward – it seems very nice. In my experience it can be rather difficult to publish a service in an open marketplace like that. So, if it stays that easy, it would be a very handy thing to have."*

User #2: *"Well, especially good is to have a server side for publishing the service that you develop, and then it makes the cycle for the development and publication starting from scratch. You just need a couple of installations stuff and then can work with this. That is a good point."*

User #6: *"I found it good that IntelliJ automatically uploads the service into the marketplace, and that there are no extra tasks that have to be done. So, I like it when everything is done in one programme, when there's only a button and it works."*

User #3: *"... And I liked the easy way, how you can publish a service to the marketplace."*

7.3.3 Main Usability Issues Revealed During the Evaluation

7.3.3.1 Main Usability Issues Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users mentioned several aspects of the SIMPLI-CITY system, which should be improved:

1. it is difficult to find the 'delete button' in the SIMPLI-CITY menu of the IntelliJ IDEA

User #5: *"Probably I would improve the menu. 'Bundle service and persist' that's not really obvious that the delete-button is in there."*

User #1: *"The delete-button placement was not intuitive at all. Because it was on the menu option that stated that it persists a service and creates the service."*

User #7: *"The deletion process of the service was ... hm ... I think there could be some menu integration for the popup for this 'persist service'. I think the popup is not necessary, so I did not like that, I think it would be easier to set up a context pane in IntelliJ."*

User #3: *"Mixture of the items is not good: so that you have a distinct menu for publishing and also included the deleting functionality. That should be in a separate menu entry. That would be more clear to say 'manage the market place services' or 'manage the published services' so that you can go to your service and say that you want to delete it or suspend it."*

2. the guidance material needs improvement

User #6: *"The help how to bundle the service, how to upload the service could be improved – First I did not get that I had to push that 'bundle it' button. Maybe that could be a little bit more clearly mentioned or there could be another structure of the help."*

User #3: "... the system might be good to use, once the manual is a bit more elaborated... For a moment I was kind of puzzled, when I saw the dummy-text in one of the chapters..."

User #1: "The documentation was not 100% intuitive, and without Maven background and without IntelliJ background, which I had before, I assume that I would have had more difficulties in locating specific elements and packaging the bundle. But this cannot be avoided I think, because it's just so that if you are familiar with a tool it's good but if you're not you have to get used to it, if you want to work with it. So, you cannot really get around this, but the documentation could be better formulated and more intuitively designed."

3. the view of the Service Marketplace and the search functionality should be improved

User #6: "I think the web view of the Service Marketplace could be a little bit over-full, if there are many apps in there. ... So this could be improved with an extra view."

User #4: "The counter-intuitive search interface puzzled me, because normally you would expect that it works case-insensitive."

4. the feedback from the system should be improved

User #1: "I was not aware first that the IDE is actually communicating and interacting with an external marketplace in the background without me noticing."

User #0: "The worst thing – well, I did not know, whether I'm done or not with this first stuff with the plugin. It was hard like – I just copied some parts, and then I did not know whether I'm done or not... I would have expected to get just some kind of confirmation."

5. functionalities of the system should be improved

User #7: "The deployment process itself could probably be simplified, cause basically the process is 'create a jar file and deploy it' – there's no necessity to have this step in between that you have to compile it with Maven first and then deploy it – you could just have a button 'deploy my thing' and the process does it for you."

User #0: "Actually, normal plugins are usually installed with some kind of installation menu: ...next → next → next... and then I check some checkboxes to not install some crapware, and then usually I get the confirmation 'installation complete', and then a checkbox like 'start it', and then I see that everything works... It would be better to have something like that."

User #2: "You need to support more than just one IDE. And also this junction when you generate a project can be richer, like provide more functionality. I mean, if I want to have a more complex application, what shall I do? You need more detail. ... It should provide more specific things, like, if I want an application for GPS, should I have a generated code interact with the Google API from someone to generate a map. So, that's what I want. So inside of the

plugin you should have some key to generate code for user – for developer that is the user of this system. The system should support more than just a skeleton.”

7.3.3.2 Main Usability Issues Noted by the Observer

In the following, the main usability issues, as noted by the observer during the test are listed (in the order of occurrence):

1. *Task 0: “Install the SIMPLI-CITY plugin for IntelliJ IDEA”*

1.1. The IntelliJ IDEA plugin installation sequence has to be simplified. Our users had significant problems installing it even if they were following the documentation.

2. *Task 1: “Try to develop a „Hello World Service“ by using the SIMPLI-CITY system”*

2.1. The user interface is suboptimal.

2.1.1. Every menu item related to developed plugin is disabled if no SIMPLI-CITY project is opened. The users have difficulties figuring out what to do to enable them.

2.1.2. The Development Guide is unclear and misleading. Users had significant problems understanding how to proceed and how to achieve their goal.

2.1.3. The manifest.xml is being rebuilt on project manipulation (bundle and persist, etc.) This may cause a problem editing and extending it.

2.1.4. Registry URL should be configurable on project creation. Currently, a user has to configure it manually in configuration.

2.1.5. Error reporting confuses developer. Whenever a developer forgets some step or configures something incorrectly, SIMPLI-CITY system should provide a reasonable message explaining the problem instead of some internal exception.

2.1.6. Automatic project packaging is required. The developed plugin has to conceal intermediate steps that do not require the user’s attention.

2.1.7. “Connect to a Service...” menu item is misleading. Users often had difficulties understanding if they need it or not.

2.1.8. “Persist Service bundle” dialog is overloaded and hard to use. It contains a lot of functionality which are hardly used or usable. The users had troubles to navigate through that dialog box and finding the correct functionality. Also it is not possible to open the Development Guide while this dialog is active.

2.1.9. Important development-related operations lack any visual feedback.

Participants were often wondering if the requested operation succeeded or not.

2.2. Problems were detected in SIMPLI-CITY plugin functionality.

2.2.1. New SIMPLI-CITY project setup is incorrect. The users had to perform additional unexpected steps (such as Java SDK setup) in order to have a working solution.

2.2.2. User Authentication Handling is missing. Actual service registry instance needs to have user authentication enabled.

3. *Task 2: “Find the created Backend Service in the Service Marketplace.”*

3.1. The need to login is not obvious. Functionality should not be completely hidden but rather “greyed out” if the user is not authorized or logged in.

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 52 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

3.2. The search menu is case-sensitive, what is unexpected for many users.

4. Task 3: “Delete created service”.

4.1. “Delete Service” is located incorrectly. The participants had difficulties finding that functionality in the SIMPLI-CITY menu item.

4.2. System feedback is missing. When a user deletes a service, it is not clear, whether the service is actually deleted or not. In addition, it was not clear why it was also deleted from the service marketplace.

After the Usability Evaluation, the described observations were provided to the respective developers and they were actively working on it in order to improve the provided functionality and resolve the detected issues.

7.3.3.3 Screenshots Documenting Potential Usability Issues

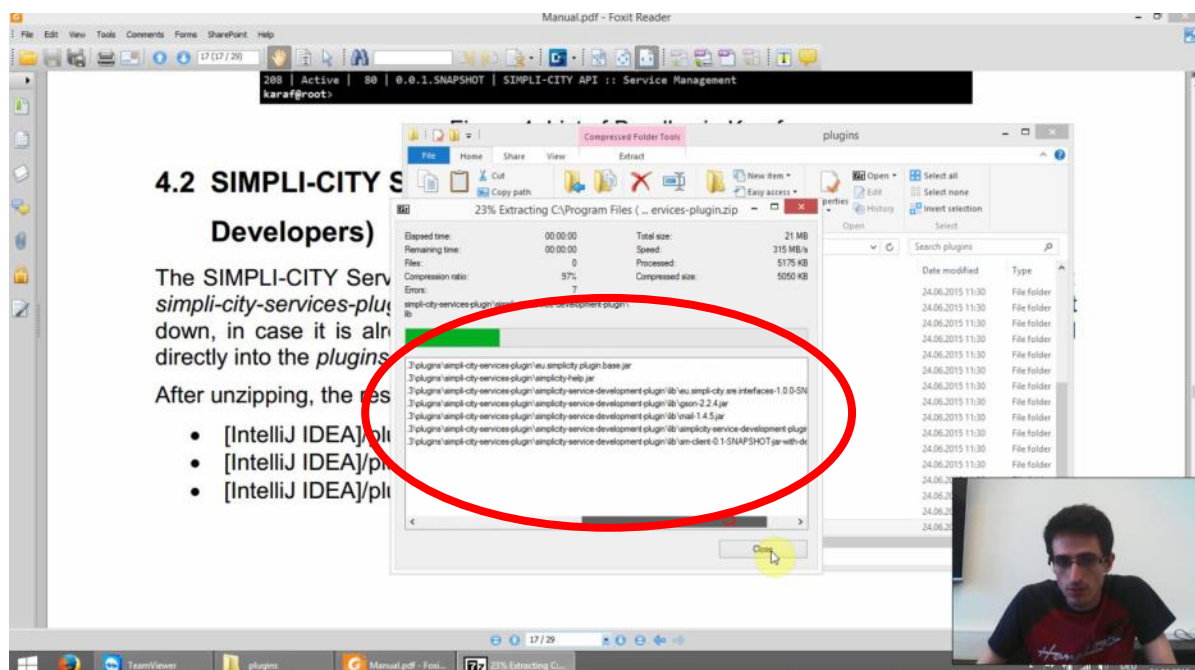


Figure 2: Access Rights Issue during Task 0 (see Issue 1.1)

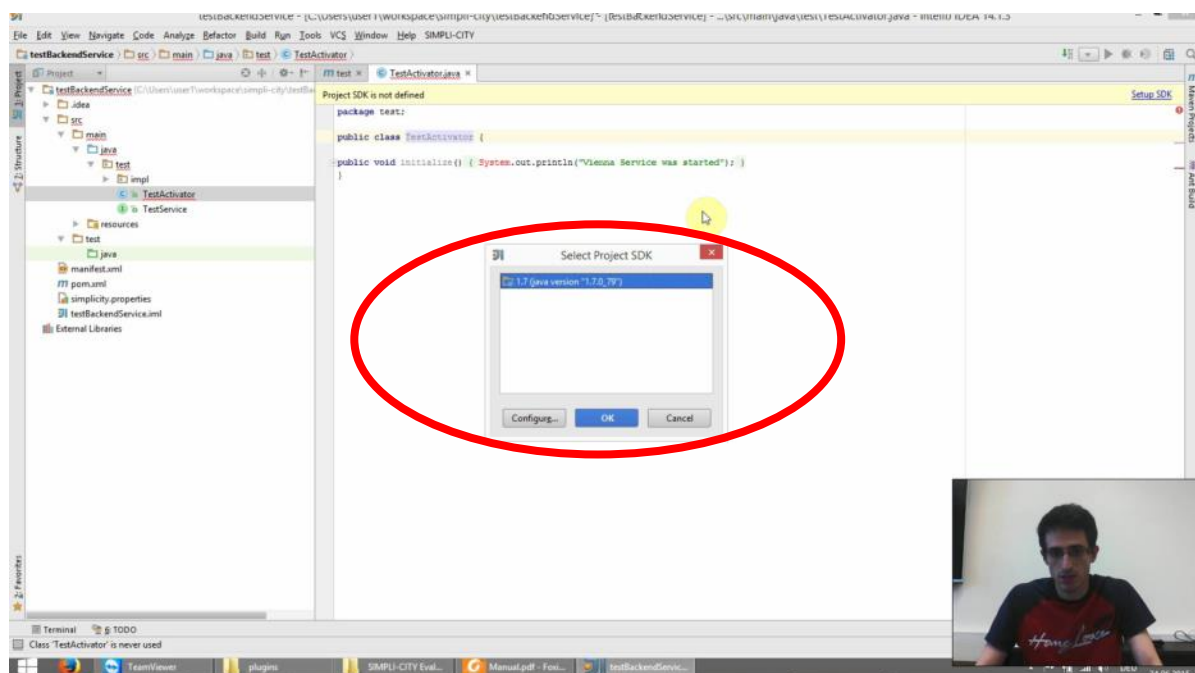


Figure 3: SDK Definition Issue during Task 1 (see Issue 2.2.1)

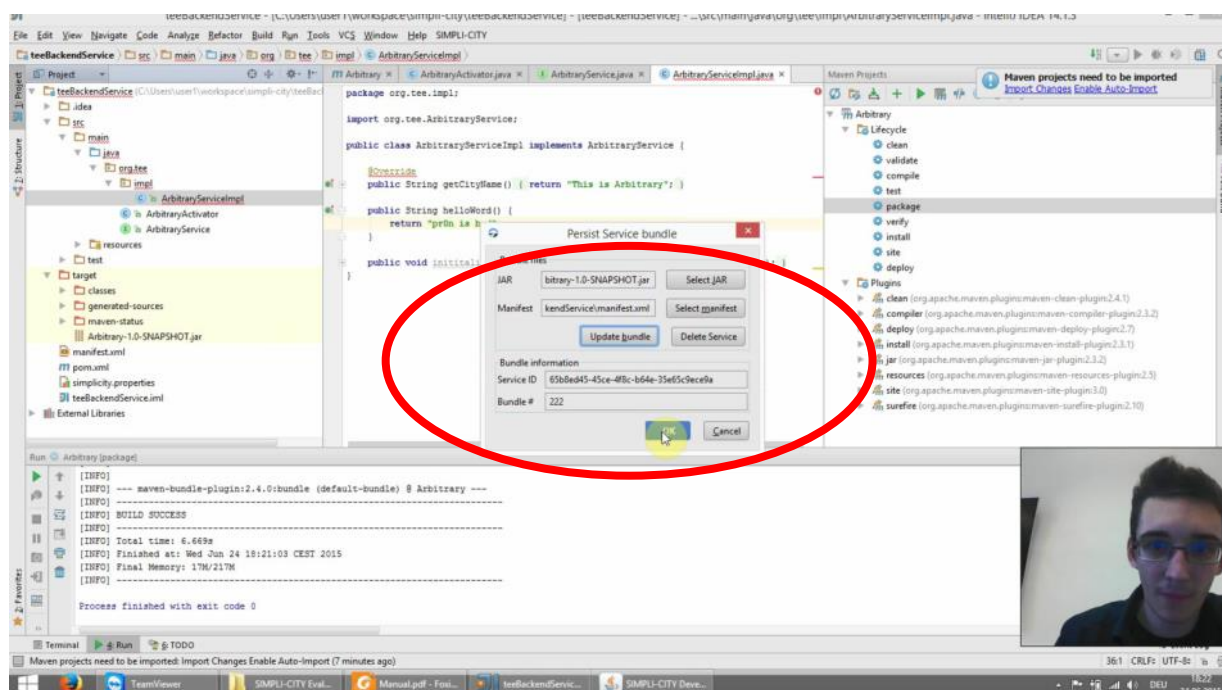


Figure 4: Overloaded Plugin Interface during Task 1 (see Issue 2.1.8)

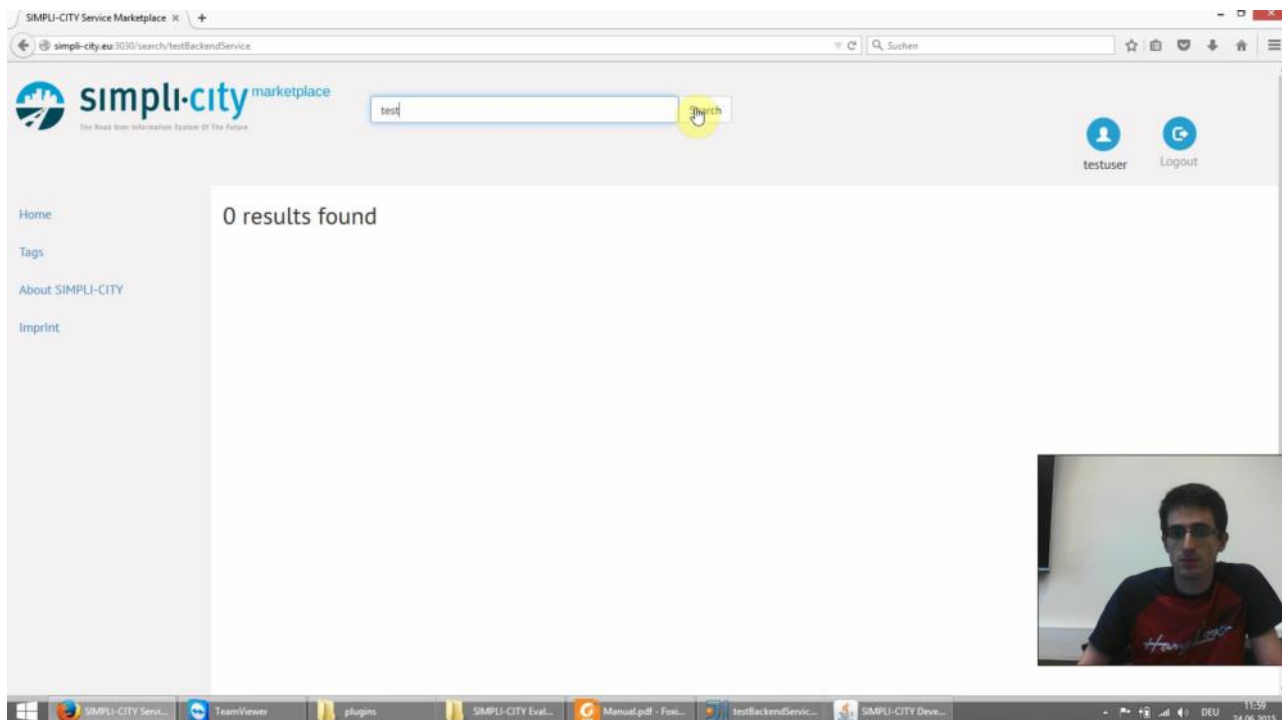


Figure 5 Marketplace Search Function is Case Sensitive (see Issue 3.2)

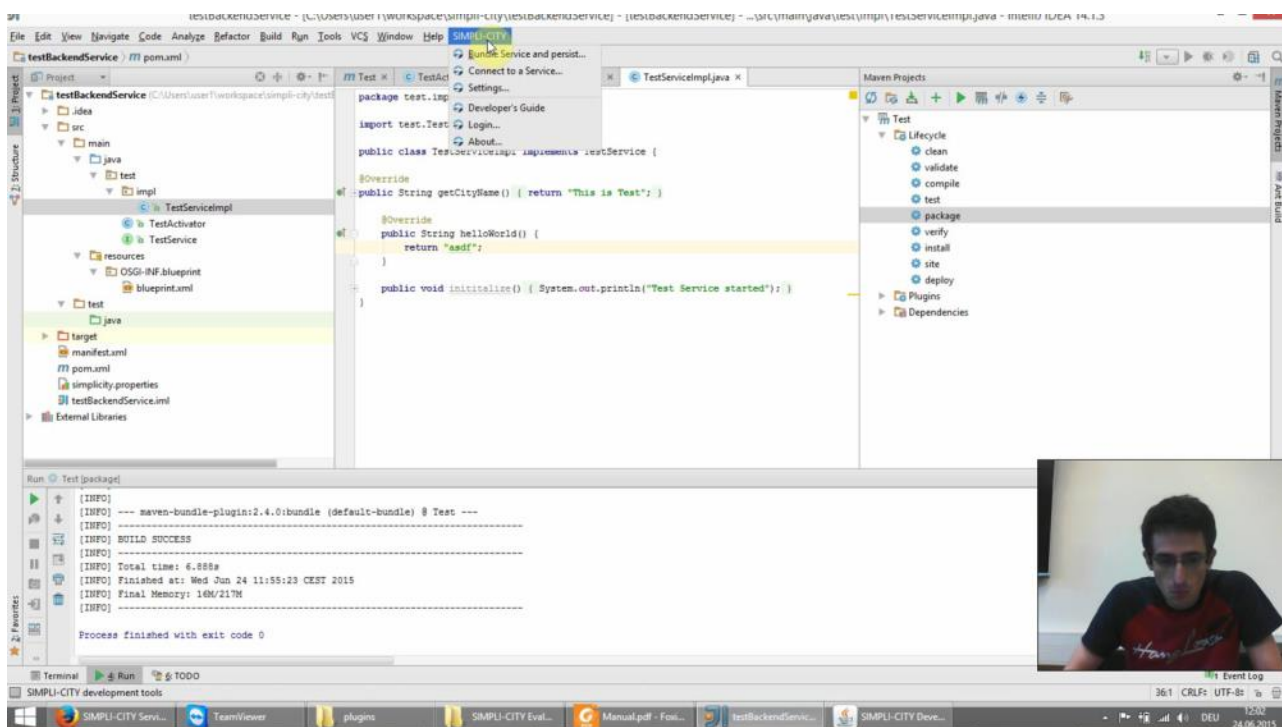


Figure 6 Delete Service Functionality is not Easy to Find (see Issue 4.1)

7.4 Task Sheets for Usability Evaluation of SIMPLI-CITY's Developer Tools

TASK 0: Install the SIMPLI-CITY plugin for IntelliJ IDEA

1. Locate “simpli-city-services-plugin.zip” and “Manual.pdf” on the desktop in “SIMPLI-CITY Evaluation” folder.
2. Open “Manual.pdf” page 17-18, Section 4.2 and follow the instructions.
 - a. **Hint:** The link to IntelliJ Idea folder is provided in “SIMPLI-CITY Evaluation” folder as well.
3. Start IntelliJ Idea and click “Create New Project”, there should be a SIMPLI-CITY project category.

Tell the moderator, when you finish this task.
Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy



TASK 1:

Try to develop a „Hello World Service“ by using the SIMPLI-CITY system

The URL of the Service Registry is:

<http://simpli-city.eu:8888/cxf/serviceRegistry>

1. Start IntelliJ Idea
2. Create a new SIMPLI-CITY Backend Service project.
(groupId/ArtifactId/ProjectName can be an almost arbitrary string, project location should be "...\\workspace\\SIMPLI-CITY\\<yourNameHere>BackendService")
3. Please ensure that Java SDK is correctly setup
 - a. Open an arbitrary java file in order to verify if Java SDK is configured properly.
(hint: there is a yellow message on top of the editor field)
 - b. You can see the standard Maven project structure on the left side (src/main/java/...).
4. Create a "helloWorld" method in the interface
"src/main/java/<GroupId>/<ArtifactId>Service" and in the class located in
"src/main/java/<GroupId>/impl/<ArtifactId>ServiceImpl".
 - a. Ignore any existing errors in the project.
5. Return an arbitrary string from the defined method implementation.
6. Change Service Registry URL in "Settings" from SIMPLI-CITY menu item.
 - a. **Hint:** Correct URL is in the task description.
7. Open the "SIMPLI-CITY Development Guide" from the "SIMPLI-CITY" Menu
 - a. Follow the "Bundling a Service" instructions.
 - b. **Hint:** the maven plugin panel is docked to the right side of IDE ("Maven Projects")
8. Ensure Service received a ServiceID.

Tell the moderator, when you finish this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy



TASK 2:**Find the created Backend Service in the Service Marketplace****The URL of the Service Marketplace is: <http://simpli-city.eu:3030/>****The username for the Service Marketplace is: “testuser”****The password for the Service Marketplace is: “testuser123”**

Tell the moderator, when you finish this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

**TASK 3:****Delete created service.**

1. Use “SIMPLI-CITY” menu item in IntelliJ Idea for that.
2. Verify in Service Marketplace that your service is gone.

Tell the moderator, when you finish this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

7.5 Transcription of the Debriefing Interviews



Figure 7: In the debriefing interview the test-users explained what they liked at the SIMPLI-CITY system and what they perceive to be the main issues that should be improved.

Test-user User #0 (Dry-run)

Moderator: *"How was it?"*

User #0: *"Well, it was fine. The first task was kind of puzzling and it was kind of complicated, but then it was easy. In general it was fine."*

Moderator: *"What was the best thing?"*

User #0: *"I think the easiest was to delete the service. That was the easiest part. But the best – I do not know – the best was that everything worked from the first time. And even though there were some errors, it still worked, I do not know why."*

Moderator: *"And what was the worst thing?"*

User #0: *"The worst thing – well, I did not know, whether I'm done or not with this first stuff with the plugin. It was hard like – I just copied some parts, and then I did not know whether I'm done or not..."*

Moderator: *"What would you have expected?"*

User #0: *"I do not know, just some kind of confirmation. Actually, normal plugins are usually installed with some kind of installation menu: ...next → next → next... and then I check some checkboxes to not install some crapware, and then usually I get the confirmation "installation complete", and then a checkbox like "start it", and then I see that everything works..."*

Moderator: *"So you would have expected something like that?"*

User #0: *"Yes, it would be better."*

Test-user User #2 (Pre-Test)

Moderator: *"How was it?"*

User #2: *"I liked it."*

Moderator: *"So, regarding the SIMPLI-CITY system, what did you find especially good?"*

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 59 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

User #2: *“Well, especially good is to have a server side for publishing the service that you develop, and then it make the cycle for the development and publication starting from scratch. You just need a couple of installations stuff and then can work with this. That is a good point, the way to interact with developer and aligned it”*

Moderator: *“What do you think is the worst thing?”*

User #2: *“The worst thing is, I’m not sure, maybe what’s not good is stick to the IDEA. The first difficulty is using the IntelliJ. I used it before several times and I do not like it. So you need to support more than just one IDE. And also this junction function when you generate a project can be richer, like provide more functionality. I mean, if I want to have a more complex application, what shall I do? You need more detail.”*

Moderator: *“So, what would you expect?”*

User #2: *“Well, it should provide more specific things, like, if I want an application for GPS, should I have a generated code interact with the Google API from someone to generate a map. So, that’s what I want. So inside of the plugin you should have some key to generate code for user – for developer that is the user of this system. Something like this. The system should support more than just a skeleton, because I’m the developer, I know how to write an interface and know how to write the implementation, and of course it’s, however, a strong motivation to publish to the market place. “*

Test-user User #1 (Test-Session 1)

Moderator: *“How was it?”*

User #1: *“At first I was not 100% sure what is the purpose or the aim of the tasks. However, since they were chained together, the picture became more clear afterwards.”*

Moderator: *“What did you find especially good? Was there anything?”*

User #1: *“The tasks were pretty straight forward, except of the bundling, which involved some extra steps. It was quite handy. I was not aware first that the IDE is actually communicating and interacting with an external marketplace in the background without me noticing, but it seemed that the publishing worked pretty straight forward – it seems very nice. In my experience it can be rather difficult to publish a service in an open marketplace like that. So, if it stays that easy, it would be a very handy thing to have.”*

Moderator: *“Was there anything that was especially bad?”*

User #1: *“The documentation was not 100% intuitive, and without Maven background and without IntelliJ background, which I had before, I assume that I would have had more difficulties in locating specific elements and packaging the bundle. But this cannot be avoided I think, because it’s just so that if you are familiar with a tool it’s good but if you’re not you have to get used to it, if you want to work with it. So, you cannot really get around this, but the documentation could be a bit better formulated and more intuitively designed. Ah, yes, and the delete-button placement was not intuitive at all. Because it was on the menu option that stated that it persists a service and creates the service.”*

Test-user User #3 (Test-Session 2)

Moderator: *“So, how was it?”*

User #3: *“Interesting. I was the first time in an evaluation and testing situation and I’m a little bit nervous and thus did not always remember the things that I’ve read before – that*

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 60 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

with the URL for example. I was a bit lost in the description of the task. It's complicated for the first time."

Moderator: *"Was there anything that you find especially good?"*

User #3: *"The service creation in IntelliJ with a separate plugin or menu structure so that you can access and find the needed methods in an easy way. And the easy way, how you can publish a service to the marketplace. But the other way around, it should be more clear how to unpublish or delete again. So it was a bit unclear to go again in the menu for the persistence and delete the service there."*

Moderator: *"Was there anything that you found to be bad, apart from this issue with the deletion of the service?"*

User #3: *"Yes, the mixture of the items: so that you first have to select to publish, so that you have a distinct menu for publishing and also included the deleting functionality. So that should be in a separate menu entry. That would be more clear to say 'manage the market place services' or 'manage the published services' so that you can go to your service and say that you want to delete it or suspend it."*

User #3*: *"It's really interesting, and the system might be good to use, once the manual is a bit more elaborated... For a moment I was kind of puzzled, when I saw the dummy-text in one of the chapters and I thought that I've seen this 'lorem ipsum...' text before... ☺ [test-user laughs]"*

(*Translation from the German part of the discussion after the "official" end of the session)

Test-user User #4 (Test-Session 3)

Moderator: *"First question: How was it?"*

User #4: *"Rather easy, besides some minor features like the counter-intuitive search interface, because normally you would expect that it works case-insensitive. But nevertheless I had no ... All task were pretty easy."*

Moderator: *"Was there something especially good?"*

User #4: *"It simply worked as intended I guess, so there is nothing pretty awesome and nothing really bad, besides the search interface, but nevertheless it worked o.k."*

Moderator: *"Any other feedback from your side?"*

User #4: *"Actually the interface, as I've seen so far, if I would develop an application for SIMPLI-CITY I would use this interface"*

Test-user User #5 (Test-Session 4)

Moderator: *"So User #5, how was it?"*

User #5: *"I do not know. My experience in service creation is not really impressive, but it was easy for a person who sees this for the first time. Or, let's say, quite easy."*

Moderator: *"What was the best thing? What do you think was good?"*

User #5: *"The tutorial was good. So, because even I did not know about what is going on here, I could go on and perform."*

Moderator: *"Was there anything especially bad?"*

User #5: *"Nothing especially bad. Only that I forgot how to write an interface"*

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 61 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Moderator: *"What would you, let's say fix or improve, if you would be the software developer who created this stuff?"*

User #5: *"Probably I would improve the menu. 'Bundle service and persist' that's not really obvious that the delete-button is in there. That's the only thing."*

Moderator: *"Nothing else to improve?"*

User #5: *"Oh yes: it was hard for me to see this 'yellow message' – it was not really yellow..."*

Test-user User #6 (Test-Session 5)

Moderator: *"How was it?"*

User #6: *"Nice and fun."*

Moderator: *"Was there anything in the system that you found to be especially good?"*

User #6: *"Yes, I found it good that IntelliJ automatically uploads the service into the marketplace, and that there are no extra tasks that have to be done. So, I like it when everything is done in one programme, when there's only a button and it works."*

Moderator: *"Was there anything especially bad?"*

User #6: *"I think the view could be a little bit over-full, if there are many apps in there."*

Moderator: *"So you mean the web view of the Service Marketplace?"*

User #6: *"Yes. So this could be a little bit improved with an extra view."*

Moderator: *"Anything else that you would improve, if you would be the developer of this stuff?"*

User #6: *"Yes, the help how to bundle the service, how to upload the service – First I did not get that I had to push that 'bundle it' button. Maybe that could be a little bit more clearly mentioned or there could be another structure of the help."*

Test-user User #7 (Test-Session 6)

Moderator: *"I've got a few questions. How was it?"*

User #7: *"Well in such a usability evaluation you concentrate also on what you say, so you concentrate on various tasks simultaneously, such as understanding the technology, speaking out loud – so you sometimes can't actually grasp what your task is. You rather concentrate on the process than on the actual result. And as far as the product goes, well I saw that there is a plugin that extends IntelliJ, but I did not actually understand what it's for; o.k. there is an App Marketplace and I can deploy services in there. That's all the information that I get of what the actual value is of this plugin and of the service platform."*

Moderator: *"Would you say that there is something that is really good?"*

User #7: *"It's easy. The buttons, the menu is very structured and the icon-integration is good, so that the icons are prominent and you can see the new plugin and the functionality of the plugin. The deployment process itself could probably be simplified, cause basically the process is 'create a jar file and deploy it' – there's no necessity to have this step in between that you have to compile it with Maven first and then deploy it – you could just have a button 'deploy my thing' and the process does it for you. But, yes I like the... the*

IntelliJ integration was good. But the value seems small, it did not appear that there was any clear value for me the system has to deliver.”

Moderator: “*Was there anything that you would rate as especially bad?*”

User #7: “*The deletion process of the service was hm... I think there could be some menu integration for the popup for the persists ...hm for this ‘persist service’. I think the popup is not necessary, so I did not like that, I think it would be easier to set up a context pane in IntelliJ.*”

8 Usability Evaluation of the Developers' Part of the App Marketplace

This chapter presents the results of the Usability Evaluation of the SIMPLI-CITY Applications Marketplace from developers', testers' and approvers' point of view.

In Section 8.1 the Usability Evaluation approach is described briefly. Section 8.2 gives a summary of the results of the Usability Evaluation, and Section 8.3 outlines the findings in detail. Furthermore, Section 8.4 includes the Task Sheets, which were provided to the test-users during the evaluation sessions, and Section 8.5 provides the transcription of the Debriefing Interviews conducted with the test-users at the end of the sessions.

8.1 Approach taken for the Usability Evaluation

8.1.1 Aim of the Usability Evaluation

This Usability Evaluation was done in order to find out how easy it is for developers to use those parts of the SIMPLI-CITY Application Marketplace, which are targeted at software developers. The results of this Usability Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

8.1.2 Method Applied

For the Usability Evaluation of the SIMPLI-CITY "Applications Marketplace", the well-established Usability Evaluation method "*Thinking Aloud Test*" was applied:

A test-user is asked to speak out loud all her/his thoughts while using the system to complete a given test-task. By observing the test-user's way of working on the task, and by analysing the reasons for confusion, hesitation, or mistakes of the test-user, valuable hints for improvement of the usability of the system can be obtained.

8.1.3 Team of Facilitators

Three persons formed the team, which organised and conducted the usability evaluation sessions of SIMPLI-CITY's "Applications Marketplace".

Table 13: Team of Facilitators of the Usability Evaluation Sessions

| Name (Organisation) | Role within the Facilitators' Team |
|-----------------------------|--|
| Xavier Cases Camats (WORLD) | "Moderator" (guided the test-users through the session) |
| Sergi Martinez (WORLD) | "Observer" (took notes of revealed usability issues) |
| Xavier Febrer (WORLD) | "Technician" (ensured proper functioning of the technical equipment, software, and recordings) |

8.1.4 Usability Evaluation Sessions

The Usability Evaluation Sessions took place at Worldline, Avinguda Diagonal 200, Barcelona, Spain. The 4 usability sessions were conducted on Friday, 17.07.2015 and on Monday, 20.07.2015. The average duration of the Usability Evaluation Sessions was 45 minutes.

All Usability Evaluation Sessions followed the same process:

First the moderator welcomed the test-user, introduced the team, and explained the purpose of the session. The moderator gave an overview of the test procedure, and asked the test-user for her/his permission that the session can be recorded (screen recording), and these records can be used for later analysis and reporting purposes. After the test-user signed the consent form, the moderator encouraged the test-user to ask questions, whenever anything is unclear. In a short introductory interview, the moderator obtained from the test-user some demographic information (gender and age) and some information regarding relevant skills and knowledge of the test-user, such as e.g., familiarity with App Stores, Android, Smartphones, knowledge of SIMPLI-CITY, and expertise with Thinking Aloud tests.

After this introductory part, the moderator explained to the test-user, how the “Thinking Aloud” method works, and asked the test-user to do one test-task after the other. While the test-user was working on the task, she/he spoke out loud all her/his thoughts and considerations, the team of facilitators watched the test-user silently, and the observer took notes of the test-user’s activities and any usability issues that became apparent.

After finishing the last test-task, the moderator thanked the test-user for her/his cooperation and asked the following three questions in the debriefing interview: “How was it?” “Was there anything that you found to be especially good?” “Was there anything that you found to be especially bad?” When the test-user finished answering these questions, the moderator asked her/him to fill-in the System Usability Scale (SUS), a standardised questionnaire to assess a user’s perception of the overall usability of a system. In case the test-user had further questions these were answered by the team, and after that the session was finished.

After each session, the recordings were saved, and the materials and setup was prepared for the next session.

8.1.5 Test-Users

In total 4 test-users participated in the Usability Evaluation of the SIMPLI-CITY “Applications Marketplace”. All of them answered the introductory questionnaire, did the 5 test-tasks, and provided their feedback in the debriefing interview and feedback questionnaire.

Table 14 gives an overview of the answers of the “Short Introduction Questionnaire” stated by each test-user.

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 65 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Table 14: Overview of the Test-Users' Introduction Questionnaire

| Test-User | User #1 | User #2 | User #3 | User #4 |
|--|--------------------|--------------------|--------------------|------------------------|
| Usability Eval. Session | Session 1 | Session 2 | Session 3 | Session 4 |
| Gender | Male | Male | Male | Male |
| Age | 20-30 | 20-30 | 20-30 | 20-30 |
| Peculiarities | None | None | None | Glasses/Contact lenses |
| Mobile Phone Ownership | Android Smartphone | Android Smartphone | Android Smartphone | Other Smartphone |
| Smartphone Usage since | > 1 year | > 1 year | > 1 year | > 1 year |
| Smartphone expertise | High | High | Medium | High |
| Android expertise | High | High | High | Low |
| Usage of Apps | High | High | Medium | High |
| Experience with App Store | High | Medium | Medium | High |
| What do you know about SIMPLI-CITY? | Heard of | Heard of | Nothing | Nothing |
| Ever participated in a Usability Study (Thinking Aloud Test) | No | Yes (Yes) | No | Yes (No) |

Three test-users self-assessed their expertise of Android, smartphones, usage of apps and experience with App Stores was high. One stated that he has a medium expertise. All test-users are still studying Computer Engineering, so their expertise on the 'core' of applications may be limited.

None of the test-users was involved in the development of the SIMPLI-CITY Applications Marketplace. Two of the test-users have heard of SIMPLI-CITY before.

Two of the test-users had participated in a Usability Study, but only one of these in a Thinking Aloud Test.

8.1.6 Evaluation Tasks

The test-users were asked to complete the following tasks:

- Task 0: Create an application, upload the binary and media needed into the SIMPLI-CITY Application Marketplace and submit for the approval (role: developer).
- Task 1: Create a new set of tests and assign it to the created application in Task 0, assign it to a tester and submit for testing (role: approver).
- Task 2: Perform the tests created in Task 1 and submit the results (role: tester).
- Task 3: Check result of the testing and approve the application for publication (role: approver).

- Task 4: Publish the application (role: developer)

Each of the tasks was explained just before it was performed, in order to ensure that the test-users have a clear idea on what was expected from them.

8.2 Summary of the Results of the Usability Evaluation

The usability evaluation demonstrated that a developer can publish an application successfully with a minimum of effort and knowledge. Nevertheless, a number of important usability issues were discovered during the evaluation sequence. In general the evaluated Application Marketplace was accepted quite well by every user. Also we can see that the platform is more useful and easy to use on each interaction.

The average task completion time (or "Time on Task", TOT) for task 0 was 4 minutes and 10 seconds, while the fastest participant required only 2 minutes and 30 seconds and the slowest required 6 minutes and 8 seconds. Further, while some users were able to complete that task on their own (50%), others needed minor help from the technician.

In the task 1 the participants were asked to create and assign a new set of tests to the application created in the previous task. On average this task was completed in 5 minutes and 47 seconds, while the fastest participant needed 4 minutes and 52 seconds and the slowest required 6 minutes and 33 seconds. However, one of the test-user needed a hint to complete the task.

Tasks 2, 3 and 4 were completed successfully by all the test-users within around 1 minute on average. It can be seen that the complexity of the tasks was lower and the design, the usability and the previous experiences of the users helped them with completing these tasks.

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed three features of the SIMPLI-CITY system, which they've especially liked:

1. The single screen form to create an application.
2. The simplicity to approve an application.

However, the test-users mentioned also some aspects of the SIMPLI-CITY system, which should be improved:

1. "+ Create App" button functionality and position.
2. Confusion between "Add binary" button and "Upload binary" button.
3. The place of the "Submit" button on the publication of an application.
4. The drag and drop in "Tests Management" tab is not intuitive.
5. Differences between test and group test are not stated/clear.
6. "Publish" button was hard to find.

After the Usability Evaluation, the results of the evaluation sessions were communicated to SIMPLI-CITY's developers. During the remaining time of the project, their work is focused on solving these issues and improving the developed software.

8.3 Detailed Documentation of the Findings

8.3.1 Usability Metrics

8.3.1.1 Task Completion Success Rate

Overall performance of each participant was evaluated depending on his/her ability to finish the task on his/her own. The task success rate score is the number of participants that finished the task successfully divided by the total number of participants.

We can see that Task 2, Task 3 and Task 4 have the highest completion rate due to the simplicity of the tasks and the familiarity with the evaluated system. Task 0 had a 50% completion rate due to a bad choice on selecting one of the two buttons to submit an application. One of them drives the user to fill up complex information that the users did not know. In Task 1, one of the users gave up and asked for help, since he had run out of ideas.

Table 15: Task Completion Success Rate

| Participant | Task 0 | Task 1 | Task 2 | Task 3 | Task 4 |
|--------------|--------|--------|--------|--------|--------|
| User #1 | ✓ | ✓ | ✓ | ✓ | ✓ |
| User #2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| User #3 | - | ✓ | ✓ | ✓ | ✓ |
| User #4 | - | - | ✓ | ✓ | ✓ |
| Success Rate | 50% | 75% | 100% | 100% | 100% |

✓ = successfully completed;

- = not completed / help needed for completion

8.3.1.2 Time on Task (TOT)

The amount of time needed by each participant to finish their tasks depended on their background knowledge about learning on the platform, task description and quality of user interface.

Task 0, required participants to create an application by filling up the name, description, language, target countries and keywords, upload a promotional image, and submit the binary file and adding the media (icon and screenshots). On average this task took 4 minutes and 10 seconds, while the fastest participant required only 2 minutes and 30 seconds and the slowest required 6 minutes and 8 seconds.

Task 1 asked participants to create a new set of tests for the application created in the previous task. Then they should assign it to the application and assign also the user that will perform these tests. The participants should be able to create the tests, set them into a group and assign the group to an application. On average this task was completed in 5 minutes and 47 seconds, while the fastest participant needed 4 minutes and 52 seconds and the slowest required 6 minutes and 33 seconds.

Task 2 required the users to perform the tests created in the previous task. On average this task took 1 minute and 28 seconds, while the shortest time was 1 minute and 1 second and longest time was 1 minute and 51 seconds.

Task 3 asked the participants to check the results of the tests and approve the application even some tests were in fail state. On average this task was completed in 47 seconds, with a fastest time of 44 seconds and a longest time of 1 minute and 5 seconds.

Task 4 was designed to users find the “Publish” button. On average this task was completed in 48 seconds, with a fastest time of 41 seconds and a longest time of 57 seconds.

Table 4: Participants’ Time-On-Task

| Participant | Task 0 | Task 1 | Task 2 | Task 3 | Task 4 |
|--------------------|---------------|---------------|---------------|-------------|-------------|
| User #1 | 2’30’’ | 5’15’’ | 1’40’’ | 44’’ | 50’’ |
| User #2 | 2’57’’ | 4’52’’ | 1’51’’ | 48’’ | 44’’ |
| User #3 | 6’8’’ | 6’27’’ | 1’1’’ | 32’’ | 41’’ |
| User #4 | 5’6’’ | 6’33’’ | 1’19’’ | 1’5’’ | 57’’ |
| Average TOT | 4’10’’ | 5’47’’ | 1’28’’ | 47’’ | 48’’ |

8.3.1.3 Rating of the Overall System-Usability

After the completion of all tasks, the participants were asked to fill in a standardised feedback questionnaire (SUS System Usability Scale).

As already explained in Section 7.3.1.4, SUS yields a single number representing a composite measure of the overall usability of the system. SUS scores have a range of 0 (“not usable at all”) to 100 (“perfectly usable system”).

Table 16: Participants’ Rating of the Overall System-Usability (SUS Score)

| Test-User | Test-User’s Rating of the 10 Questions in the SUS questionnaire | | | | | | | | | | SUS-Score ⁶ |
|-----------|---|-----|-----|-----|-----|-----|-----|-----|-----|------|------------------------|
| | Q 1 | Q 2 | Q 3 | Q 4 | Q 5 | Q 6 | Q 7 | Q 8 | Q 9 | Q 10 | |
| User #1 | 3 | 2 | 3 | 3 | 3 | 1 | 4 | 1 | 3 | 1 | 70 |
| User #2 | 2 | 0 | 2 | 1 | 3 | 0 | 4 | 2 | 2 | 0 | 82.5 |
| User #3 | 3 | 1 | 4 | 0 | 4 | 0 | 3 | 1 | 3 | 0 | 87.5 |
| User #4 | 2 | 3 | 2 | 1 | 3 | 1 | 3 | 1 | 1 | 3 | 55 |

Analysis of the answers of the test-users to the feedback questionnaire reveals that the SIMPLI-CITY “Applications Marketplace” has got a SUS score of 73.75. According to

⁶ According to User #0 Brooke „SUS – A quick and dirty usability scale”

(<http://hell.meiert.org/core/pdf/sus.pdf>): “To calculate the SUS score, first sum the score contributions from each item. Each item’s score contribution will range from 0 to 4. For items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10 the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of the SUS score. SUS scores have a range of 0 to 100.”

Bangor A. et al⁷, this SUS score can be translated to users' perceiving that the usability of the system is "acceptable / o.k." This is quite good for a prototype system, but indicates that there's still some improvement of the system's usability necessary.

8.3.2 Main Positive Aspects Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed three features of the SIMPLI-CITY system, which they've especially liked:

- The single screen form to create an application.

User #2: *"I liked the one screen form creation page. It is a pain when you have to scroll for completing fields, or you are in the middle of a wizard."*

- Going through the tests.

User #4: *"It was like a quiz, very easy, and very fast."*

- The simplicity to approve an application.

User #2: *"Very easy to manage to give the approval to an application, I would expect a more complex procedure."*

User #1: *"I find how to do it pretty fast, so I think this is a good think. Sometimes you need to get use to the platform before do something, it does not happened to me doing that."*

8.3.3 Main Usability Issues Revealed During the Evaluation

8.3.3.1 Main Usability Issues Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users mentioned several aspects of the SIMPLI-CITY system, which should be improved:

- "+ Create App" button

User #1: *"The button is confusing, I did not know if I should to click it before or after the completion of the fields."*

User #2: *"The position of the button should be placed in the bottom-left corner, following the standards."*

- Confusion between "Add binary" button and "Upload binary" button

User #4: *"I did not know what button push, I thought that first we should add a binary and then update it, but when I pressed "Add binary" button, I did not know how to go back."*

User #3: *"I did a bad choice with "Add binary" button, but it should be nice to have a message explaining what is "Add binary" for..."*

- The "Submit" button on the publication of an application

⁷ Adjective ratings related to SUS scores: 25 = "worst imaginable", 38 = "poor", 52 = "acceptable / o.k.", 74 = "good", 85 = "Excellent", 100 = "best imaginable" (Source: Bangor A., Kortum P., Miller J.: "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale", in Journal of Usability Studies JUS, Vol. 4, Issue 3, May 2009, pp. 114-123)

User #2: *"I think you should place the "Submit" button in a place more visible/accessible, probably outside the binaries tab..."*

User #3: *"The button "Submit" should be placed in a more general place."*

- The drag and drop in "Tests Management" tab

User #3: *"The drag and drop was anything but intuitive."*

User #1: *"I figured it out the drag and drop in the "Test Management" by test and fail, it was been hard to find out that you should drag from the name and not from the icon."*

User #4: *"The zone to drop should be marked once you are trying to drag something, this way it will be more intuitive."*

User #2: *"I could not understand why dragging from the icon it did not work. Strange!"*

- Differences between test and group test

User #2: *"I could not understand why a test should be added to a group."*

User #4: *"I expected to find the tests created in the page to assign to the application even if I did not want to add it to a group."*

User #1: *"I would expect a wizard to create the tests, so then will not be confusion between groups and tests and so on..."*

- "Publish" button was hard to find

User #1: *"It was confusing, I did not know where to find the "Publish" button, and it was there, next to the binary, I would place it outside."*

User #3: *"I guessed that the "Publish" button was inside the binary tab, because the "Submit" button was there in the creation of the application."*

User #4: *"I noticed that the "Publish" button was in the binary tab because other applications had a green check with "published" next to it in the binary column!"*

User #2: *"I would expect to find the "Publish" button outside, next to the list."*

8.3.3.2 Main Usability Issues Noted by the Observer

In the following, the main usability issues, as noted by the observer during the test are listed (in the order of occurrence):

Task 0: "Submit/Add the new binary"

- The actions to be done to upload the binary should be sequential or simplified.

Task 1: "Create a set of tests"

- The user interface does not tell you how to proceed.
- The users do not know if they have to create a "group", and what is the meaning of it.
- The users do not know how to assign the tests to a group.

Task 2: "Assign tests/user to application"

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 71 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

- The users try to drag and drop from tests to users, from users to tests, and even in zones from the applications that are not in the droppable section.

After the Usability Evaluation, the described observations were provided to the respective developers and they were actively working on it in order to improve the provided functionality and resolve the detected issues.

8.3.3.3 Screenshots Documenting Potential Usability Issues

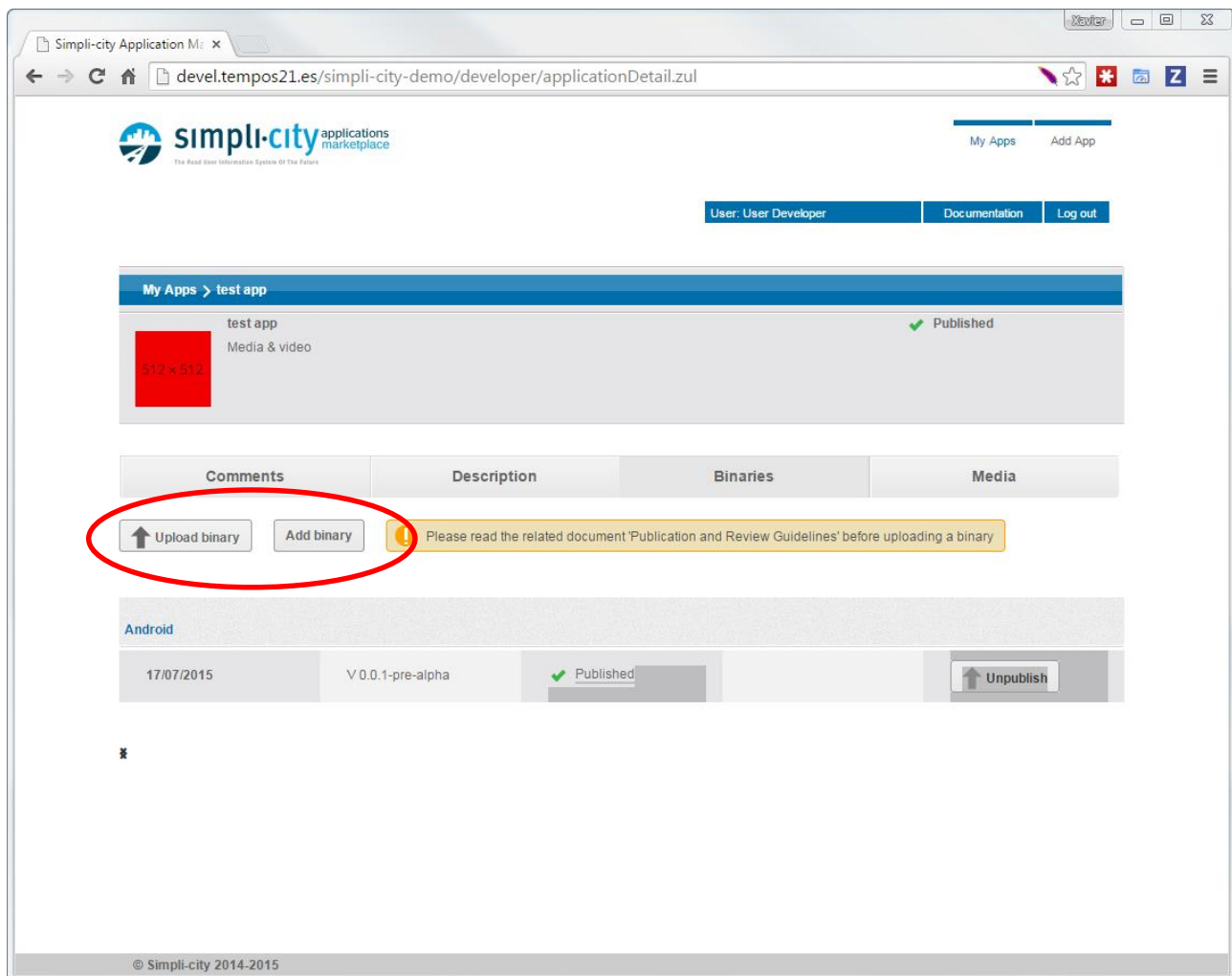


Figure 8: Submitting binary Issue during Task 0

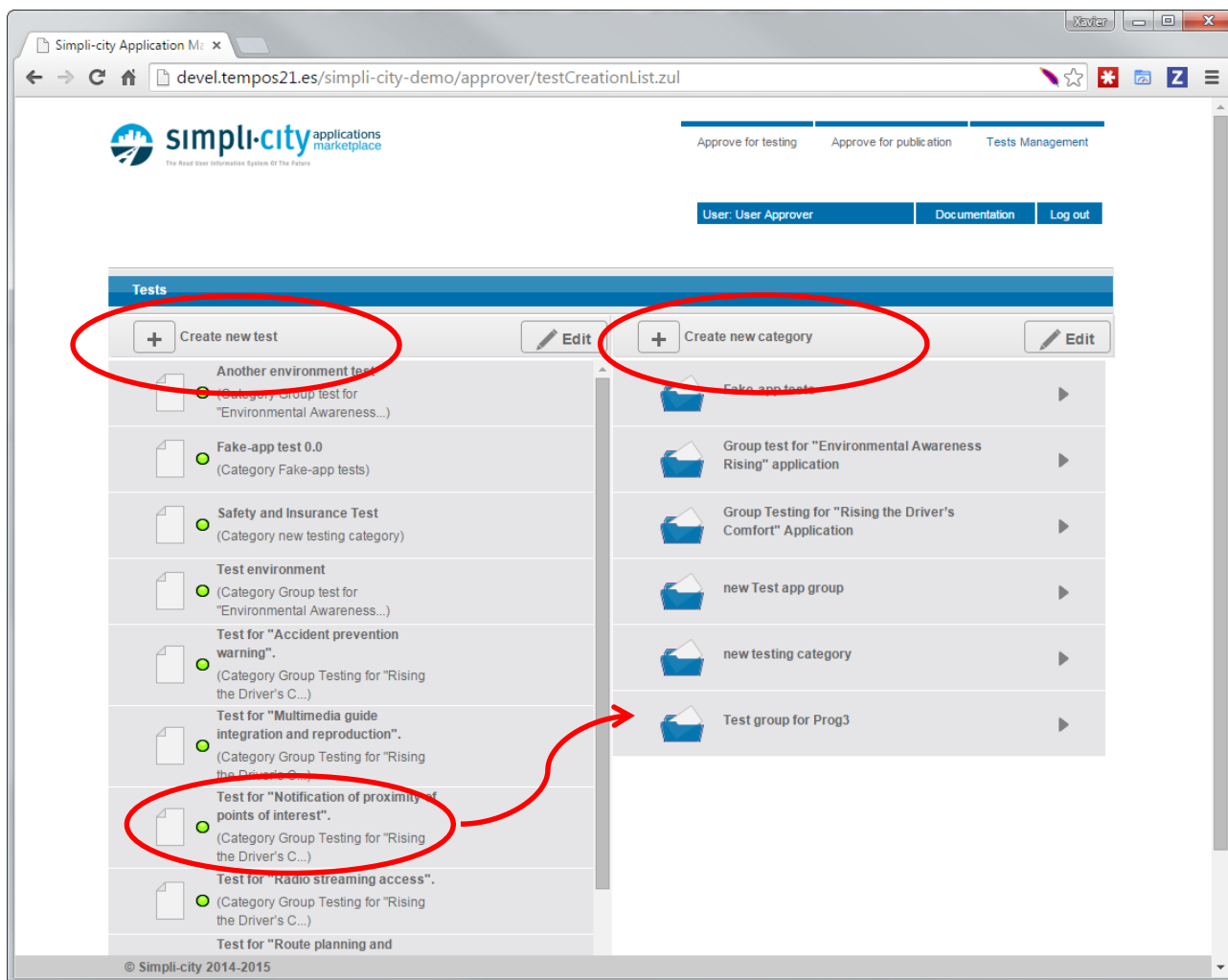


Figure 9: Creation of tests Issue during Task 1

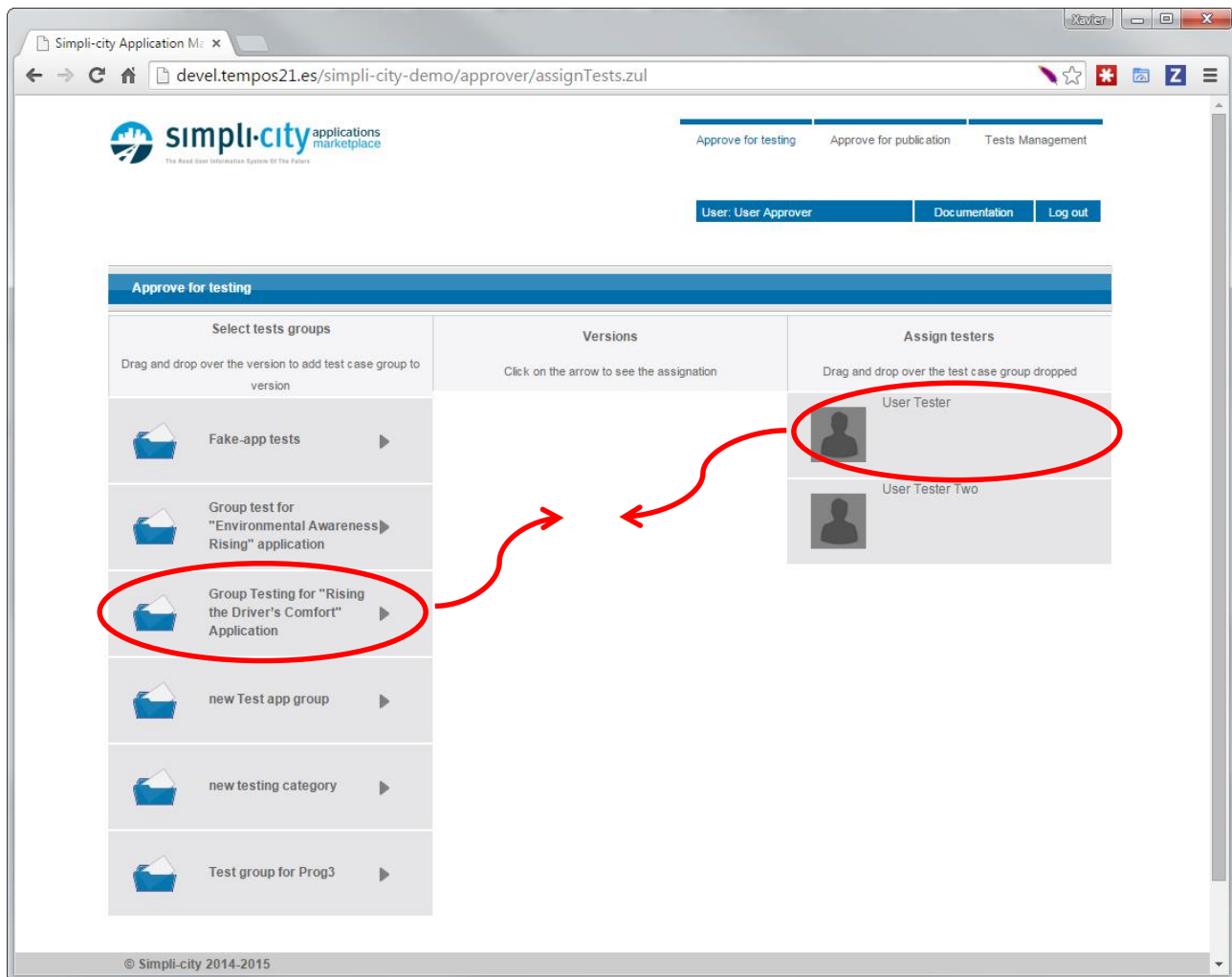


Figure 10: Assign tests/user during Task 1

8.4 Task Sheets for Usability Evaluation of the App Marketplace from Developers' Point of View

TASK 0:

Create/add a new application.

1. Fill in all needed fields/media for a new app.
2. Submit the binary and the media needed.
3. Submit the application for approval.

Tell the moderator, when you finish this task.



TASK 1:**Create a new set of tests and assign it to an application.**

1. Create a new set of tests.
2. Assign the tests to the application.
3. Assign a user to do the testing.
4. Accept it as “ready for testing.”

Tell the moderator, when you finish this task.

**TASK 2:****Perform the tests.**

1. Perform the tests.
 - a. Add comments if needed.

Tell the moderator, when you finish this task.

**TASK 3:****Approve the application as valid for publication.**

1. Check results of testing (if needed).
2. Approve the application for publication.

Tell the moderator, when you finish this task.

**TASK 4:****Publish the application.**

Tell the moderator, when you finish this task.

8.5 Transcription of the Debriefing Interviews

User #1 (Test-Session 1)

Moderator: *"How was it?"*

User #1: *"All right. All the information needed to perform the tasks was explained step by step and it was very clear."*

Moderator: *"What was the best thing?"*

User #1: *"For me, the best thing is the overall functionalities offered by the platform for all the roles: developer, approver and tester... in a single platform in a centralized way."*

Another thing to point out is that the learning curve is quite fast. It is true that some tasks were more difficult and complex than others and facing a new environment, unknown and with a lot of functionalities is always complicated, at least the first time. With that I do not want to say that is a complex platform because once you did a task, do it again is pretty easy."

Moderator: *"And what was the worst thing?"*

User #1: *"The worst thing was the first contact with the application, you face an unknown environment and you have to be focused to understand what to do in each moment to accomplish the task. As I said before, this only happens at the beginning. I think that with 1 hour I will be able to master the platform."*

User #2 (Test-Session 2)

Moderator: *"How was it?"*

User #2: *"Most of the time the application was intuitive enough to guide me without much effort, but there were also some tasks (specially the creation of tests) that felt a little bit frustrating without any help."*

First of all the tab changing (from open tests to tests management or something similar) was not intuitive (and the fact that the tab's title is very small just adds difficulty up), then you have to make up for yourself that you need to create a new TestGroup and that a single test can contain more than one question."

Aside from this, which was (answering the third question) the worst part, the application felt messy, the colour scheme and the iconology are too simplistic, It all felt the same: very plane, very boring. This does not influence the core functionality but it does not motivate you to use repeatedly the service."

Moderator: *"What was the best thing?"*

User #2: *"In my opinion the best part was the idea of cooperative working (developer tester and approver all working side by side, dividing the work). If that does not count, I'll keep the fact that when you (as a developer) are creating a new app, all of the fields you have to fill are visible at the same time without any need to scroll down, creating a feeling that it's all compact, all important, all easy."*

Moderator: *"And what was the worst thing?"*

User #2: *"It may be deuced between the previously mentioned messy aspect of the test creation task and the position of the 'publish app' button inside the 'binaries' tab in developer role."*

| | | | | |
|---|--------------------------|--|----------------------|-------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 76 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Aside from this, I had the overall feeling that I wouldn't have known I had already finished the task if it was not for external help. I lacked some 'operation succeeded' pop-ups for example."

User #3 (Test-Session 3)

Moderator: *"How was it?"*

User #3: *"The product is very good, it has a simple user interface and is easy to use. Still needs to be improved, but it could be very competitive."*

Moderator: *"What was the best thing?"*

User #3: *"The user interface, it was very intuitive."*

Moderator: *"And what was the worst thing?"*

User #3: *"The worst is the first time you use the platform, some buttons are confusing, but I am sure after the first time there will not be problems."*

User #4 (Test-Session 4)

Moderator: *"How was it?"*

User #4: *"I found the platform very useful, but I think that new prototypes have to be developed in an iterative way to get an optimal user experience, or at least better than the actual, for me..."*

Moderator: *"What was the best thing?"*

User #4: *"What I liked most of the platform was the coherence of the spatial distribution and colour. We can take the navigational menus as an example (on right-top), that makes it easy to learn. In the other and, using only two colours with a not overwhelming, simple design brings lots of free space."*

Moderator: *"And what was the worst thing?"*

User #4: *"That if you are not familiar with the functional process of the tasks (ie., if you are a young developer and it is your first time attempting to submit an application), the process and/or the information of help given is very limited to understand how to do all the steps correctly."*

9 Usability Inspection from Developers' Point of View

This chapter presents the results of the Usability Inspection of the SIMPLI-CITY system from software developers' point of view.

In Section 9.1 the Usability Inspection approach is described briefly. Section 9.2 gives a summary of the results of the Usability Inspection, and Section 9.3 outlines the findings in detail.

9.1 Approach Taken for the Usability Inspection

9.1.1 Aim of the Usability Inspection

This Usability Inspection was done in order to find out how easy it is for developers to learn by exploration to use the SIMPLI-CITY tools targeted at software developers. The results of this Expert Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

9.1.2 Method Applied

For Expert Evaluation of the SIMPLI-CITY “developer package”, the well-established Usability Inspection method “*Cognitive Walkthrough*” was applied:

The usability inspection method “Cognitive Walkthrough” focuses on the learnability of a system, and is designed to find gaps and problems a user might have in determining the next appropriate step for completing a task.

A team of evaluators examines a system by stepping through the exact sequence of actions, which are necessary to complete a user's task. For each action, the evaluators discuss whether or not a novice user would know what to do at this step, and whether or not a novice user would notice that she/he is on the right way, when she/he has done the right action. All problems, reasons and assumptions revealed during this discussion process are recorded, in order to help improving the system.

9.1.3 Team of Evaluators

Table 17: Team of Evaluators for Usability Inspection from Developers' Point of View

| Name | Organisation | Professional Background |
|--------------------------|--------------|--|
| David Turon Bes | TIE | Experienced software developer with background in Java and Scala |
| Fran Rodriguez Montero | TIE | Experienced software developer with background in Java and .Net |
| Arturo Brotons Cartagena | TIE | Experienced Java developer |
| Vadim Petrenko | TIE | Senior software developer with background in Java |

9.1.4 Usability Inspection Sessions

The usability inspection session took place on the 26th of August 2015 at the TIE headquarters. Taking about one hour, there was a plenty of opportunity to discuss different aspects of the development package. Evaluators were given an opportunity to speak and express their opinions, which were recorded for further analysis.

9.1.5 Context Information

| | |
|------------------|--|
| Product | <p>The SIMPLI-CITY “developer package”, which was inspected during this Expert Evaluation, included the following components:</p> <ul style="list-style-type: none"> • Application Design Studio • Service Development API • Service Marketplace • Application Marketplace |
| Users | <p>It is expected that users of this product belong to the following groups:</p> <ul style="list-style-type: none"> • software developers, who programme services • software developers, who programme (Android) applications <p>It is assumed that the users of this product have the following relevant skills, knowledge, and experience:</p> <ul style="list-style-type: none"> • Java standard edition • Android • Experience in developing native Android mobile apps • Understanding of push notifications • Understanding of a client-server model • Ability to learn and use third party APIs |
| Tasks | <p>The main tasks that users perform with this product are:</p> <ul style="list-style-type: none"> • develop an App using the SIMPLI-CITY system • develop a service using the SIMPLI-CITY system • utilise a service from the SIMPLI-CITY Service Marketplace for creating an App • put an App on the SIMPLI-CITY Application Marketplace • put a service on the SIMPLI-CITY Service Marketplace |
| Test Environment | <p>The Usability Inspection was done in the following technical environment:</p> <ul style="list-style-type: none"> • Evaluators used screen sharing to watch the presentation performed by Arturo Brotons in real-time, which he was doing using an ordinary laptop of average configuration. • Development tools being evaluated were running locally on Arturo's laptop and the Service and Application marketplaces were running on Ascora's and WORLD's external development servers. |

9.2 Summary of the Results of the Usability Inspection

As the usability inspection involved evaluation of development tools (Application Design Studio and Service Development API plus SIMPLI-CITY marketplaces), it was taken care that all evaluators invited to the session already had an ample experience with different development tools, IDE (integrated development environments) and so on, which they use in their daily work. Thus it was ensured that they could subjectively and partly objectively judge whether they could use the tools being tested in their real life work. They could compare the tools being discussed with alternatives they have used in the past, and they were able to compare the implementation of standard features, ways of interface organisation, responsiveness, developer friendliness, and other aspects.

The unanimous opinion of the evaluators was positive. They liked what the SIMPLI-CITY development tools allow them to do for development of SIMPLI-CITY services and apps, and how these services and apps can be later marketed on the SIMPLI-CITY marketplaces.

Specifically, the evaluators noted positively, that the development tools are based on one of the best freely available IDEs: IntelliJ IDEA, which together with Eclipse is one of the most widely used IDEs in the industry.

Furthermore, the evaluators noted that the extensions provided for SIMPLI-CITY specific work are logical and conscious, the layout is straight forward, and this makes it easy for developers to use the tools.

The evaluation went smoothly as all features being evaluated worked as supposed. Based on their experience, the evaluators suggested several improvements, which are listed in detail in the subsequent sections of this document. However these suggestions were mostly cosmetic touches and user interface improvements. According to the evaluators, the overall logic of tooling is absolutely adequate and makes it easier for an experienced developer to achieve her/his goals.

9.3 Detailed Documentation of the Findings

The following 5 tasks are typical tasks, which a developer would most likely want to complete using the SIMPLI-CITY system:

1. Develop an App using the SIMPLI-CITY system
2. Develop a service using the SIMPLI-CITY system
3. Utilise a service from the SIMPLI-CITY Service Marketplace for creating an App
4. Put an App on the SIMPLI-CITY Application Marketplace
5. Put a Service on the SIMPLI-CITY Service Marketplace

Therefore, the evaluation team examined these tasks using the "Cognitive Walkthrough" method.

The following subsections list the findings of this Usability Inspection for each of these tasks.

9.3.1 Task 1: Develop an App Using the SIMPLI-CITY System

| Action Sequence for Completion of Task 1 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|---|---|
| Install the App Design Studio in IntelliJ IDEA, following the install instructions in the provided ZIP file | User will know what to do at this step | x | Installing IDE plugins as described is common practice. | While the Application Design Studio can be easily installed from the supplied zip archive, it is worth checking, if the plugin itself can be uploaded to the main IntelliJ plugin repository (where third party plugins are located: http://plugins.jetbrains.com/), then installation would be even easier. |
| | User will know that this was the right action | x | | |
| Open IntelliJ IDEA | User will know what to do at this step | x | Developers start their IDEs daily. | Rearrange menu system of the IDE: put the "SIMPLI-CITY" menu item before the "Help" menu item. |
| | User will know that this was the right action | x | | |
| Create a new SIMPLI-CITY App project following the New Project wizard under [File > New Project...] menu option. | User will know what to do at this step | x | This is a usual step when creating a new project independently of the project nature. | |
| | User will know that this was the right action | x | | |

| Action Sequence for Completion of Task 1 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|---|---|
| Write Maven options (i.e., group ID, artefact ID and version) | User will know what to do at this step | x | Since the majority of modern projects use Maven for project dependency control and building, developers are likely to be already familiar with Maven. | Add tooltips to the Maven option fields, explaining what they are. However, for an experienced developer the fields are self-explanatory. Add F1 or the “Help” button, which would invoke help describing these options. |
| | User will know that this was the right action | x | | |
| Write SIMPLI-CITY options and choose location of the project | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e. know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |

| Action Sequence for Completion of Task 1 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|---|---|
| Make any modifications on the original example | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e. know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | Manifest Editor needs to be stretched to fit the window. Add an example of fully featured app. Now it is a skeleton project containing all needed files, but these files do not contain much logic. It would be better for developer understanding if it included a fully-fledged app. |
| | User will know that this was the right action | x | | Add more comments to the example app files, describing necessary steps in the app code. |
| Generate the bundled App out of the project using [SIMPLI-CITY > Send App to Device...] and then [Build]. | User will know what to do at this step | x | As the developer is ready to try her/his app on the device, she/he will come to this menu option. | |
| | User will know that this was the right action | x | | |

| Action Sequence for Completion of Task 1 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|---|---|
| Copy the bundled App to the Device using [Send Bundle]. | User will know what to do at this step | x | Sending the developed app to the actual device is a logical and expected step. | <p>Rename "Send Bundle" to "Send to device".</p> <p>After clicking on "Restart ADB" (Android Debug Bridge), display a message that ADB has been restarted.</p> <p>When clicking the "Send Bundle" button, display a window with the current upload speed and then a dialog (upload complete).</p> <p>Add confirmation dialog, asking if the developer really wants to send the app to the device.</p> <p>The publish button is missing (together with username/password combination). This functionality is needed for immediate publishing of the app to the marketplace. At the moment this is done using the web UI of the marketplace and not via the plugin.</p> |
| | User will know that this was the right action | x | | |
| Run the Personal Mobility Assistant, or close and run if it was already running. | User will know what to do at this step | x | To test the deployed app, the developer has to run the PMA, which the developer knows how to do as they are familiar with SIMPLI-CITY basics. | |
| | User will know that this was the right action | x | | |

9.3.2 Task 2: Develop a Service Using the SIMPLI-CITY System

| Action sequence for completion of Task 2 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|--|--|
| Install the Service Design Studio in IntelliJ IDEA, following the install instructions in the provided ZIP | User will know what to do at this step | x | Installing IDE plugins as described is common practice. | While the Service Design Studio can be easily installed from the supplied zip archive, it is worth checking if the plugin itself can be uploaded to the main IntelliJ plugin repository (where third party plugins are located: http://plugins.jetbrains.com/), then installation would be even easier. |
| | User will know that this was the right action | x | | |
| Open IntelliJ IDEA | User will know what to do at this step | x | Developers start their IDEs daily. | Rearrange menu system of the IDE: put the "SIMPLI-CITY" menu item before the "Help" menu item. |
| | User will know that this was the right action | x | | |
| Create a new SIMPLI-CITY Service project following the New Project wizard under [File > New Project...] menu option. | User will know what to do at this step | x | This is a usual step when creating a new project independently of the project nature. | |
| | User will know that this was the right action | x | | |
| Choose between Backend, External | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | Add tooltips explaining the difference between these options. However, for a developer knowing basics of SIMPLI-CITY these are self-explanatory. |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 2 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|--|---|
| Write Maven options (i.e., group ID, artefact ID and version) | User will know what to do at this step | x | Since the majority of modern projects use Maven for project dependency control and building, developers are likely to be already familiar with Maven. | Add tooltips to the Maven option fields, explaining what they are. However, for an experienced developer the fields are self-explanatory. Add F1 or the “Help” button, which would invoke help describing these options. |
| | User will know that this was the right action | x | | |
| Write project name and location on local file system | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Open Project Structure (Ctrl+Alt+Shift+S) and choose JDK 1.7 (if not already selected) | User will know what to do at this step | x | This specific step should be memorised by the developer. While not self-evident, it is described in the user manual. | |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 2 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|---|--------------------------------|
| Make any modifications on the original example | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Use [Maven Projects > Your project > Lifecycle > package] to generate the project | User will know what to do at this step | x | Majority of modern projects use Maven for project dependency control and building. Developers are likely to be already familiar with Maven. Described is a standard way of building a project. | |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 2 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|---|--|
| Go to [SIMPLI-CITY > Bundle Service and persist...] and select [Bundle it] | User will know what to do at this step | x | Once a service is developed, the developer naturally wants to deploy it to the Service Runtime Environment for tests. | Rename menu item "Bundle & persist" to "Upload". Use a bigger window for the "Bundle & persist" dialog. Move Service Runtime Environment URL setting to this dialog (from the separate settings dialog). |
| | User will know that this was the right action | x | | Add a list of SRE servers to persist the bundle (if the developer wants to deploy to local or to remote servers, then they just need to select from the list instead of retyping the URL). Rename "Delete service" to "Uninstall service" (this option uninstalls the service from the SRE). Rename "Bundle It" to "Deploy service" (this option deploys the service to the SRE). When clicking "Bundle It" do not hide the dialog window, but just display a confirmation OK dialog on top after bundling is finished. Remove the "Select manifest" option as it is not really needed. When publishing a service to the Service Marketplace, offer the developer a possibility to specify username and password to access the marketplace. |

9.3.3 Task 3: Utilise a Service from the SIMPLI-CITY Service Marketplace for Creating an App

| Action sequence for completion of Task 3 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|--|--------------------------------|
| Open the web browser and go to the Service Marketplace web interface (http://simpli-city.eu:3030/) | User will know what to do at this step | x | The URL has been communicated to the developer. | |
| | User will know that this was the right action | x | | |
| Login with your username and password (Note: at the moment there is no open user registration for the Service Marketplace, login credentials are provided by SIMPLI-CITY personnel on request) | User will know what to do at this step | x | Using username and a password for authentication is common practice. | |
| | User will know that this was the right action | x | | |
| Search for an existing service | User will know what to do at this step | x | Searching by going through the list of services or using a search field is common in modern web sites. | |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 3 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|--|--|
| Look for the ID of the service and the list of methods available | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Open IntelliJ IDEA | User will know what to do at this step | x | Developers start their IDEs daily. | |
| | User will know that this was the right action | x | | |
| Open an existing SIMPLI-CITY App project or create a new one | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e. know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Use an existing example of how to connect to SIMPLI-CITY services to connect to real ones | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e. know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | One of evaluators suggested adding video tutorials to the help section of the Application Design Studio, where using a series of screencasts several important techniques can be described, for instance, connecting to an existing service. |
| | User will know that this was the right action | x | | |

9.3.4 Task 4: Put an App on the SIMPLI-CITY App Marketplace

| Action sequence for completion of Task 4 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|--|---|-----|--|--------------------------------|
| Open IntelliJ IDEA | User will know what to do at this step | x | Developers start their IDEs daily. | |
| | User will know that this was the right action | x | | |
| Open an existing SIMPLI-CITY App project | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Generate the bundled App out of the project using [SIMPLI-CITY > Send App to Device...] and then [Build] | User will know what to do at this step | x | The developer knows the main dialog for working with bundling of apps is located here. | |
| | User will know that this was the right action | x | | |
| Click [Open Location] and copy the full-bundle.zip file somewhere to be used later | User will know what to do at this step | x | This step is described in the manual. | |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 4 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|---|--------------------------------|
| Open the web browser and go to the App Marketplace web interface (https://devel1.tempos21.com/simpli-city-demo/) | User will know what to do at this step | x | The URL has been communicated to the developer. | |
| | User will know that this was the right action | x | | |
| Login with your username and password (Note: at the moment there is no open user registration for the App Marketplace, login credentials are provided by SIMPLI-CITY personnel on request) | User will know what to do at this step | x | Using username and a password for authentication is common practice. | |
| | User will know that this was the right action | x | | |
| Go to Add App, fill all the required fields and add some media images | User will know what to do at this step | x | This step is self-explanatory. | |
| | User will know that this was the right action | x | | |
| Click Create App, then Binaries and Upload Binary and select the previously generated full-bundle.zip | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e. know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |

9.3.5 Task 5: Put a Service on the SIMPLI-CITY Service Marketplace

| Action sequence for completion of Task 5 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|--|--------------------------------|
| Open IntelliJ IDEA | User will know what to do at this step | x | Developers start their IDEs daily. | |
| | User will know that this was the right action | x | | |
| Open an existing SIMPLI-CITY Service project | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |
| Generate the bundled service out of the project using [Maven Projects > Your project > Lifecycle > package] | User will know what to do at this step | x | The developer knows the main dialog for working with bundling of services is located here. | |
| | User will know that this was the right action | x | | |
| Open the location of the bundle, under [Project location]/target and copy the JAR file somewhere to be used later | User will know what to do at this step | x | Developers should already have some SIMPLI-CITY development background, i.e., know how SIMPLI-CITY works, how it is organised and how to develop services and apps within its framework. | |
| | User will know that this was the right action | x | | |

| Action sequence for completion of Task 5 | | Yes | Success / Failure Story | Comment / Idea for Improvement |
|---|---|-----|--|--------------------------------|
| Open the web browser and go to the Service Marketplace web interface (http://simpli-city.eu:3030/). | User will know what to do at this step | x | The URL has been communicated to the developer. | |
| | User will know that this was the right action | x | | |
| Login with your username and password | User will know what to do at this step | x | Using username and a password for authentication is common practice. | |
| | User will know that this was the right action | x | | |
| Go to Add Service, fill all the required fields and add some media images | User will know what to do at this step | | Currently this option [add service] is not available in the Service Marketplace. Instead, the developers can deploy their service to the SIMPLI-CITY SRE (Service Runtime Environment) using the IntelliJ plugin, and then the Service Marketplace will list this service. | |
| | User will know that this was the right action | | | |
| Click Create Service, then Binaries and Upload Binary and select the previously generated JAR file | User will know what to do at this step | | Currently this option [add service] is not available in the Service Marketplace. Instead, the developers can deploy their service to the SIMPLI-CITY SRE (Service Runtime Environment) using the IntelliJ plugin, and then the Service Marketplace will list this service. | |
| | User will know that this was the right action | | | |

10 Functional Evaluation of SIMPLI-CITY from Developers' Point of View

In August / September 2015, during functional evaluation of SIMPLI-CITY, the partners' software developers were asked to express their subjective opinions about using components developed by other partners in the project. This included not only generic thoughts, but also concrete suggestions on how to improve existing components and make them more developer friendly, and also experiences in working around shortcomings, which developers as users of these components had to face. Also improvement suggestions, which go beyond the specifications of the components within the SIMPLI-CITY project, were made by the partners.

The results of this evaluation were communicated to the authors of the described components so that they could get an impression of how useful their components were, and how these components could be further improved. Some of the suggestions were concrete enough to result in a bug or feature request and could be added directly to the Jira bug tracking system for later fixing.

10.1 Detailed Documentation of the Findings

| Partner | Opinion and Comments |
|---------|--|
| TUV | <p><i>Cloud-Based Information Infrastructure for the Service Registry</i></p> <p>The Cloud-Based Information Infrastructure provides a robust API which is accessible via REST. All the defined functionalities are provided, however, it would be good if additional query (or search) functionality would be implemented. For now, only JSON-based queries are allowed.</p> <p><i>External Data Sources</i></p> <p>Requesting data from OpenGovernment, OpenStreetMap, (i.e., external data sources) is quite stable and works as expected. However, since these data sources are outside of the controlled environment, we are not able to cover all errors, e.g., the services are under control of a different provider who might change the API or remove the service at all without prior notice. This will influence the overall stability as errors may appear unexpectedly. Hence, if the project will be continued, contracts to external data sources providers need to be established (SLAs).</p> <p><i>REST Proxy for invoking backend services from outside of the SRE</i></p> <p>The REST Proxy is a well document and robust component. It provides all the requirements as defined in the project documentation. However, although backend services are under control of the SRE, the REST Proxy is not always able to figure out the cause of an error, e.g., if a backend service failed or just returned NULL, the only thing the REST Proxy can do, is forwarding that information to the requester. Hence, it would be good to provide additional error handler or monitoring components. In addition, so far, the REST Proxy does not provide any load balancing. This means, in the case of high load (many parallel requests), the SRE is in danger to overload or even crash.</p> |

| Partner | Opinion and Comments |
|---------|--|
| | <p><i>Application Runtime Environment (ARE)</i></p> <p>The ARE for deploying single Apps on the smartphone is a fully functional component. It provides the means to add additional Apps in an easy way. In addition, its multi-model user interface allows users to interact either via touch or speech with the Apps.</p> <p><i>Dialogue Interface and Multi-Modal UI within the Apps</i></p> <p>The Dialogue Interface within the Apps which are deployed in the ARE on the smartphone provides a novel approach for designing user interactions. By simply providing an additional small backend service running on a server, additional functionality can be added to any App without the need of restarting it. This component is well designed and provides useful feedback in the case of errors. So far, this component provides the required functionality as specified within the project's specification. However, additional functionalities, like providing speech recognition without any backend servers, would be quite useful, this goes beyond the requirements, but would be a great extension for future work.</p> <p><i>Sensor Abstraction component within the Apps which are deployed in the ARE</i></p> <p>The Sensor Abstraction component is a nicely designed component which allows easy access to arbitrary sensors via a single interface, e.g., it provides the means to read the current user's location (phone's GPS), calendar, contact list, as well as the multiple car sensors (fuel level, etc.).</p> |
| ASC | <p><i>Service Registry</i></p> <p>Partner Ascora used the Service Registry while developing the Service Marketplace as well as the Service Development API, which is integrated into the Application Design Studio. In both cases the Service Registry proved to be a fully functional, yet dynamically extendable component which provided all necessary interfaces to access and maintain the list of the available Services to 2nd party partners (project partners). For usage in SOA Environments the Service Registry, as well as its parent component, the Service Runtime Environment itself, are more than feasible to be re-used.</p> <p><i>Sensor Abstraction</i></p> <p>Ascora has used the Sensor Abstraction and Interoperability components during its WP8 work for connecting to car based sensors. The components have proven to demonstrate a good start on how to access sensor systems and on how to get values of them. For further development, an abstraction from the car manufacturer implementations of the OBD protocol would further enhance the component beyond the scope of the project.</p> <p><i>Talkamatic Speech Backend (MMDI)</i></p> <p>While creating the Application Runtime Environment partner Ascora came in contact with the Voice Recognition and Speech Output system of partner TALK. The use of the component turned out to be very easy and straight forward. A good documentation and very good communication with partner TALK helped</p> |

| Partner | Opinion and Comments |
|---------|---|
| | <p>while using the component. Partner Ascora is planning to use this very component for their own WP8 mini-project, which is explained in more detail in the respective document. The components provide an excellent way for applying voice interaction to apps within and beyond SIMPLI-CITY. An enhanced support of additional languages could enhance the components after the project's ending.</p> <p><i>Application Runtime Environment</i></p> <p>For the creation of the ARE Push Service, which is the connection between the Application Runtime Environment and the Service Runtime Environment, the Message Handler component of the ARE had to be used, which proved to be as fully function as expected. Apart from that the Service pulling from the Service Runtime Environment had to be used.</p> <p><i>EcoIndex Calculation</i></p> <p>The EcoIndex concept has also been used by Ascora within the course of the eco contest implementation of WP8. The EcoIndex has turned out to be a good gamification-like approach to quickly summarize the indication of environmental impacts that a driving behavior has. After the project, it could be further increased in popularity by open sourcing the algorithm behind it or by providing its calculation as a service.</p> <p><i>Application Marketplace</i></p> <p>Ascora has used the application marketplaces for experimenting with apps and their listings. Findings have shown an interesting way to publish and promote apps, which even goes beyond the project and could possibly easily be transferred to other use cases including use case specific app marketplaces such as e.g. the eHealth sector.</p> |
| TIE | <p><i>Service Registry</i></p> <p>Partner TIE used Service Registry as part of the Service Runtime Environment in different tasks and especially in the implementation of the Service Development API IntelliJ Plugin / Application Design Studio and for the REST Proxy. The Service Registry component was nice to work with, well documented and concisely organised. Using modern technologies, it was easy to extend when TIE needed extra functionality for its mini project. A desired feature would be to have a full text search in the Registry, however, this has been added as part of the mini project.</p> <p><i>Service Runtime Environment</i></p> <p>TIE used SRE as the container for hosting real backend services (e.g. EcoIndex service) and communicated to it via REST Proxy. Using the OSGi standard underneath, SRE has proven to be a robust, extensible platform. Especially it should be mentioned that a wealth of information and documentation is available on the Internet, as the OSGi platform is widely used. SIMPLI-CITY takes advantages of OSGi functionality such as hot redeployment of services, management functionality and much more, however lacks elasticity and fault</p> |

| Partner | Opinion and Comments |
|---------|---|
| | <p>tolerance, where a new instances of SRE could be started should load increase. Such functionality was foreseen as optional in the original set of requirements, however implementing it would be beneficial especially in the real production used of SIMPLI-CITY.</p> <p><i>PMA (Personal Mobility Assistant)</i></p> <p>The PMA was used by TIE to help other partners implement use case apps and during the work on the Application Runtime Environment (ARE). This component has proven to be quite complex and requiring help from the developing partner. The component needs more maturation and a better set of documentation for smoother integration. Nevertheless, the authors of the component were ready to help and solved many issues.</p> |
| TUDA | <p><i>Service Runtime Environment and the Cloud-based Information Infrastructure</i></p> <p>Partner TU Darmstadt used the Service Runtime Environment and the Cloud-based Information Infrastructure. The SRE component and the Cloud-based Information Infrastructure component have proven to be mature, well documented components, offering robust functionality within a well-defined set of interfaces exposed via REST web services. Deploying TUDA's bundles into the SRE worked without any problems. The given examples have shown to be a very good extension to the documentation. Both components provided an easy way to interact and fulfil partner's tasks.</p> |
| IBM | <p>Partner IBM had a direct interaction with the following two components (only), as IBM is involved in the data layer part (i.e., no interaction with WP5/6):</p> <p><i>Cloud-based Information infrastructure for storing and accessing data</i></p> <p>The Cloud-based Information infrastructure for storing and accessing dynamic, historic and remote data from heterogeneous sensors and actuators. The infrastructure has been demonstrated to be scalable in the context of the WP7 scenario where thousands of data points are collected on an hourly basis. The response time of the infrastructure is as defined during the requirement specification. One of the main next steps to be considered is the synchronisation stage (when accessed / modified / updated by different users in same time) while this was not defined in the specification. The latter limitation is by the way the major limitations of cloud offerings nowadays.</p> <p><i>Sensor abstraction and interoperability interfaces</i></p> <p>The Sensor abstraction and interoperability interfaces to access all data from car sensors. Data is accessed through OBD connection and stored on the Cloud infrastructure. Data is collected dynamically and processed by the data processing component for further analysis and reasoning. Interfaces are all well-defined and documented. The interaction with any car sensors is then easy and straightforward even for non-experts (the documentation is important though since many APIs are exposed). The main limitation (which goes beyond the scope of the project) is the adaptability to different car manufacturers. This is also known as a common limitation of commercial offerings.</p> |

| Partner | Opinion and Comments |
|---------|--|
| WORLD | <p><i>Application Design Studio</i></p> <p>Partner WORLD developed three apps using the Application Design Studio (ADS). The ADS is easy to install and comfortable to use, with clear instructions in the help system. Adding the possibility of debugging code in the future will be very important for the development process.</p> <p><i>Personal Mobility Assistant (PMA), Multimodal Dialogue Interface (MMDI)</i></p> <p>The PMA (and basically MMDI) still needs some development for being a mature product, as some user interfaces are difficult to create or WORLD had to use some workarounds in order to get the desired result. This is a normal process of developers needing elements that were not initially planned, and should evolve altogether with developer engagement.</p> <p>The MMDI is probably the most complex part to use, not only technically but conceptually. The learning curve is really high, so having a good documentation and samples is the key for this part.</p> <p>This last advice is a recommendation for the whole project: in order to engage developers to work within the SIMPLI-CITY environment a better documentation should be developed. WORLD had to invest a lot of time learning how things work (and also received a lot of help from other partners), but for external developers, having clear documentation, tutorials, samples or similar elements will be really important.</p> |
| CRF | <p><i>Personal Mobility Assistant (PMA), Multimodal Dialogue Interface (MMDI)</i></p> <p>Partner CRF developed an App that includes functionalities in two languages: Italian and English. The PMA was used for building the multimodal interface in both languages.</p> <p>The main difficulties were due to the limitations of the GUI that has forced developers to redesign some interactions (i.e., it was impossible to visualize image and text in the same screen, etc.). Moreover, for the setting up of the dialogue interaction it was necessary to invest more resources than initially planned.</p> <p>The way of setting up things to make the whole MMDI interface fully working is quite complex and in CRF case a lot of support from partner TALK's staff was needed. Some samples provided were of great support, but further documentation and guidelines are necessary for allowing external developers to use this component easily. In CRF view the set up needed should be simplified (otherwise a lot of files, configuration, etc. needed to be provided).</p> <p><i>Sensor Abstraction</i></p> <p>CRF also extended sensor abstraction for getting access to the own car sensors. The available documentation together with minor support from partner TUDA was sufficient for the completion of this task.</p> |

Partner TALK did not use any of the SIMPLI-CITY components directly in its work, and therefore abstained from evaluation.

Since the partners FGM and SRM were not a developing partner, FGM and SRM did not participate in this functional evaluation of SIMPLI-CITY's developer tools.

10.2 Summary of the Results of the Functional Evaluation from Developers' Point of View

In general, partners were positive about the SIMPLI-CITY components they had to use in the course of development. This comes to no surprise as the Functional and Technical specifications produced in the beginning of the project in cooperation with all partners were thought out in the smallest details, laying out specifics of interfaces and underlying technologies that partners wanted to be used. The majority of comments, which were received from partners during this functional evaluation of SIMPLI-CITY's developer tools, is related to possible future enhancements, some of which were considered as optional "nice to have" features in the earlier stages of the project and were skipped during the implementation phase, because the development of other more important "must have" features occupied time and resources.

To summarise, it can be said that apart from the fact that the documentation and guidance could be improved, overall SIMPLI-CITY partners' software developers were quite satisfied with the way they had to work to develop within the SIMPLI-CITY framework. The organisation of components and the overall architecture had proven to be up to the expected industry standards.

11 Usability Inspection of the Road Users' Part of the App Marketplace

This chapter presents the results of the Usability Inspection of the SIMPLI-CITY App Marketplace from road-users' point of view. This Usability Inspection was done in July/August 2015 in Bologna and Graz.

In Section 11.1 the Usability Inspection approach is described briefly. Section 11.2 gives a summary of the results of the Usability Inspection, Section 11.3 outlines the findings in detail, and finally Section 11.4 presents screenshots taken by the evaluators to document identified potential usability issues.

11.1 Approach taken for the Usability Inspection

11.1.1 Aim of the Usability Inspection

This Usability Inspection was done in order to reveal potential usability issues of this part of the SIMPLI-CITY App Marketplace, which is dedicated to users. (Usability Inspection of that part of the App Marketplace, which is dedicated to software developers, was done separately.) The results of this Expert Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

11.1.2 Team of Evaluators

The Usability Inspection of the road-users' part of the App Marketplace was done by the four evaluators indicated in Table 18, who did not do the Usability Inspection as a team-work, but each of them inspected the application independently.

Table 18: Evaluators of the App Marketplace

| Initials | Name | Organisation |
|----------|------------------|--------------|
| MB | Mauro Borioni | SRM |
| GL | Giuseppe Liguori | SRM |
| MK | Michaela Kargl | FGM |
| DW | Doris Wiederwald | FGM |

11.1.3 Context Information

Information related to the context in which the Usability Inspection was carried out is presented in the following Table 19.

Table 19: Context information

| | |
|------------------|---|
| Users | <p>User of this product may be anybody, who owns an Android Smartphone and wants to use SIMPLI-CITY Apps. It is assumed that the users of this product have the following relevant skills, knowledge, and experience:</p> <ul style="list-style-type: none"> • Good English language skills (but not necessarily English native speaker). • Familiar with surfing the internet using the Smartphone. • Experienced in getting Apps from Google Play Store. |
| Tasks | <p>The main tasks that users is expected to perform with this product are:</p> <ul style="list-style-type: none"> • Login to the SIMPLI-CITY Application Marketplace. • List all the Apps in the Marketplace. • List those Apps in the Marketplace, which are already installed on the Smartphone. • List those Apps in the Marketplace, which are not installed on the Smartphone, yet. • Select a non-installed App and request App installation. • Once installation confirmation is received, check that the App is installed on the device, and execute the App to check for correct installation. |
| Test Environment | <p>The Usability Inspection was done in the following technical environment:</p> <ul style="list-style-type: none"> • Smartphone LG Nexus 5 operated by the system Android 4.4.4, with browser Chrome installed, and PMA installed, under an active Wi-Fi connection. • Smartphone Samsung S3 mini GT-I8200 operated by the system Android 4.4.2, with browser Chrome installed, and PMA not installed, under an active Wi-Fi connection. • Smartphone Motorola Moto G operated by the system Android 5.0, with browser Chrome installed, and PMA not installed, under an active Wi-Fi connection. |

11.1.4 Method Applied

For the Expert Evaluation of the SIMPLI-CITY App Marketplace, the well-established Usability Inspection method “*Heuristic Evaluation*” was applied: At the core of the usability inspection method “Heuristic Evaluation” is the evaluation of a user interface to a checklist of heuristic usability design rules in order to reveal key user interface issues.

The evaluators examined the SIMPLI-CITY App Marketplace and assessed its compliance with the “*Usability Heuristics for touchscreen-based mobile devices with voice-based user interfaces*” listed below. The evaluators followed the task-based Heuristic Evaluation methodology, and examined the App Marketplace by following the user’s main tasks as listed in the context information table above. The evaluators were advised to work alone,

without communicating with the other evaluators, and delivered a list of their positive findings as well as a list of the potential usability issues they have revealed. The evaluators' individual lists were consolidated and combined into the "heuristic evaluation report", which can be found in Section 11.3 of this document.

11.1.5 List of Heuristics

The list of heuristic usability principles is at the core of the "Heuristic Evaluation" method. For the Heuristic Evaluation of the SIMPLI-CITY App, the evaluators used the following set of 12 usability principles as reference. This list of "*Usability Heuristics for touchscreen-based mobile devices with voice-based user interfaces*" is compiled based on Jakob Nielsen's "10 general principles for interaction design"⁸, the "Usability Heuristics for Touchscreen-based Mobile Devices"⁹, the "Usability Checklist for Voice-based User Interfaces"¹⁰, and Gerry Gaffney's "Voice Interaction Checklist"¹¹.

Table 20: List of Usability Heuristics

| Usability Heuristics for Touchscreen-based Mobile Devices with Voice-based User Interfaces | |
|---|---|
| 1 | <p>Visibility of system status / Suitable feedback</p> <p>The application should provide feedback to every action of the user; the application should inform the user about successful/not successful recognition of the user's input; the application should keep the user informed about all processes and state changes in a reasonable time; if the application takes a longer processing time, the user should be informed about the current state and also about the expected duration of this waiting time.</p> |
| 2 | <p>Match between the system and the real world</p> <p>The application should speak the user's language with words, phrases and concepts familiar to the user; real-world conventions should be followed, and information should be displayed in a logical and natural order.</p> <p><u>Voice-based UI:</u></p> <p>The dialogue structure should resemble human dialogue, and alternative options should be clearly delineated by clear pauses.</p> |
| 3 | <p>User control and freedom</p> <p>The application should allow the user to undo and redo her/his actions, and it should provide "emergency exits" to leave the unwanted state. The user should be allowed to walk through the dialogues in her/his own pace. The user should be able to interrupt at any time, and shouldn't be forced to pass through the extended dialogue. The application should provide a clear way to return to a starting point or main menu.</p> |

⁸ <http://www.nngroup.com/articles/ten-usability-heuristics/>

⁹ developed at the "School of Informatics Engineering of the pontifical Catholic University of Valparaiso, Chile and Polytechnic School of Chimborazo, Ecuador. Source: R. Inostroza, C. Rusu, S. Roncagliolo, C. Jinénez, V. Rusu: "Usability Heuristics for Touchscreen-based Mobile Devices", in 2012 Ninth International Conference on Information Technology – New Generations, IEEE, DOI 10.1109/ITNG.2012.134

¹⁰ Valeria Farinazzo, Martins Salvador, Andre Luiz S. Kawamoto and Joao Soares de Oliveira Neto (2010). An Empirical Approach for the Evaluation of Voice User Interfaces, User Interfaces, Rita Matrai (Ed.), ISBN: 978-953-307-084-1, InTech, Available from: <http://www.intechopen.com/books/user-interfaces/an-empirical-approach-for-the-evaluation-of-voice-user-interfaces>

¹¹ Gerry Gaffney, 2001 Information & Design Pty Ltd, "Voice Interaction checklist", www.infodesign.com.au

| Usability Heuristics for Touchscreen-based Mobile Devices with Voice-based User Interfaces | |
|---|--|
| 4 | Consistency and standards The application should follow the established conventions; the user should be able to do things in a familiar, standard and consistent way. |
| 5 | Error prevention In order to prevent errors non-available functionalities should be hidden or disabled, and the user should be able to get additional information about all available functionality. Users should be warned when errors are likely to occur. The application should provide feedback to the user, when the user's input has not been understood, and ask for more information, if the user's input has been ambiguous or inconsistent. The application should use a dialogue strategy based on confirmation. |
| 6 | Help users recognise, diagnose, and recover from errors Error messages in the application should be expressed in plain language (no abbreviations, no codes), precisely indicating the problem, and constructively suggesting a solution. |
| 7 | Customisation and shortcuts The application should provide basic configuration options and should give expert users access to advanced configuration options. The application should provide shortcuts to the most frequent tasks and should allow their customization or definition. The application should provide suitable messages that match the level of a variety of users. The application should allow step-by-step actions for novices and more complex inputs for advanced users. |
| 8 | Aesthetic and minimalist design The application should use simple and clear language, and short sentences; the information given should be concise, correct, and relevant. |
| 9 | Minimise the user's memory load The user should not have to remember information from one part of the dialogue to another. Objects and options should be visible, and instructions for use should be easily retrievable. <u>Voice-based UI:</u> The application should take into account that users' auditory memory is limited to a few short items; the menu structure should be simple with no unnecessary levels; there should not be more than five options in any menu. |
| 10 | Help and documentation Whenever the user needs it (at any stage of the dialogue), the application should provide easy-to-retrieve instructions and help, centred on the user's current task. A list of concrete steps to carry out should be provided. |
| 11 | Interaction and ergonomics <u>Touch UI:</u> User interface elements should be placed in a recognisable position and should fit the natural posture of the hand. <u>Voice-based UI:</u> The application's outputs should be clear, with natural intonation and rhythm; the application should understand natural user speech, and be tolerant of differences in accent and speech patterns; the application should recognise male, female and child voices equally well; the system should be tolerant of noise. |
| 12 | Privacy Information about how personal data is protected and about content's copyright should be given. |

11.2 Summary of the Results of the Usability Inspection

The version of the SIMPLI-CITY App Marketplace, which was delivered to the evaluators for Usability Inspection, still needed some improvements before it can be considered as usable and functioning as expected. The potential usability problems, which were revealed by the Usability Inspection, were forwarded to the respective developer partners, so that the SIMPLI-CITY App Marketplace can be further improved.

Table 22 in Section 11.3 presents a list of all the potential usability problems, which were revealed by the evaluators. The main issues were:

- The App Marketplace seems to be not designed as a "responsive website" --> the page-view does not adjust automatically to the size of the display, and thus the page is only partially displayed on the small screen of the device. (e.g., the "search button" is not visible alongside with the respective communication field where the user enters the search term, and the "install button" is not visible when reading the App description and vice versa).
- It is not possible to install the App by clicking on the "install button", and no feedback is provided to the user in case the App cannot be installed.
- The user is frequently thrown out from the marketplace and has to re-login.
- There is no information about the marketplace (not even an introductory text), no description of the functionality, no help supporting the user.

Despite the potential usability issues, the Usability Inspection revealed also some positive findings. Table 21 in Section 11.3 presents all positive findings. The main positive aspects noted by the evaluators were:

- The graphic approach and the navigation are good and easy.
- The search function is case insensitive and works also with fragments of words.

11.3 Detailed Documentation of the Findings

The following Table 21 lists all positive findings that were observed by the evaluators during the Heuristic Evaluation:

Table 21: Positive Findings

| # | Description of the Observed Positive Findings | Reported by Evaluator | | | |
|---|---|-----------------------|----|----|----|
| | | MB | GL | MK | DW |
| 1 | In general the graphic approach and the navigation are good and easy. | x | x | | |
| 2 | Login works smoothly. | | | x | |
| 3 | When clicking the "Install" or the "Search" button, the user receives visual feedback from the button that clicking was successful (- button colour changes). | | | x | |
| 4 | Search function is case insensitive and works also with fragments of words. | x | | x | |
| 5 | Download of file under the menu "Documentation" works. | | | | x |

| # | Description of the Observed Positive Findings | Reported by Evaluator | | | |
|----|---|-----------------------|----|----|----|
| | | MB | GL | MK | DW |
| 6 | Step back from App to Marketplace works but only once. (When entering another App and step back once more, the Login-Page appears.) | | | | x |
| 7 | Once found out how the search function has to be handled it works (via "search" button and not "search app"). | | | | x |
| 8 | Every app has a short description, and the average rate (in stars) of previous users is clear. | x | | | |
| 9 | Within the app mask, some very essential information is provided related to the app (e.g., category, version, published date). | x | | | |
| 10 | Within the app mask, some pictures taken by the app are provided. This is useful to let users have a better idea of the app itself. | x | | | |

The following Table 22: Potential Usability Issues includes all potential usability issues that were revealed by the evaluators. Related screenshots are provided in Section 11.4. These potential usability problems found in course of the Heuristic Evaluation are ordered according to their severity, with the most critical problems on top of the list. The severity has been assessed by the evaluators according to the following scale:

- 0 – I do not agree that this is a usability problem at all
- 1 – Cosmetic problem (is not necessary to be fixed)
- 2 – Medium severe usability problem (should be fixed)
- 3 – High level usability problem (imperative to fix this before product can be released)

Table 22: Potential Usability Issues

| # | Description of the Observed Usability Issue | Violated Heuristic | Severity | Reported by Evaluator | | | | Corresponding Screenshot(s) (to be found in Section 11.4) |
|---|--|--------------------|----------|-----------------------|----|----|----|--|
| | | | | MB | GL | MK | DW | |
| 1 | It is not possible to install the app, even pushing on the "Install" button. It seems it is not active at all, and no message is provided to inform the user about that. It's also not possible to check the installation process. | 1 | 3 | x | x | x | | #17_1 |
| 2 | App Marketplace is not designed as a "responsive website" --> the page-view does not adjust automatically to the size of the display, and thus the page is only partially displayed on the small display of the mobile phone. | 8, and 11 | 2.75 | x | x | x | x | #01, #03 |

| # | Description of the Observed Usability Issue | Violated Heuristic | Severity | Reported by Evaluator | | | | Corresponding Screenshot(s) (to be found in Section 11.4) |
|----|--|--------------------|----------|-----------------------|----|----|----|--|
| | | | | MB | GL | MK | DW | |
| 3 | Sometime it happens that the user is "thrown out" by the App Marketplace, and has to re-login. Not clear when connection is timed-out (i.e. when you have to re-enter the password), jumps back to Login-page very quickly - after a minute or so (without explanation). | 1 | 2.75 | x | | x | x | n.a. |
| 5 | When the user push the button "back" to move from a page to the main menu it seems that a logout process happens, since if the user wants to visit another app page the login is requested. | 3 | 2.75 | | x | | x | n.a. |
| 6 | When entering text in the communication field and then pressing "Search App" - Login Page appears. What is the "Search App"-Button for? (as the communication field is there anyway). | 3, 4 and 5 | 2.5 | | | | x | #04 |
| 7 | "Search" button not visible when entering text in the communication field. | 4, and 11 | 2.25 | x | | x | x | #03, #04 |
| 8 | The install button is too far on the right side and thus not visible when reading the app description. | 11 | 2.25 | | | x | | #16 |
| 9 | For the App "Environmental Awareness Rising" a button-like "Installed" label is displayed. The user would expect to get the possibility to uninstall the App, when clicking on the button "Installed". However, this is no clickable button, it just looks like a button; Furthermore, also it is marked as "installed" the App "Environmental Awareness Rising" cannot be found among the Apps installed on the mobile phone. | 2, 4 | 2.25 | | | x | | #18 |
| 10 | User cannot enlarge or reduce the screen, the zoom is not possible. | 11 | 2 | x | | | | n.a. |
| 11 | There is no information about the marketplace, and no description of the functionality, no help. | 5, and 10 | 2 | | | x | x | #01 |
| 12 | No support at Login Page in case password has been forgotten. | 3 | 2 | | | | x | #25 |
| 13 | The compatibility with OS and devices is not foreseen. | 10 | 2 | | x | | | n.a. |
| 14 | Button "Search App" and "Search" redundant? | 8 | 1.75 | | | | x | #04 |

| # | Description of the Observed Usability Issue | Violated Heuristic | Severity | Reported by Evaluator | | | | Corresponding Screenshot(s) (to be found in Section 11.4) |
|----|--|--------------------|----------|-----------------------|----|----|----|--|
| | | | | MB | GL | MK | DW | |
| 15 | No explanations under the respective apps on how to download them, integrate comments, etc. | 5 | 1.75 | | | | x | #22 |
| 16 | Within the app mask, it is not provided the information about the size of the app | 10 | 1.75 | x | | | | #15 |
| 17 | The developer(s) name, contacts and reputation are not displayed. | 10 | 1.75 | x | x | | | #22 |
| 18 | A link to SIMPLI-CITY website is missing. | 10 | 1.75 | x | x | | | #01 |
| 19 | No first-time-registration possible at Login-Page. | 3 | 1.75 | | | | x | #25 |
| 20 | It's not clear how it will be possible to rate the apps. | 11 | 1.75 | | x | | | n.a. |
| 21 | No explanation what the Documentation Link contains. | 5 | 1.5 | | | | x | #06 |
| 22 | Why is there a link to the "Publication and Review Guidelines" at the site dedicated to consumers/road-users? - that's confusing. | 8 | 1.5 | | | x | | #07 |
| 23 | Unclear/not explained what the communication field is for. Entering text there is possible, no search function, nothing happens. | 5 | 1.5 | | | | x | #01, #03 |
| 24 | There is no introductory text on the App Marketplace. | 5 | 1.25 | | | | x | #01 |
| 25 | Design is not consistent - it is not clear, which elements are clickable "buttons", which elements are links, and which elements are expandable menu headings. | 4 | 1,25 | | | x | | #04 |
| 26 | The area allocated to include the logo of the apps is too big, considering the size of the logos. | 11 | 1,25 | x | | | | #01 |
| 27 | There's no explanation what the stars dedicated to the apps mean. | 5 | 1 | | | | x | #04, #05 |
| 28 | Rather a remark: it is possible to access the App Marketplace without entering a password. | n.a. | 0.75 | | | | x | n.a. |

11.4 Screenshots Documenting Potential Usability Issues

This section displays a digest of the screenshots taken by the evaluators during their assessment in order to document identified potential usability issues:

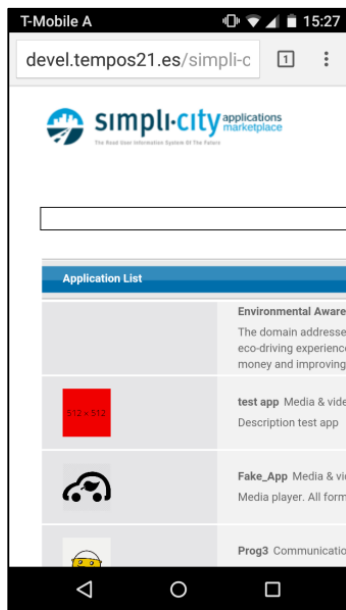


Figure 11: Screenshot #01



Figure 12: Screenshot #25

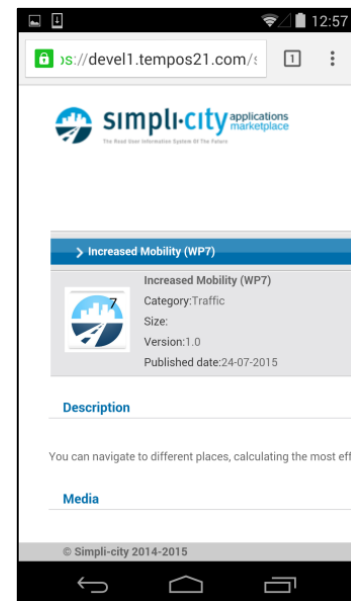


Figure 13: Screenshot #22

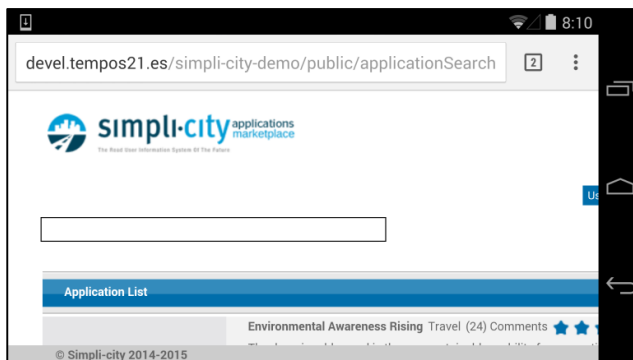


Figure 14: Screenshot #03

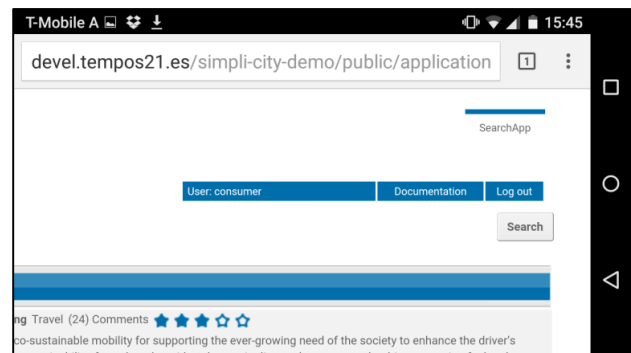


Figure 15: Screenshot #04

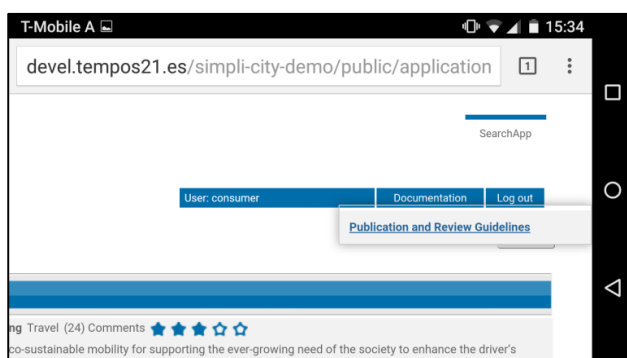


Figure 16: Screenshot #07

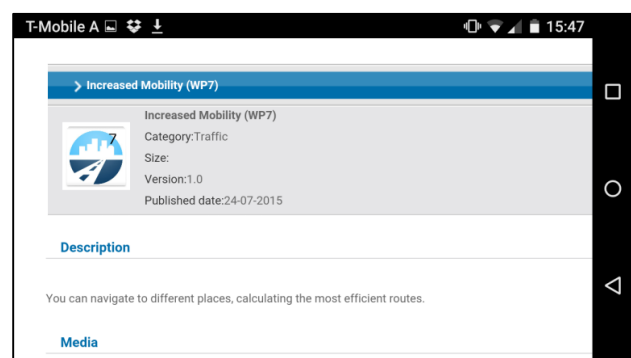


Figure 17: Screenshot #16

12 Usability Evaluation of the Road Users' Part of the App Marketplace

This chapter presents the results of the Usability Evaluation of the SIMPLI-CITY Application Marketplace from road-users' point of view.

In Section 12.1 the Usability Evaluation approach is described briefly. Section 12.2 gives a summary of the results of the Usability Evaluation, and Section 12.3 outlines the findings in detail. Furthermore, Section 12.4 includes the Task Sheets, which were provided to the test users during the evaluation sessions, Section 12.5 provides a transcription of the Debriefing Interviews conducted with the test users at the end of the sessions, and Section 12.6 includes the notes taken by the observer during the usability evaluation sessions with the test users.

12.1 Approach taken for the Usability Evaluation

12.1.1 Aim of the Usability Evaluation

This Usability Evaluation was done in order to find out how easy it is to use this part of the SIMPLI-CITY Application Marketplace that is targeted at road-users. The results of this Usability Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

12.1.2 Method Applied

For the Usability Evaluation of the SIMPLI-CITY Application Marketplace from road-users' point of view, the well-established Usability Evaluation method "*Thinking Aloud Test*" was applied:

A test-user is asked to speak out loud all her/his thoughts while using the system to complete a given test-task. By observing the test-user's way of working on the task, and by analysing the reasons for confusion, hesitation, or mistakes of the test-user, valuable hints for improvement of the usability of the system can be obtained.

12.1.2.1 Team of Facilitators

Three persons formed the team, which organised and conducted the usability evaluation sessions of the road-users' part of the SIMPLI-CITY Application Marketplace.

Table 23: Team of Facilitators of the Usability Evaluation Sessions

| Name (Organisation) | Role within the Facilitators' Team |
|-----------------------|--|
| Michaela Kargl (FGM) | "Moderator" (guided the test-users through the session) |
| Markus Schuster (FGM) | "Observer" (took notes of revealed usability issues) |
| Andreas Maurer (FGM) | "Technician" (ensured proper functioning of the technical equipment, software, and recordings) |

12.1.2.2 Technical Setup for the Usability Evaluation

The test-user worked on a Laptop running Windows 7 as operating system. On this computer the browsers Mozilla Firefox, Google Chrome and Internet Explorer were installed, which the test-users utilised for accessing the SIMPLI-CITY Application Marketplace when working on the test-tasks.

TeamViewer 10 was used for displaying the test-computer's screen on the observer's computer, and CamStudio v2.7.4 was used for screen recording. A Logitech 920 HD Pro webcam and the related software were utilised for recording the test users' facial expressions and voice during their work on the test-tasks, and a Panasonic Lumix GX7 camera was used for video and audio recording of the debriefing/feedback interviews.

12.1.2.3 Usability Evaluation Sessions

The Usability Evaluation Sessions took place at FGM's premises, Schönaugasse 8a, 8010 Graz, Austria. The pre-test was done on Friday, 04.09.2015, and 5 usability sessions were conducted on Wednesday, 09.09.2015. The average duration of the Usability Evaluation Sessions was 42 minutes, with the longest session lasting for 51 minutes and the shortest session taking 35 minutes.

All Usability Evaluation Sessions followed the same process:

First the moderator welcomed the test-user, introduced the team, and explained the purpose of the session. The moderator gave an overview of the test procedure, and asked the test-user for her/his permission that the session can be recorded (video, audio and screen recording), and these records can be used for later analysis and reporting purposes. After the test-user signed the consent form, the moderator encouraged the test-user to ask questions, whenever anything is unclear. In a short introductory interview, the moderator obtained from the test-user some demographic information (gender and age) and some information regarding relevant skills and knowledge of the test-user, such as e.g., smartphone usage, experience with app stores and internet browsing in general, knowledge of SIMPLI-CITY, and expertise with Thinking Aloud tests.

After this introductory part, the moderator explained to the test-user, how the "Thinking Aloud" method works, and asked the test-user to do one test-task after the other and answer the Single Ease Question (SEQ) immediately after completion of each task. While the test-user was working on the task, she/he spoke out loud all her/his thoughts and considerations, the team of facilitators watched the test-user silently, and the observer took notes of the test-user's activities and any usability issues that became apparent.

After finishing the last test-task, the moderator thanked the test-user for her/his cooperation and asked the following three questions in the debriefing interview: "How was it?" "Was there anything that you found to be especially good?" "Was there anything that you found to be especially bad?" When the test-user finished answering these questions, the moderator asked her/him to fill-in the System Usability Scale (SUS), a standardised questionnaire to assess a user's perception of the overall usability of a system. In case the test-user had further questions these were answered by the team, and after that the session was finished.

After each session, the recordings were saved, and the materials and setup was prepared for the next session.

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 111 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

12.1.2.4 Test-Users

In total 5 test-users participated in the Usability Evaluation of the SIMPLI-CITY “App Marketplace”. All of them answered the introductory questionnaire, did the 6 tasks, and provided their feedback in the debriefing interview and feedback questionnaire.

The 5 test-users were persons with different experiences in using App stores. We’ve had four men and one woman. The test-users were between 25 and 54 years old.

Table 14 gives an overview of the relevant skills and knowledge of the test-users as stated by them in the introduction questionnaire.

Table 24: Overview of the Test-Users’ relevant Skills and Knowledge

| Test-User | User #1 | User #2 | User #3 | User #4 | User #5 |
|---|-----------------------------|-----------------------------|-----------|-------------|-----------------------------|
| Usability Eval. Session | Session 1 | Session 2 | Session 3 | Session 4 | Session 5 |
| Peculiarities | glasses / contact lenses | glasses / contact lenses | x | x | glasses / contact lenses |
| Mobile Phone Ownership | Android | iOS | Android | Android | iOS |
| Smartphone Usage since | >1 year | >1 year | >1 year | >1 year | >1 year |
| Usage of Apps | low/ seldom | high/ a lot | medium | high/ a lot | high/ a lot |
| Experience with App Stores | low | medium | medium | high | high |
| Experience with browsing the internet | high | medium | high | high | high |
| what do you know about SIMPLI-CITY | heard of | heard of | heard of | worked for | heard of |
| Ever participated in a Usability Study | yes | yes | no | no | No |

Three of the test-users had an Android smartphone, two of them had a smartphone operating with iOS. All of the test-user had used a smartphone for longer than a year. Three of the test-users stated that their usage of Apps is “high”, one of them said to have “medium” usage and one said that he uses Apps “seldom”. Four of the test-users stated that their experience in browsing in the internet is “high”, one test-user self-assessed that his experience is “medium”.

None of the test-users was involved in the development of the SIMPLI-CITY “App Marketplace”. One of the test users was working for other parts of SIMPLI-CITY, three of the test-users had heard of SIMPLI-CITY before, and one test-user did not know anything about SIMPLI-CITY.

Two of the test-users had participated in a usability study before, but none of them in a Thinking Aloud Test.

12.1.2.5 Evaluation Tasks

The test-users were asked to complete the following tasks:

- Task 1: Login to <https://devel1.tempos21.com/simpli-city-demo/> as a user.
- Task 2: When you've logged in to <https://devel1.tempos21.com/simpli-city-demo/>, find out what this website is about.
- Task 3: Search for Apps that support you with navigation, and help you to find the best route for driving to any destination. (Hint: The site offers a search function)
- Task 4: Find out more detailed information for any of the Apps you are interested in.
- Task 5: Install any of the Apps on your Smartphone.
- Task 6: Log out from the SIMPLI-CITY App Marketplace.

Section 12.4 includes the task-sheets, which the test-users got during the usability evaluation session.

12.2 Summary of the Results of the Usability Evaluation

The Usability Evaluation sessions with the test-users were quite successful, since a number of important usability issues were discovered during the test sessions.

In addition to important qualitative observations, also some quantitative metrics were obtained, such as the test-user's time on task and the completion rate for each of the test-tasks. However, due to different levels of experiences of the users, high variations of the users' task completion times in the test-tasks 2, 3, 4 and 5 were noticed.

Most test-users found the test-task 2 ("Find out what this website is about"), test-task 3 ("Search for an app that helps you to find the best route") and test-task 5 ("Install any of the apps listed in the SIMPLI-CITY marketplace on the smartphone") to be quite difficult:

During the debriefing interview at the end of the Usability Evaluation sessions, the test-users stressed three features of the SIMPLI-CITY App marketplace, which they've especially liked:

- The website is very clear and simple.
- The site is not overloaded.
- The design of the login page and the logo of the App marketplace are appealing.

However, the test-users mentioned also some aspects of the SIMPLI-CITY Application Marketplace, which should be improved:

- There should be some introductory text at the top of the page describing what this website is about (some sort of "Welcome text", which explains to the user the features of the marketplace)
- The search function should be better (not only full text search of the app descriptions, but also finding suitable apps when typing in more general keywords such as e.g. navigation or routing.
- There should be used graphics and not only text in the list of the Apps.
- There should be more information about costs and size of the Apps.
- The website should be optimized for smartphones.

After the Usability Evaluation, the results of the evaluation sessions were communicated to the developers of the SIMPLI-CITY Application Marketplace.

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 113 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

12.3 Detailed Documentation of the Findings

12.3.1 Usability Metrics

12.3.1.1 Task Completion Success Rate

Overall performance of each participant was evaluated depending on his/her ability to finish the task on his/her own. The task success rate score is the number of participants that finished the task successfully divided by the total number of participants.

Table 25: Task Completion Success Rate

| Participant | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---------------------|-------------|------------|------------|------------|-----------|-------------|
| User #1 | ✓ | ✓ | - | ✓ | - | ✓ |
| User #2 | ✓ | - | - | - | - | ✓ |
| User #3 | ✓ | ✓ | ✓ | ✓ | - | ✓ |
| User #4 | ✓ | ✓ | ✓ | ✓ | - | ✓ |
| User #5 | ✓ | ✓ | ✓ | ✓ | - | ✓ |
| Success Rate | 100% | 80% | 60% | 80% | 0% | 100% |

✓ = successfully completed;

- = not completed / help needed for completion

12.3.1.2 Ease-of-Task Ratings

After each task, the participants rated the ease of completing the task, by answering the “Single Ease Question” (SEQ)¹² “*Overall this task was?*” on a 7-point rating scale ranging from “Very Difficult”(1) to “Very Easy”(7).

Table 10 shows the ratings given to the single tasks by the test-users, and provides an average of these ratings for each task.

Table 26: Ease-of-Task Ratings

| Participant | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---------------------------------------|-------------|------------|------------|------------|------------|-------------|
| User #1 | 7 | 2 | 2 | 6 | 2 | 7 |
| User #2 | 6 | 3 | 4 | 1 | 1 | 7 |
| User #3 | 7 | 5 | 4 | 6 | 5 | 7 |
| User #4 | 7 | 6 | 2 | 7 | 1 | 7 |
| User #5 | 7 | 5 | 5 | 7 | 4 | 7 |
| Average Rating | 5.1 | 3.1 | 2.9 | 3.5 | 1.8 | 5.3 |
| Percent “Easy” (Rating 6 or 7) | 100% | 17% | 17% | 67% | 0% | 100% |

¹² Jeff Sauro: “10 Things to know about the Single Ease Question (SEQ)”, online at <https://www.measuringu.com/blog/seq10.php>, visited 12.06.2015

12.3.1.3 Time on Task (TOT)

The variations in time needed by each participant to finish their tasks were due to their knowledge about App stores and their experience in usage of smartphones.

Table 27: Participants' Time-On-Task

| Participant | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---------------------------------|------------|------------|----------|------------|----------|------------|
| User #1 | 3 | 6 | 4 | 2 | 7 | 3 |
| User #2 | 3 | 3 | 2 | 4 | 5 | 1 |
| User #3 | 3 | 4 | 5 | 3 | 3 | 2 |
| User #4 | 2 | 4 | 2 | 2 | 5 | 1 |
| User #5 | 2 | 2 | 2 | 3 | 5 | 2 |
| Average TOT (in minutes) | 2.6 | 3.8 | 3 | 2.8 | 5 | 1.8 |

12.3.1.4 Rating of the Overall System-Usability

After the completion of all tasks, the participants were asked to fill in a standardised feedback questionnaire (SUS System Usability Scale).

As explained in Section 7.3.1.4, SUS yields a single number representing a composite measure of the overall usability of the system. SUS scores have a range of 0 ("not usable at all") to 100 ("perfectly usable system").

Analysis of the answers of the test-users to the feedback questionnaire reveals that the road-users' part of the SIMPLI-CITY App Marketplace has got a SUS score of 61.7. According to Bangor A. et al¹³, this SUS score can be translated to users' perceiving that the usability of the system is "acceptable / o.k." This indicates that there's still some substantial improvement of the system's usability necessary.

12.3.2 Main Positive Aspects Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users stressed two features of the SIMPLI-CITY App Marketplace, which they've liked:

1. the log-in page is good

User #2: "Ya, at the beginning it was in principle fine – the log in was fine. Also the log out was fine"

User #5: "So, first of all, the SIMPLI-CITY logo is great, and also the log-in page is o.k. So the first catch is o.k."

2. it is good to find the list of apps at the first page, where you enter the market place

User #3: "It's very simple to find the apps, because you just see them immediately after log-in."

¹³ Adjective ratings related to SUS scores: 25 = "worst imaginable", 38 = "poor", 52 = "acceptable / o.k.", 74 = "good", 85 = "Excellent", 100 = "best imaginable" (Source: Bangor A., Kortum P., Miller J.: "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale", in Journal of Usability Studies JUS, Vol. 4, Issue 3, May 2009, pp. 114-123)

User #1: *“The overview of the app list is actually good, you see it directly when you start on the front page when you have entered in. You have entries that tell you if there is a discussion on an app, or an app itself offered. This is quite fine.”*

12.3.3 Main Usability Issues Revealed During the Evaluation

12.3.3.1 Main Usability Issues Mentioned by the Test-Users

During the debriefing interview at the end of the Usability Evaluation Sessions, the test-users mentioned several aspects of the SIMPLI-CITY system, which should be improved. The main issues were:

1. Introductory text is missing

User #2: *“When you arrive on the website, where you have the selection of apps, there should be a brief explanation, where I am. That would be the most important thing, saying: “This is SIMPLI-CITY and you can download various apps here...” A brief explanation, that would be the most important thing.”*

User #3: *“I did not know what to expect when I typed in the link. That’s why I was confused. I was not sure whether this is the first page. I would expect a bit of explanation on the first page, what is the website about or something. There was just a list of apps. Of course, if you actually want to search for an app it’s quite easy, because you see the list at once and do not need a lot of time. But maybe a sentence or two at the beginning would be useful for people, who have not visited the web page before.”*

User #5: *“Maybe something on top about this app store or app marketplace, whatever, some background information.”*

2. App Marketplace should work on a Smartphone

User #2: *“There was one important thing: on the Android on the screen of the mobile it was... the install-button was not visible. You had to scroll, so that shouldn’t be the case.”*

User #3: *“Really bad is that it is not adjusted for smartphone. Because, if you download apps for smartphones, also the webpage should be adjusted for smartphone. I was irritated with sweeping left and right. This was very bad.”*

3. Layout of the “Search” function is confusing

User #5: *“I need a better search function. A search field or so with written “search” inside.”*

User #4: *“From the design you can see that it is not far in the development, yet. So the search field is separated from the button, and so on.”*

4. “Search” should work with the “enter-key”

User #1: *“The search option should react on the enter-button. I guess I’m not the only one who is typing and then not using the mouse to click on anything but just firing at the return key.”*

5. Search algorithm should be improved

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 116 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

User #4: *“Well, bad was the search. You do not find anything. I just typed in two words from the task description. And “navigation” or “routing” should find a navigation app, and it did not, so that was bad. they have to add more search words, I guess. So, not only the description, but maybe some extra words, ...”*

6. Installation procedure is confusing

User #2: *“This response case is confusing. When this little screen pops up and says it is installing – I do not remember exactly what was written. For me it was completely confusing – “What is this?” It should be a bit clearer in the wording, or not pop up at all and simply start installing. Yes.”*

User #3: *“I was not sure if it is installed or not on the phone, this is confusing. I was not sure if I should do anything else after pressing the install button. This is confusing. That’s all I could say.”*

12.3.3.2 Main Usability Issues Noted by the Observer

In the following, the main usability issues, as noticed by the observer during the usability evaluation sessions with the test users are listed:

Task 1: “Login to <https://devel1.tempos21.com/simpli-city-demo/> as a user.”

- no usability issues encountered: login task was straight forward for all test-users

Task 2: “When you’ve logged in to <https://devel1.tempos21.com/simpli-city-demo/>, find out what this website is about”

- all test-users had difficulties to find out, what this website is about: they were reading the texts carefully, but could not find a hint about the purpose and features of the website, since the only texts available were (dummy) app descriptions (see screenshot Figure 18)
- only one of the test-users had the idea to use the “help” link for this task, but then was a bit irritated at first, since the help link opened a word-document and the test user stated that this is rather unusual

Task 3: “Search for Apps that support you with navigation, and help you to find the best route for driving to any destination. (Hint: The site offers a search function)”

- all test-users typed into the search-field “navigation”, “routing”, “route”... - but only with the word “routes” the respective app is found, as the search function finds only words that are included in the description of the app
- all test-users pressed the enter-key when they had finished the entry into the search field; all test-users expected the search to be triggered by pressing the enter-key; after some hesitation they finally found the search-button and clicked on it
- the test-users were confused, when the search function had not found a corresponding app, because there was only an empty screen displayed, and there was no feedback from the system (see screenshot Figure 19); Thus the test-users did not know, whether the search is still working (and they need to wait a bit longer) or whether there is no such app available in the marketplace...

Task 4: “Find out more detailed information for any of the Apps you are interested in.”

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 117 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

- all test-users clicked on the short app description in order to find out more details, and as expected they were transferred to the page with the “more detailed app description”; However, since there was not additional information available on this page, the test-users were disappointed. (see screenshot Figure 20)
- The test-users stated that they would have expected to find a more detailed description of the apps functionality, screenshots of the app, price information, etc.

Task 5: “Install any of the Apps on your Smartphone.”

- all test-users first tried to find the app on the Google Play Store, only after a hint from the moderator they tried to install the app directly from the SIMPLI-CITY App Marketplace
- 4 out of 5 test-users immediately after reading the task description took the smartphone and tried to do this task on the smartphone, since they know from their experience that installing apps on a smartphone is done by accessing an app store with the respective smartphone...; only one of the test-users knew, that it is also possible to install apps on the smartphone by using the PC/laptop
- all test-users found the “install-button” easily. However, after pressing the button they were puzzled, as a pop-up window opened telling “App Installation requested” and they had to press “o.k.” – they were completely confused and not sure what to do now: they did not know, whether the app is now installed on the smartphone, or whether they need to do any further steps... (see screenshot Figure 21)

Task 6: “Log out from the SIMPLI-CITY App Marketplace.”

- no usability issues encountered: logout task was straight forward for all test-users

12.3.3.3 Screenshots Documenting Usability Issues

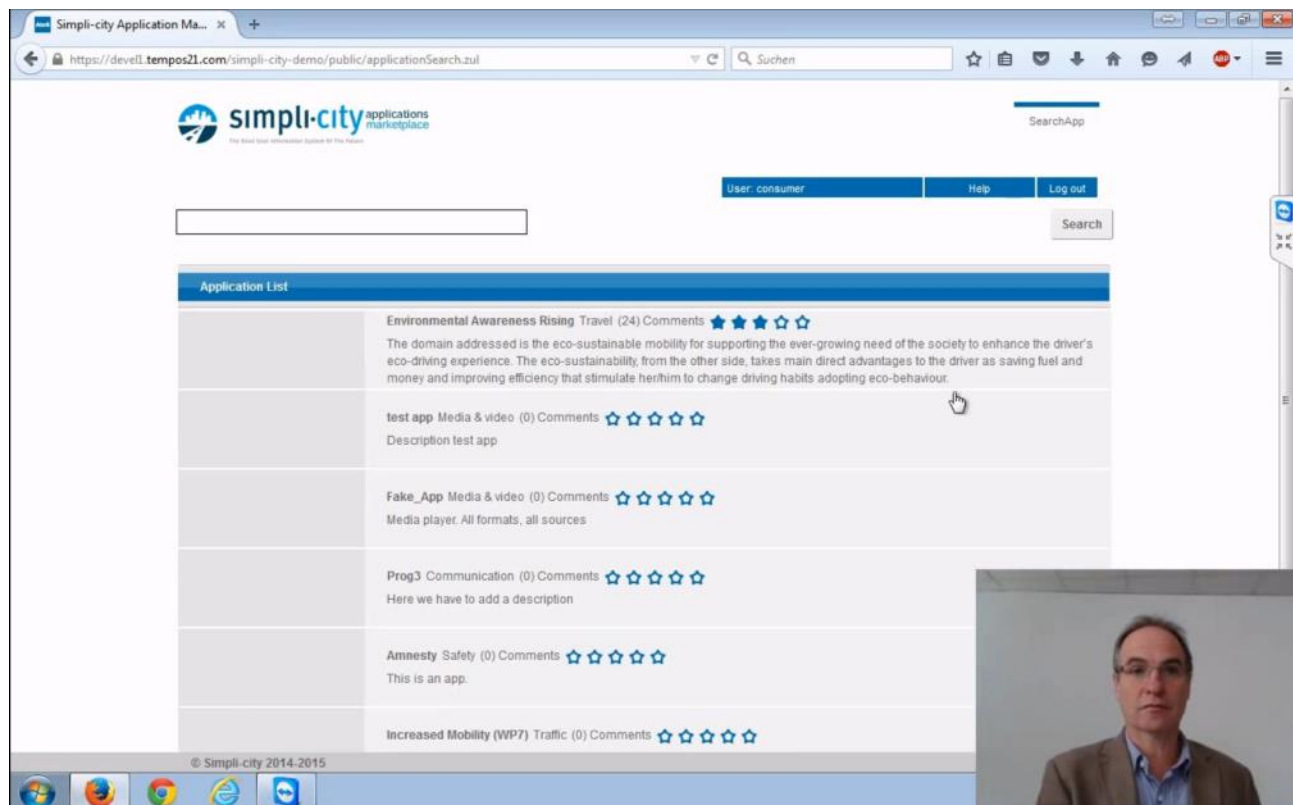


Figure 18: No introductory text on the App Marketplace

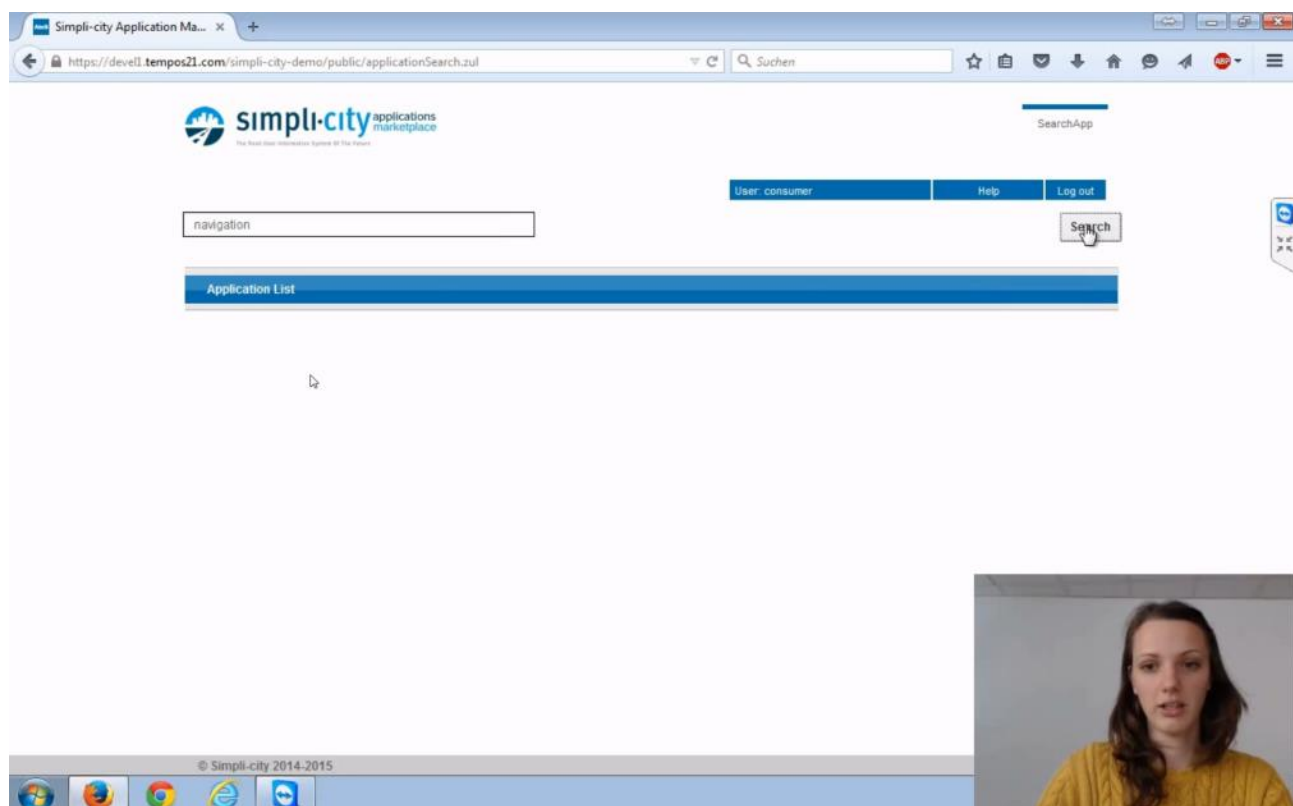


Figure 19 Search result for "navigation" – Missing system feedback

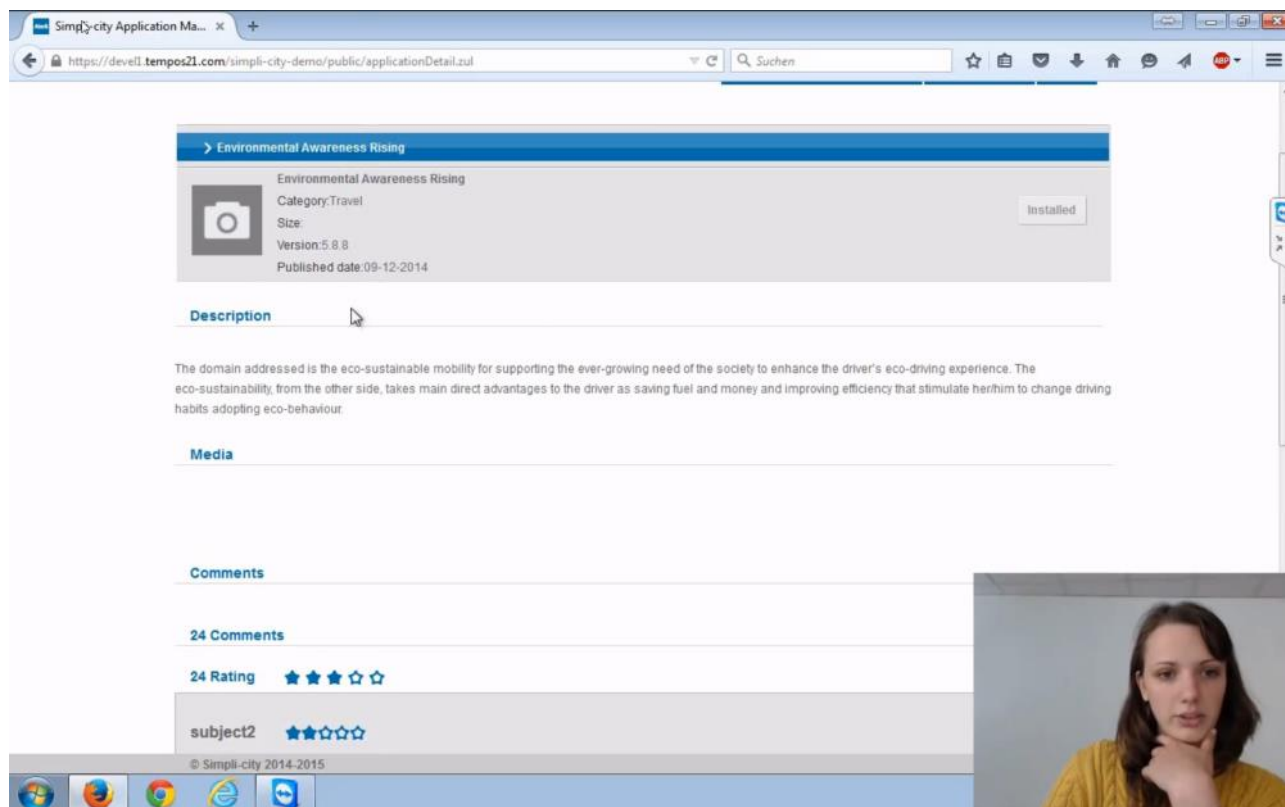


Figure 20: Description of the App “Environmental Awareness Rising” – poor information

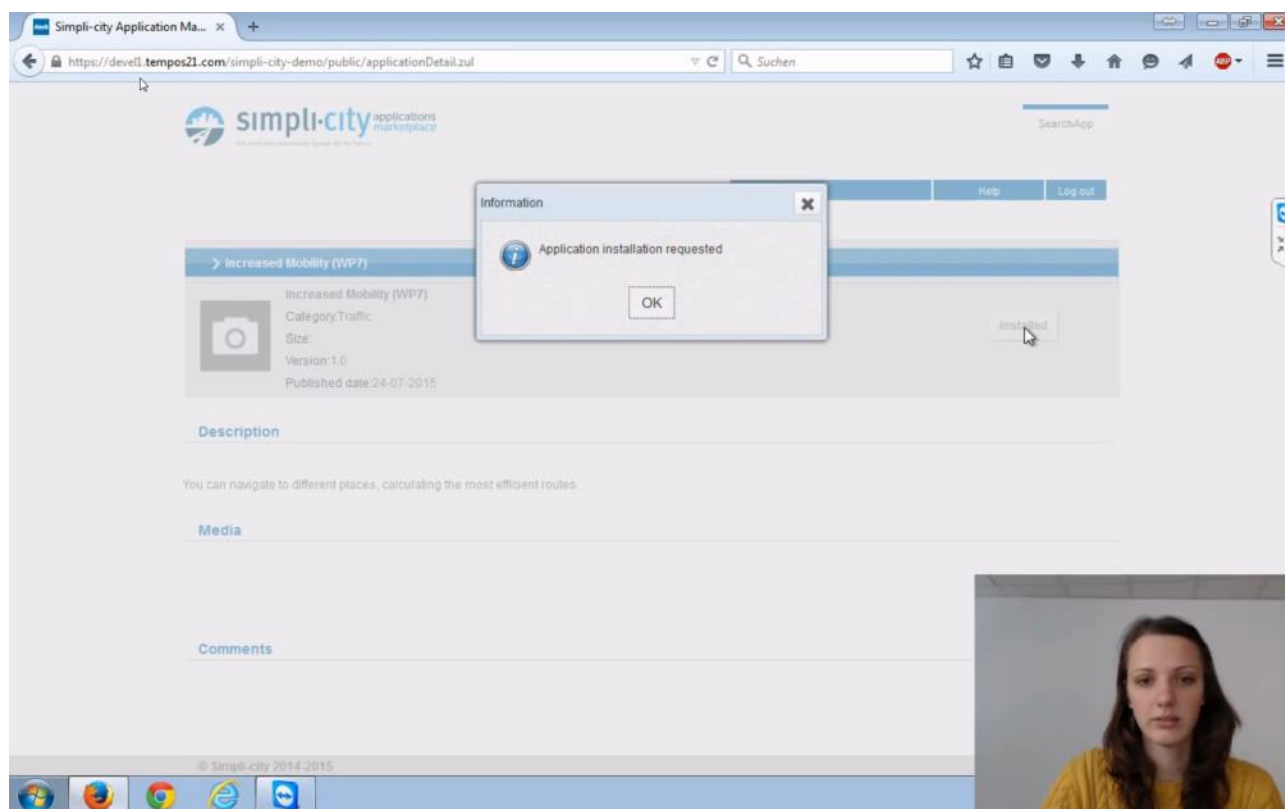


Figure 21 Pop-up window appearing after clicking the install button

12.4 Task Sheets for Usability Evaluation of the App Marketplace from road-users' point of view

TASK 1:

Login to <https://devel1.tempos21.com/simpli-city-demo/> as a user.

Hints:

Your username is “consumer”

Your password is also “consumer”.

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐



TASK 2:

When you've logged in to <https://devel1.tempos21.com/simpli-city-demo/>, find out what this website is about.

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐



TASK 3:

Search for Apps that support you with navigation, and help you to find the best route for driving to any destination.

(Hint: The site offers a search function)

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

TASK 4:**Find out more detailed information for any of the Apps you are interested in.**

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐**TASK 5:****Install any of the Apps on your Smartphone.**

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐**TASK 6:****Log out from the SIMPLI-CITY App Marketplace.**

Tell the moderator, when you have finished this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

12.5 Transcription of the Debriefing Interviews

In the debriefing interview the test-users explained what they liked at the SIMPLI-CITY App Marketplace and what they perceive to be the main issues that should be improved.



Figure 22: Test-Users answering the Moderator's Questions during Debriefing Interview

User #1 (Evaluation Session 1)

Moderator: "How was it?"

User #1: "Interesting. It seems to be, that it could be a good marketplace to see if this are apps for direct use or guidance for development of apps. At this stage I'm not able to tell if this is going to be a good user experience, since it's not developed that far."

Moderator: "Was there anything that you find especially good?"

User #1: "The overview of the app list is actually good, you see it directly when you start on the front page when you have entered in. You have entries that tell you if there is a discussion on an app, or an app itself offered. This is quite fine. That's it."

Moderator: "Something especially bad?"

User #1: "Hmm, it's not especially bad, but the search option should react on the enter-button. I guess I'm not the only one who is typing and then not using the mouse to click on anything but just firing at the return key. And, hmm, from my experience now it's of course disappointing not to have anything like an app working, which you can install or where you can have a look on. On this stage it is poor information at a very low level, and would not motivate me to go back on the page if I wouldn't know that it's a test, and it's going to be further developed."

Moderator: "Anything else you want to tell us?"

User #1: "Maybe the developers should think about where to place the apps, or also the app marketplace, maybe it's possible to place it in an existing AppStore (e.g., Google) or something like that."

User #2 (Evaluation Session 2)

Moderator: "How was it?"

User #2: "It was a bit frustrating. At the beginning it started nice, but then you could not get any further."

Moderator: "Was there anything that was good?"

User #2: "Ya, at the beginning it was in principle fine – the log in was fine. Also the log out was fine, but in the middle it did not work. So, that was good: the beginning and the end."

Moderator: *“Was there something especially bad. Or lets formulate it in another way: What would you improve, if you were the developer of this stuff?”*

User #2: *“When you arrive on the website, where you have the selection of apps, there should be a brief explanation, where I am. That would be the most important thing, saying: “This is SIMPLI-CITY and you can download various apps here...” A brief explanation. That would be the most important thing.”*

Moderator: *“And the second important?”*

User #2: *“Second important is that for each app you have a very brief, just one line so that you can easily see what it is about. Then of course by clicking on the app you get the description of the app. You should get a more detailed description of the app not just the same text as in the overview.”*

Moderator: *“Anything else that you would like to tell the developers?”*

User #2: *“Well I suspect that this is sort of a test of the surface of the whole thing, so I think that apps will come in the future, right? So I won’t say much about it. Of course, there was one thing, on the Android on the screen of the mobile it was... the install-button was not visible. You had to scroll, so that shouldn’t be the case.”*

Moderator: *“Anything else?”*

User #2: *“Yes, this response case is confusing. So it confused me, when it said that... When this little screen pops up and says it is installing – I do not remember exactly what was written. For me it was completely confusing – “What is this?” It should be a bit clearer in the wording, or not pop up at all and simply start installing. Yes.”*

User #3 (Evaluation Session 3)

Moderator: *“How was it?”*

User #3: *“It was interesting. I liked it. I mean I did not do something like this before, but it was o.k., I enjoyed it.”*

Moderator: *“Was there anything of the system that you especially liked or that you find it was implemented quite good?”*

User #3: *“It’s very simple to find the apps, because you just see them immediately after log-in. But I did not know what to expect when I typed in the link. That’s why I was a bit confused. I was not sure whether this is the first page. I would expect a bit of explanation on the first page, what is the website about or something. There was just a list of apps, but if you actually want to search for an app it’s quite easy, because you see the list at once and do not need a lot of time. But maybe a sentence or two at the beginning would be useful for people, who have not visited the web page before.”*

Moderator: *“Was there anything that you found really bad?”*

User #3: *“Really bad is that it is not adjusted for smartphone. Because, if you download apps for smartphones, also the webpage just be adjusted for smartphone. I was irritated with sweeping left and right. This was very bad.”*

Moderator: *“Is there anything else that you would improve, if you would be the developer of this stuff?”*

User #3: *“Well, it’s white and a little bit blue. Maybe it could be a bit more colourful. I know that has not influence on how it works, but it could also look better.”*

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 124 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

Moderator: *“Anything else?”*

User #3: *“Well, I know that this is not done, yet. But with installation: I was not sure if it is installed or not on the phone, this is confusing. I was not sure if I should do anything else after pressing the install button. This is confusing. That’s all I could say.”*

User #4 (Evaluation Session 4)

Moderator: *“How was it?”*

User #4: *“Well, you said at the beginning that it is just a pre-alpha state. So it’s o.k. You can log in you can find some apps, but they have to add more search words, I guess. So, not only the description, but maybe some extra words, whatever. Well, and if this works with the install-button, it’s good. From the design you can see that it is not far in the development, yet. So the search is separated from the button and so on. But I think it’s on a good way.”*

Moderator: *“Was there anything that you find especially good?”*

User #4: *“Not really. It’s not bad, but there’s also nothing especially good.”*

Moderator: *“Was there anything that you find especially bad?”*

User #4: *“Well, bad was the search. You do not find anything. I just typed in two words from the task description. And “navigation” or “routing” should find a navigation app, and it did not, so that was bad.”*

Moderator: *“Anything else? So, if you would be the developer of this stuff, what would you improve first?”*

User #4: *“I would improve the design. It is not... It does not look very good and appealing. But at the current stage, it’s just a technical proof I think so they do not need a fancy stylish website. It works, that’s o.k. And I think I would improve the details-page. Just add a couple of screenshots of the app, describe just what this app does in a couple of sentences. But, as I said before, I think it’s just the beginning of this marketplace, so everything can follow.”*

User #5 (Evaluation Session 5)

Moderator: *“How was it?”*

User #5: *“I expected a more mature version. So I do not understand why you already test it at this stage. Well, I can understand, I can assume, why you test it right now. But I would improve it, improve it, just improve it, improve it.”*

Moderator: *“Well, that’s exactly the purpose of this usability test, that you find out at an early stage of development where the flaws and problems are, and the way of thinking of the users, so that the developers can adjust to that and improve it accordingly.”*

User #5: *“Yea, but you could have made it better already now. I think there are so many basics not done right now, that the test just shows you the basics, and everything more detailed would be lost.”*

Moderator: *“Yes, that’s true. You have to make more usability tests in the course of the development. That’s completely true. So this is the first usability evaluation of this product – developing product, and of course you need to have much more of these tests in the course of the development.”*

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 125 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

User #5: *“o.k. Who is responsible for the upkeep? So, who is the operator? I’m just thinking, cause app stores are already existing, so why to invent a new one?”*

Moderator: *“Well, the App Marketplace is just one component of the SIMPLI-CITY system...”* [moderator explains the prototypes developed within SIMPLI-CITY to the test-user]

User #5: *“o.k.”*

Moderator: *“Was there anything that you find good?”*

User #5: *“So, first of all, SIMPLI-CITY logo. So the first catch is o.k. But the rest was just rather unimpressive right now.”*

Moderator: *“Was there anything that you find especially bad?”*

User #5: *“Well, I think everything is based on the user interface. So, it should be based on graphics and things like this, and it’s just using text right now, it’s not ... Well, how to say it. You can’t say anything right now, it’s just some text, it’s like: What?...You need more to say something what is good or bad. – You know, graphics...”*

Moderator: *“Anything else? What would you improve?”*

User #5: *“I need a better search function. A search field or so with written “search” inside. Maybe something on top about this app store or app marketplace whatever, some background information. I would like to have categories. So I do not know what’s coming out, but there was something like route navigation and something with eco driving. Or funny games for eco education – something like this. So I just would look at every other app store and take the best out of it. I would also add more information, more information about what I need to use it. If I have to pay for it. If there are in-app purchases, because there are other products for route navigation and other modes of transport where you have to pay something after downloading the app and I do not like this at all, but... That’s it, I think.”*

12.6 Observer Protocol

Test User 1, 09.09.2015

Timeline:

- Introduction 10:10 - 10:17; Explaining example task 10:17 - 10:21, User Test 10:17 - 10:47, Debriefing Interview: 10:48 – 10:52

Task 1:

- 10:21 Test user starts with task 1
- Types in the URL, comes to the login page
- 10:23 task was completed, the test user says it was very easy

Task 2:

- 10:24 Test user starts with task 2
- User reads the application list
- User chooses the first App in the list
- It is unclear for the user, because there are only test comments
- The user tries the next App in the list. - There is no description
- User types in bicycle in the search area. - no result
- User clicks on help, a word document is downloaded.

- Question from the User, reading the Word document: „What is PMA?“
- After reading the document, the answer of the user is: „Its a marketplace for App development“

Task 3:

- 10:30 Test user starts with task 3
- types in „navigation“ in the search area, no result
- types in „routing“ in the search area, no result
- types in „navigation app“ in the search area, no result
- „best route“ – no results
- „destination“ - no results
- „app“ - 3 results
- clicks on “test” app - no description
- clicks on the “fake” app - no description
- searches for „app“ again
- the User can't find a navigation app and gives up.

Task 4:

- 10:34 Test user starts with task 4
- clicks on the first app in the list
- reads the description
- test user thinks that he's found the right page and finished the task

Task 5:

- 10:36 Test user starts with task 5
- entering Play Store on the Android smartphone
- Searching for „Simpli-City“
- a lot of different apps are listed
- entering „Simpil City“, no result
- entering „eco driving“
- Hint from Moderator: „You will not find it on the play store“
- The User is opening a browser (Chrome) on his smartphone and types in the address of the simpli-city marketplace.
- he finds the login page and zooms in.
- types in username and password
- User tries to install the app, but the button is not reacting.
- User says it was very difficult because he did not find the app on the PlayStore

Task 6:

- 10:43 Test user starts with task 6
- Types on the button logout on the desktop - Done
- Types on the button logout on the smart phone - Done.
- 10:47 finished – user states that it was easy

Test User 2, 09.09.2015:

Timeline:

- Introduction 11:14 - 11:20; Explaining example task 11:20 - 11:24, User Test 11:24 - 11:43, Debriefing Interview: 11:43 – 11:49

Task 1:

- 11:24 Test user starts with task 1
- Types in the address, comes to the login page

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 127 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

- The browser repeats the address from test user 1
- 11:26 task completed, the test user says it was relatively easy, but the address is very long

Task 2:

- 11:27 Test user starts with task 2
- it's not clear for the test user how he shall solve this task
- clicks on the first app.
- User #2 does not know exactly what the website is about
- test user says this task was very difficult

Task 3:

- 11:30 Test user starts with task 3
- types in „navigation“ in the search area, no result
- types in „destination“ in the search area, no result
- back to the app list
- clicks on the last App „Increased Mobility“
- test user states that this task was not so easy

Task 4:

- 11:32 Test user starts with task 4
- going back to the overview
- first App is the only one that has stars in the ranking
- the button does not look like a button
- clicks on the first app in the list
- reads the description, the test user thinks it's a little bit strange
- test user tries another App.
- He can't find more information in the detailed information than in the overview
- finished task – test user states it was not possible to find more information

Task 5:

- 11:36 Test user starts with task 5
- User #2 has no idea how to do this.
- He does not have experience with Android phone.
- Login with the username
- clicks on first App on the list, its already installed
- clicks on another App
- he has to login once again
- clicks on another App.
- Gets a message, but he finds this message weird/confusing
- he clicks on the app again
- gets the message again, nothing happens
- User #2 goes back to the overview and tries again;
- test user states he thinks it does not work, he gives up

Task 6:

- 11:41 Test user starts with task 6
- Types on the button logout on the smart phone - Done.
- Types on the button logout on the laptop - Done.
- 11:42 finished – test user states that it was easy

Test User 3 (User #3), 09.09.2015:

Timeline:

- Introduction 14:04 - 14:14; Explaining example task 14:14 - 14:18, User Test 14:18 - 14:47, Debriefing Interview: 14:47 – 14:55

Task 1:

- 14:18 Test user starts with task 1
- Types in the address and comes to the login page
- The browser suggests the address from test user 2
- 14:21 task was completed, the test user says it was very easy

Task 2:

- 14:21 Test user starts with task 2
- User #3 sees an app-list but it's not clear if it is the first page, clicks on the simpli-city logo.
- The page refreshes.
- User #3 clicks on 2 apps and reads the descriptions.
- Test user's answer (for completion of the task): „It is an overview of the application, and I can install it on my computer.“
- test user states that this task was difficult

Task 3:

- 14:25 Test user starts with task 3
- types in „navigation“ in the search area, enter does not work, clicks the search button, and waits for the result, not sure if further waiting is necessary, test-user thinks, maybe the internet connection is low and waits a bit longer, no result
- test user types in „route“ in the search area, finds one app „Increased mobility“ and clicks on the App.
- test user states that there should be more description, and there should be a note how big is the App.
- test user says that the task is finished, but it was not easy - test user rates it close to difficult

Task 4:

- 14:30 Test user starts with task 4
- going back to the overview
- test user knows how to find more information, because test user did it before (in task 2).
- Clicks on „Environmental Awareness Rising“
- there is not more information than in the overview, and there are no real comments.
- Size information is empty, test user states that it would be good to see a screenshot of the App.
- test user states that this task was a little bit difficult

Task 5:

- 14:33 Test user starts with task 5
- clicks on first App, but this is already marked as installed,
- so test-user clicks on the „increased Mobility“ App
- User #3 clicks on the install button
- Gets a message that the „installation is requested“
- test user searches for the App on the Computer via the Search function.
- test user can't find it.

- test user thinks it is not installed already, because it was too fast , and he could not find it

Task 6:

- 14:36 Test user starts with task 6
- Types on the button logout on the desktop - Done.
- test user states that it was easy

Test user wants a second trial of Task 5:

- 14:38 tries it on the smartphone to install an app.
- Types in the address in the browser.
- Page is not adjusted to the smartphone screen.
- User #3 thinks that's not good, because it is hard to navigate.
- test user logs in into the simpli-city marketplace website.
- test user clicks on install, test user has to login again.
- Login again, tries again.
- Types in „navigation“, no result
- Types in „route“, Enter does not work, test user has to scroll to the right on the page and click on the button.
- test user finds the App „Increased mobility“, test user clicks on the App.
- test user can't find any comments.
- test user goes back to the App list and clicks on „Environmental Awareness Rising“, but it's marked as installed already
- test user clicks on the simpli-city Logo to go back.
- test user clicks on „Increased Mobility“ App again
- Clicks on the Install Button, and gets a message „Installation requested“.
- Checks the phone if it has been installed, test user finds on the phone a Simpli-City App and opens this app, but this is not the requested App.
- 14:47 finished – test user says that it's probably not possible to install an app

Test User 4, 09.09.2015:

Timeline:

- Introduction 15:06 - 15:15; Explaining example task 15:15 - 15:19, User Test 15:19 - 15:35, Debriefing Interview: 15:35 – 15:42

Task 1:

- Test user starts with task 1
- Types in the address and comes to the login page,
- It's a very long address
- 15:21 task was completed, the test user says it was very easy

Task 2:

- 15:21 Test user starts with task 2
- User #4 thinks it's the Simpli-City marketplace, but to be sure he downloads the help document.
- He does not know what is a “PMA”.
- It's a kind of a marketplace, he thinks that he could install this on his phone.
- It was easy to find the help but it provides not much information.

Task 3:

- 15:25 Test user starts with task 3
- Clicks on „Search App“

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 130 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

- types in „navigation“ in the search area, clicks the search button.
- no result
- types in „navigate“
- finds „Increased Mobility“ App
- it was difficult, because there was no result on „navigation“

Task 4:

- 15:27 Test user starts with task 4
- going back to the overview
- Clicks on „Environmental Awareness Rising“
- there is not more information than in the overview...
- He expects: screenshots, information about which rights need the App, how many downloads, more detailed description than 2 sentences.
- very easy to find the relevant page, but no information provided there...

Task 5:

- 15:29 Test user starts with task 5
- Clicks on navigation App
- User #4 was logged out, logged in again
- Clicks on the App and then on the Install button.
- He thinks it does not work
- Clicks on the „test app“, gets a message.
- User #4 thinks he'll maybe get an E-Mail for the App on the “consumer” email address.
- test user states: “Maybe it should pair with my account with my smartphone, when I click “Install” on the desktop...”
- test user thinks that this task is very difficult, because it does not work

Task 6:

- 15:34 Test user starts with task 6
- Types on the button logout on the desktop
- 15:35: Done. - test user states that this task was easy

Test user 5 (User #5), 09.09.2015:

Timeline:

- Introduction 16:04 - 16:12; Explaining example task 16:12 - 16:16, User Test 16:16 - 16:38, Debriefing Interview: 16:39 – 16:49

Task 1:

- 16:16 Test user starts with task 1
- Types in the address (that's very long) and comes to the login page,
- Logged in and finished the task
- 16:18 task was completed, the test user says it was very easy

Task 2:

- 16:18 Test user starts with task 2
- Read through the Description of the first App „Environmental Awareness Rising“
- User #5 thinks the website is about Eco driving

Task 3:

- 16:20 Test user starts with task 3
- Clicks into the Search field
- types in „navigation“ in the search area, clicks the search button, no result

- types in „route“, finds „Increased Mobility“ App
- test user states that this task was not too difficult,

Task 4:

- 16:22 Test user starts with task 4
- test user states he would like to have a button „show all“
- Clicks on the „Amnesty“ App
- test user states that he would expect to get information about: what do I need (which operating system, or WLAN connection) and the price, pictures, clear headline and keywords, user rating, ...
- test user states that it was a very easy task, but that he had not found the expected information

Task 5:

- 16:25 Test user starts with task 5
- User #5 takes the smartphone and tries to find the app on the Google Play Store.
- Searches for „Amnesty“ App
- Hint from the Moderator: „You can't find it in the Play Store“
- User #5 looks on the simpli-city marketplace Website, if there is any download part...
- Clicks on „Help“ and downloads the Word document
- Reads through the chapter describing, how he can install an App.
- Goes to the „Amnesty“ App, and clicks on the Install Button.
- User assumes that it is downloaded on the laptop and he has to copy it onto the smartphone. – However, he cannot find it on the laptop.

Task 6:

- 16:30 Test user starts with task 6
- Clicks on the button logout
- 16:32 finished- test user states that it was easy

Test user wants a second trial of Task 5:

- 16:32 Trying task 5 on the smartphone
- Login to the SIMPLI-CITY marketplace in the smartphone's browser
- Test user is surprised that the site does not fit to the screen – the user thinks that the font is very small.
- Test user states that there should be more information about the project on the marketplace website
- Test user searches for eco driving App in Google Play Store
- Moderator tells the user that the SIMPLI-CITY app is not available on the Google Play Store
- 16:38 test user stops trying task 5 on the smartphone

13 Usability Inspection of the Use Case I related SIMPLI-CITY App

13.1 Approach Taken for the Usability Inspection

Starting from July 2015, a Usability Inspection was conducted by SRM every time when a new release of the WP7 related SIMPLI-CITY app “Increased Mobility” was available. The Usability Inspection method used was task-based Heuristic Evaluation as described in detail in Section 11.1.4. The results of each of these Usability Inspections were communicated to the developers immediately after each inspection, so that the app could be improved accordingly.

Table 28 below gives some general information regarding the expert who did the Usability Inspection, the place where the inspection was conducted, and some relevant technical details about the device used for the Usability Inspection.

Table 28: Usability Inspection(s) of the “Increased Mobility” App - General Information

| | |
|---|---|
| Evaluator Name / Organisation: | Giuseppe Liguori / SRM |
| Date(s) of Usability Inspection: | July – September 2015 |
| Place of Usability Inspection: | Bologna |
| Test-Device Model: | LG NEXUS 5 |
| Test-Device Operating System: | Android 4.4.4 |
| Internet Connection: | wifi |
| Preferences: | Installation of Apps from unknown sources enabled |

13.2 List of Usability Issues Found

The following Table 29 includes all potential usability issues that were revealed by the evaluator in the last Usability Inspection (end of September 2015). These potential usability problems found in course of the Heuristic Evaluation are ordered according to their severity, with the most critical problems on top of the list. The severity was assessed according to the following scale:

- 0 – I do not agree that this is a usability problem at all
- 1 – Cosmetic problem (is not necessary to be fixed)
- 2 – Medium severe usability problem (should be fixed)
- 3 – High level usability problem (imperative to fix this before product can be released)

Table 29: Usability Issues Found by Heuristic Evaluation

| # | Description of the Observed Usability Issue | Violated Heuristics ¹⁴ | Corresponding Screenshot Filename ¹⁵ | Severity |
|----|---|-----------------------------------|---|----------|
| 1 | SIMPLI-CITY Account creation (registration) - There is no possibility to register but only to sign in- If the account is still to be created there's no way to create it. The functionality is foreseen by the account menu but not working. The user is re-directed to a login mask on HTTP://130.83.245.97:4321/cxf/users/login but the system says "Self registration is not allowed. Ask administrator to register you". No description on how to ask to administrator and who is the administrator are provided. | 7, 10 | S1, S2, S3 | 3 |
| 2 | In the settings menu there is the Apps Settings menu that allows users to set Origin and Destination cities. However after having set them the PMA (Increased mobility > Navigate) asks again the user to "select the destination city" by VUI. | 1, 4 | | 3 |
| 3 | Increased mobility > Navigate> Select the destination city - No GUI interaction allowed (white screen, no list of cities available). | 1, 11 | | 3 |
| 4 | Increased mobility > Navigate> Select the destination city (Bologna) > Select the destination street in Bologna (Via parigi) > Select the destination street number for Via Parigi in Bologna (one) - the PMA shows a second main menu that is different and longer than the first one displayed as starting menu. In this menu there is the possibility to to select time slot (that makes sense) select transport mode (that makes sense) and show route that does not work at all, blocking the app and forcing the user to restart. | 1, 3, 5, 7 | S7, S8 | 3 |
| 5 | Increased mobility > Navigate> Select the destination city (Bologna) > Select the destination street in Bologna - No GUI interaction allowed (white screen, no list of cities available). | 1, 11 | | 3 |
| 6 | Increased mobility > Settings> Configure the emission level value > specify the value > (euro 3) - GUI - No possibility to come back to the main menu. The app must be restarted. | 1, 3 | | 3 |
| 7 | Increased mobility > Settings> Configure the Power source value > specify the value > (diesel)) - GUI - No possibility to come back to the main menu. The app must be restarted. | 1, 3 | | 3 |
| 8 | Increased mobility > Map > Position - the position is always unknown. | 1 | S5 | 3 |
| 9 | Increased mobility > Map > refresh button does not work and the position is always unknown | 1 | S6 | 3 |
| 10 | Increased mobility > Settings> Configure the Power source value > specify the value > (diesel) - GUI - No feedback is given to the input of the emission value. | 1, 3 | | 2 |

¹⁴ refer to Section 11.1.5 for details on the utilised List of Heuristics¹⁵ Related screenshots are provided in Section 13.2.1.

| # | Description of the Observed Usability Issue | Violated Heuristics ¹⁴ | Corresponding Screenshot Filename ¹⁵ | Severity |
|----|---|-----------------------------------|---|----------|
| 11 | Increased mobility > Settings> Configure the emission level value > specify the value > (euro 3) - GUI - No feedback is given to the input of the emission value. | 1, 3 | | 2 |
| 12 | The button back in the upper menu (close to the logo) does not work. | 1, 3, 4 | S9 | 2 |
| 13 | POIs button is apparently missing (D7.1.2 the section concerned is 6.1.4 POI/Blackspot Management). | | S7, S8 | 2 |
| 14 | Anomaly information button is apparently missing | | S7, S8 | 2 |
| 15 | Traffic diagnosis button is apparently missing. | | S7, S8 | 2 |
| 16 | When the app is open and the user does not interact for a short time, the app sometimes freezes and there's no possibility to go back to the menu. Even the selection of one of the options displayed does not produce any effect. The app must be restarted. | 1, 3 | | 1 |

13.2.1 Screenshots Corresponding to the Revealed Usability Issues

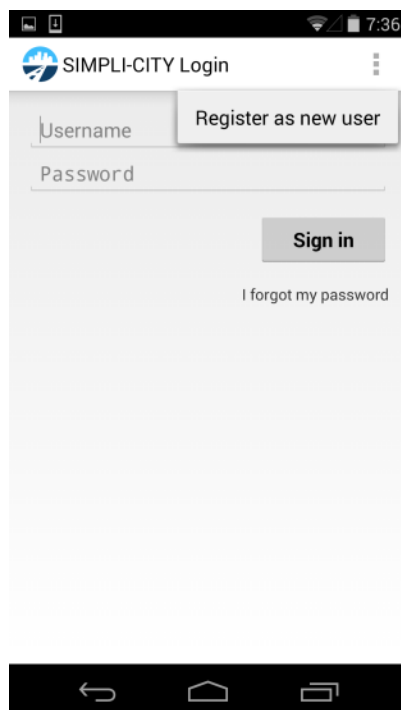


Figure 23: Screenshot S1

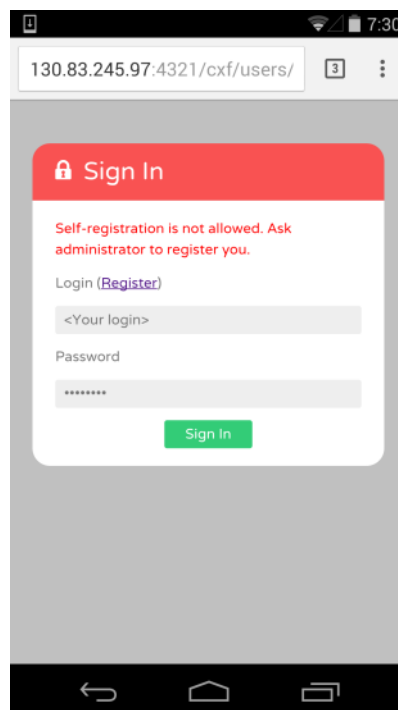


Figure 24: Screenshot S2

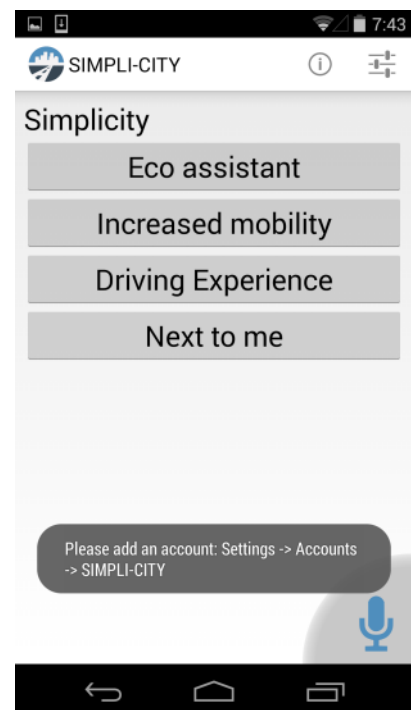


Figure 25: Screenshot S3

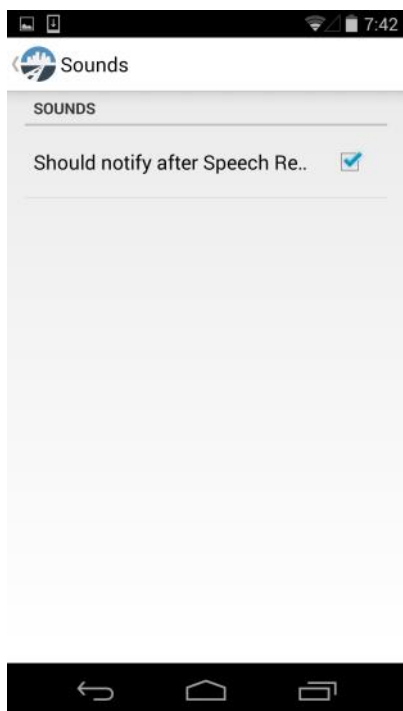


Figure 26: Screenshot S4

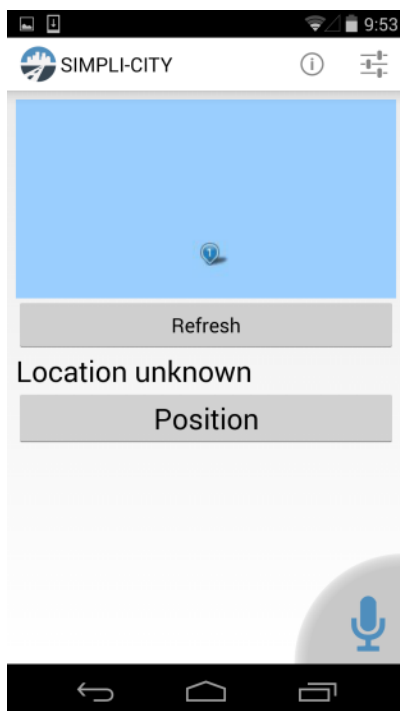


Figure 27: Screenshot S5

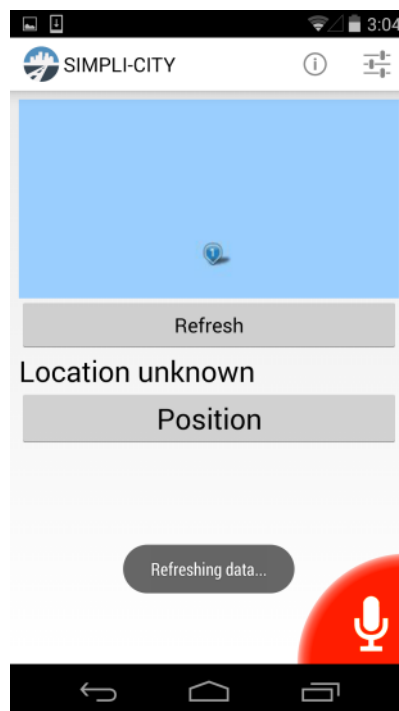


Figure 28: Screenshot S6

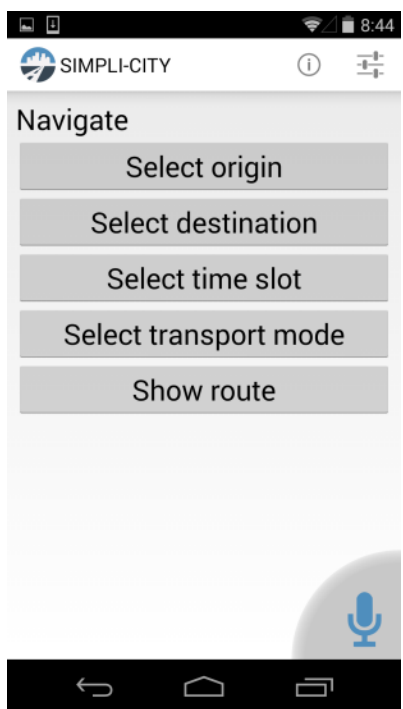


Figure 29: Screenshot S7

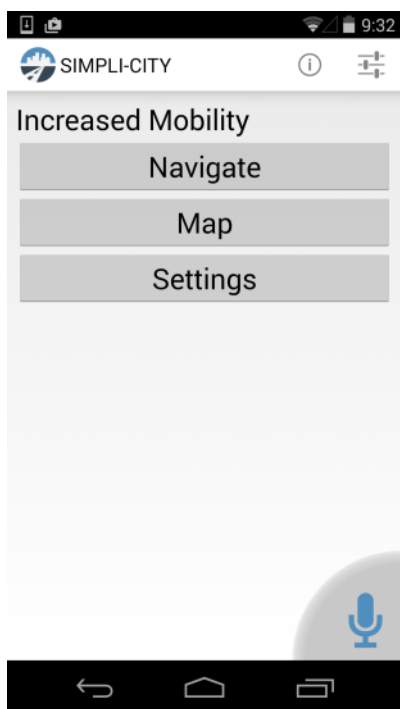


Figure 30: Screenshot S8

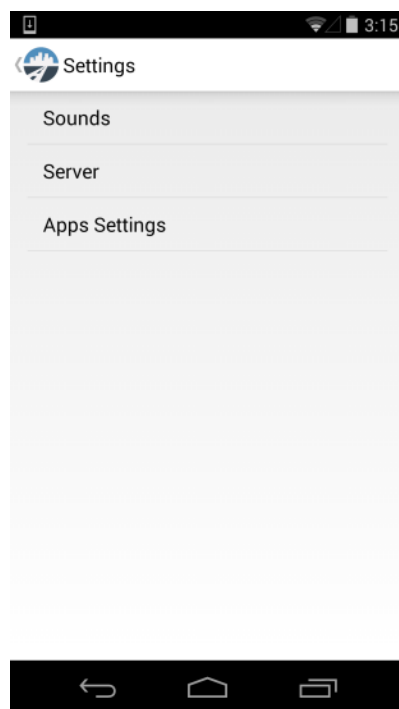


Figure 31: Screenshot S9

14 Usability Evaluation of the Use Case I related SIMPLI-CITY App

This chapter presents the results of the Usability Evaluation of the Use Case I related SIMPLI-CITY app from road-users' perspective.

In Section 14.1 the Usability Evaluation approach is described briefly. Section 14.2 gives a summary of the results of the Usability Evaluation, and Section 14.3 outlines the findings in detail. Furthermore, Section 14.4 includes the Task Sheets, which were given to the test-users during the evaluation sessions.

14.1 Approach taken for the Usability Evaluation

14.1.1 Aim of the Usability Evaluation

This Usability Evaluation was done in order to find out the main usability problems novice users encounter while trying the "Increased Mobility" app. It was a usability evaluation both on Vocal and on Graphic interface.

The results of this Usability Evaluation give valuable hints for further improvement of the SIMPLI-CITY system.

14.1.2 Method Applied

For the Usability Evaluation of the "Increased Mobility" app, the well-established Usability Evaluation method "*Thinking Aloud Test*" was applied: A test-user is asked to speak out loud all her/his thoughts while using the system to complete a given test-task. By observing the test-user's way of working on the task and by analysing the reasons for confusion, hesitation, or mistakes of the test-user, valuable hints for improvement of the usability of the system can be obtained.

14.1.3 Facilitator

The Usability Evaluation of the "Increased Mobility" app was organised and conducted by Giuseppe Liguori, working in SRM and strongly involved in the SIMPLI-CITY project activity. He dealt with the preparation of the testing sessions and ensured proper functioning of the technical equipment and software. He guided the test-users through the session as moderator, and took notes of revealed usability issues.

14.1.4 Test-Users

In total 4 test-users participated in the Usability Evaluation of the "Increased Mobility" app. All of them answered the introductory questionnaire, did the expected 4 test-tasks and provided their feedback in the debriefing interview and feedback questionnaire.

Three out of the four test-users were men and were working in SRM (2 employees, 1 consultant), while the fourth was a woman working as employee in the Municipality of Bologna. All of them were between 20 and 50 years old.

None of the test-users was directly involved in the development of the "Increased Mobility" app, but some of them have been involved in the SIMPLI-CITY project. So they knew the

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 137 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

main aim of the project and also the expected final outputs, even if they were not familiar with the application itself.

Table 30 gives an overview of the relevant skills and knowledge of the test-users as stated by them in the introduction questionnaire.

Table 30: Overview of the Test-Users' relevant Skills and Knowledge

| Test-user | User 1 MA | User 2 EM | User 3 MB | User 4 MO |
|---|--------------------------------|-----------------------|-----------------------|---------------------|
| Usability Evaluation Session | Session 1 | Session 2 | Session 3 | Session 4 |
| Gender | M | M | M | F |
| Age | 31-40 | 31-40 | 41-50 | 20-30 |
| Peculiarities | Glasses / contact lenses | NO | NO | NO |
| Mobile phone ownership | Other smartphone | Android smartphone | Android smartphone | Other smartphone |
| Smartphone usage since | > 1 year | > 1 year | > 1 year | > 1 year |
| Smartphone user expertise | High | High | Medium | Medium |
| Android user expertise | Medium | High | Medium | Low |
| Usage of Apps | High | High | Medium | Medium |
| Usage of voice commands | No | Seldom | Seldom | Seldom |
| Car driving expertise | Medium | Medium | High | Medium |
| Use smartphone in the car | Medium | High | Medium | Medium |
| Experience with routing apps | YES | YES | YES | YES |
| What do you know about SIMPLI-CITY? | Worked for | Worked for | Worked for | Heard of |
| Ever participated in a Usability study? | YES | NO | NO | NO |

14.1.5 Usability Evaluation Sessions

The Usability Evaluation of the “Increased Mobility” app took place in SRM's office in Bologna, Italy.

A dedicated usability testing place was arranged in SRM's office and equipped with two cameras recording the session. One camera filmed the device screen, and the activity of the hands of the test-user. The other camera pointed at the face of the test-user, in order to capture her/his expression and feeling while using the application. With this camera, in addition to the video, the audio was recorded as well.



Figure 32: Test Place Arranged in SRM's Office for the Usability Evaluation Sessions

Four usability evaluation sessions were done end of September 2015. The average duration of each Usability Evaluation Session was 25 minutes.

All the Usability Evaluation Sessions followed the same process: First the facilitator welcomed the test-user, described briefly the SIMPLI-CITY system, and explained the purpose of the session. Then the facilitator gave an overview of the test procedure and asked the tester for her/his permission that the session can be recorded, and the records can be used for later analysis and reporting purposes. After the test-user signed the consent form, the facilitator encouraged him to ask questions, whenever anything is unclear. In a short introductory interview, the facilitator obtained from the test-user some demographic information (gender and age), and some information regarding relevant skills and knowledge, such as familiarity with smartphones, Android and expertise with Thinking Aloud tests.

After this introductory part, the main part of the test-session could begin, in which the test-user was asked to complete the following evaluation tasks:

- Task 0: Use the SIMPLI-CITY information App to find out the SIMPLI-CITY speech server.
- Task 1: Insert settings related to your car (car plate, code of permission, emission level value, and power source).
- Task 2: Insert a destination within Bologna: Select an area within the city centre of Bologna or reachable from your current location by passing through the city centre.
- Task 3: Use the SIMPLI-CITY information App to find out the SIMPLI-CITY website address.

Section 14.4 includes the task-sheets, which the test-users got during the usability evaluation session.

Throughout the sessions, the facilitator used a predefined grid structured to simplify data collection.

After finishing the last test-task, the facilitator thanked the test-user for her/his cooperation and asked the following three questions in the debriefing interview: "How was it?"; "Was there anything that you found to be especially good?"; "Was there anything that you found to be especially bad?" When the test-user finished answering these questions, the

facilitator asked her/him to fill-in the System Usability Scale (SUS), a standardised questionnaire to assess a user's perception of the overall usability of a system. In case the test-user had further questions, these were answered by the facilitator and after that the session was finished.

After each Session, the facilitator took a time to control his notes and organize them as memory of the just-ended trial was vivid. Furthermore, the recordings were saved, and the material and setup was prepared for the next Session.

14.2 Main Issues Revealed During the Usability Evaluation

14.2.1 Issues Noted by the Facilitator

The Usability Evaluation of the “Increased Mobility” app demonstrated a confusion and low confidence by test-users while dealing with some tasks, despite the fact that the tasks were substantially basic. That was specifically noted by the facilitator during the tasks 1 and 2, but also task 3 raised some usability problems.

In general, often test-users were not sure that their input was really accepted by the application, being missing a confirmation or a feedback by the application itself.

Furthermore, in some sub-menus it was not possible to return back to the previous menu, and test-users were annoyed when forced to close the application and restart it again in order to go ahead the task.

In task 1, the test-users were not sure where the requested data was expected to be entered. They proceeded by chance and felt in wrong paths, and some test-users did not succeed at all in accomplishing the task. Maybe it would be more convenient to have only one section in which the user is allowed to enter user data, rather than two as it was in the tested version of the application.

Also in task 2, test-users needed some trials even those who successfully completed the task. Some test-users did not push on the screen to activate the microphone button, and they did it only when guided by the facilitator. In general, the use of the microphone button seemed not to be intuitive, and often the voice of the test-user overlapped with the system voice.

Although task 3 was successfully accomplished by all the test-users, some test-users would have appreciated, if they were allowed to access the SIMPLI-CITY website by simply taping on the address on the screen.

14.2.2 Usability Issues Revealed by the Test-Users

During the Usability Evaluation Sessions, the test-users mentioned several issues and aspects of the SIMPLI-CITY system, which should be improved:

1. Interaction with the system

Users 1 and 2: The start of application is too slow, and the screen remains dark for too long time. It is not clear if the system is starting or not.

Users 1, 3 and 4: The system does not confirm the actual acquisition of data. No feedback is provided to the user.

Users 2 and 3: The system does not show the user what data have already been inserted and saved.

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 140 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |

User 1: Saving data already inserted is not possible at user willing.

User 2: A “Return home” feature is not present.

Users 1, 2, 3 and 4: There are sections (i.e., Emission level, and Power source) where the user is cannot go back to the previous section. Restart of the application is necessary.

User 2: The system does not react to user commands.

User 3: User is not allowed to close the application by a dedicated feature within. The application can be closed only through a forced feature of the device.

2. Logic design

Users 1, 2, 3 and 4: Settings and user data are requested in different sections, not linked logically.

User 3: The application allows the user to insert the destination, even if the destination has already been inserted previously in a different section.

3. Vocal system

Users 2, 3 and 4: It is not clear how the user can activate the microphone to insert expected data via voice input.

Users 1, 2 and 3: The microphone activation is not comfortable.

User 3: It would be useful if the microphone showed that it is in active mode, and receiving sounds (e.g., by flashing).

User 2: The application does not work when using only the user voice.

4. User assistance

Users 1 and 3: The SIMPLI-CITY website is not linked within the application.

User 2: A search by text feature is not present in the application.

14.3 Detailed Documentation of the Findings

14.3.1 Usability Metrics

14.3.1.1 Task Completion Success Rate

Overall performance of each participant was evaluated depending on her/his ability to finish the task on her/his own. The task success rate score is the number of participants that finished the task successfully divided by the total number of participants.

Participants had some difficulties with Task 1 (GUI), and they were not completely able to complete it on their own. In Task 2 (VUI) the test-users failed in most cases because of problems with the vocal system. However in two cases out of four (i.e., Task 0 and Task 3) all test-users were able to complete the task without the intervention of the moderator. Task 3 was completed successfully by all the testers, as well as the Task 0, which was used as example task to introduce the test-users in the session.

Table 31: Task Completion Success Rate

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|--------------------------|-------------|------------|------------|-------------|
| | GUI | GUI | VUI | GUI |
| User 1 - MA | ✓ | ✓ | ✓ | ✓ |
| User 2 - EM | ✓ | - | ✓ | ✓ |
| User 3 - MB | ✓ | - | - | ✓ |
| User 4 - MO | ✓ | ✓ | - | ✓ |
| Task Success Rate | 100% | 50% | 50% | 100% |

✓ = successfully completed;

- = not completed / help needed for completion

14.3.1.2 Ease-of-Task Ratings

After each task, the participants rated the ease of completing the task, by answering the “Single Ease Question” (SEQ)¹⁶ “Overall this task was?” on a 7-point rating scale ranging from “Very Difficult”(1) to “Very Easy”(7).

Table 10 shows the ratings given to the single tasks by the test-users, and provides an average of these ratings for each task.

Table 32: Ease-of-Task Ratings

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|-----------------------|-------------|-------------|------------|----------|
| | GUI | GUI | VUI | GUI |
| User 1 - MA | 6 | 2 | 4 | 7 |
| User 2 - EM | 5 | 3 | 3 | 7 |
| User 3 - MB | 6 | 1 | 4 | 7 |
| User 4 - MO | 6 | 3 | 3 | 7 |
| Average Rating | 5.75 | 2.25 | 3.5 | 7 |

Task 3 has been perceived by the users as the easiest one, Task 0 has been perceived as pretty much easy, while Task 1 and Task 2 have been perceived as the most difficult ones. In particular Task 1 is rated as very difficult, and most probably some efforts in further developments are required in order to fix the main issues. Task 2 is right on the average score and also in this case further development is required in order to enhance the users’ experience.

14.3.1.3 Time on Task (TOT)

The amount of time needed to finish a task varies among the test-users. However, all test-users faced some problems with the vocal interface was affected by some system issues.

¹⁶ Jeff Sauro: “10 Things to know about the Single Ease Question (SEQ)”, online at <https://www.measuringu.com/blog/seq10.php>, visited 12.06.2015

Table 33: Participants' Time-On-Task

| Participant | Task 0 | Task 1 | Task 2 | Task 3 |
|------------------------------|------------|------------|-------------|------------|
| | GUI | GUI | VUI | GUI |
| User 1 - MA | 1.5 | 6.0 | 4.0 | 0.5 |
| User 2 - EM | 1.5 | 4.0 | 6.0 | 0.5 |
| User 3 - MB | 1.0 | 8.0 | 6.0 | 0.5 |
| User 4 - MO | 2.0 | 4.0 | 7.0 | 0.5 |
| Average TOT (minutes) | 1.5 | 5.5 | 5.75 | 0.5 |

14.3.1.4 Rating of the Overall System-Usability

After the completion of all tasks, participants were asked to fill in a standardised feedback questionnaire (SUS System Usability Scale).

As explained in Section 7.3.1.4, SUS yields a single number representing a composite measure of the overall usability of the system. SUS scores have a range of 0 ("not usable at all") to 100 ("perfectly usable system").

Table 34: Rating of the Increased Mobility application (SUS score)

| SUS Questions | User 1 MA | User 2 EM | User 3 MB | User 4 MO |
|---|--------------|--------------|--------------|--------------|
| 1. I think that I would like to use this system frequently. | 2 | 3 | 5 | 2 |
| 2. I found the system unnecessarily complex. | 4 | 3 | 3 | 3 |
| 3. I thought the system was easy to use. | 2 | 2 | 3 | 2 |
| 4. I would need the support of an experienced person to be able to use this system. | 2 | 1 | 2 | 4 |
| 5. I found the various functions in this system were well integrated. | 1 | 2 | 3 | 3 |
| 6. I thought there was too much inconsistency in this system. | 4 | 2 | 5 | 3 |
| 7. I would imagine that most users would learn to use this system very quickly. | 3 | 3 | 2 | 2 |
| 8. I found the system very cumbersome to use. | 4 | 4 | 4 | 4 |
| 9. I felt very confident using the system. | 2 | 3 | 2 | 2 |
| 10. I needed to learn a lot of things before I could get going with this system. | 2 | 2 | 3 | 3 |
| SUS Score (intermediate, per test-user) | 35 | 52.5 | 45 | 35 |
| SUS Score (overall average) | 41.9 | | | |

Analysis of the answers of the test-users to the feedback questionnaire reveals that the “Increased Mobility” app reaches a SUS score of 41.9. According to Bangor A. et al¹⁷, this SUS score can be translated to users’ perceiving that the usability of the system is “poor”. Confirming that the system is a prototype, this indicates that there’s still significant improvement of the system’s usability necessary.

14.4 Task Sheets for Usability Evaluation of the WP7 related App

TASK 0:

Use the SIMPLI-CITY App to find out the SIMPLI-CITY speech server.

Tell the moderator, when you finish this task.
Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐



TASK 1:

INCREASED MOBILITY > Insert settings related to your car (car plate, code of permission, emission level value, power source)

Tell the moderator, when you finish this task.
Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐



¹⁷ Adjective ratings related to SUS scores: 25 = "worst imaginable", 38 = "poor", 52 = "acceptable / o.k.", 74 = "good", 85 = "Excellent", 100 = "best imaginable" (Source: Bangor A., Kortum P., Miller J.: “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale”, in Journal of Usability Studies JUS, Vol. 4, Issue 3, May 2009, pp. 114-123)

TASK 2:

INCREASED MOBILITY > Insert a destination within Bologna: Select an area within the city centre of Bologna or reachable from your current location by passing through the city centre.

Tell the moderator, when you finish this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

List of suggested addresses

1. Via Parigi
2. Piazza Maggiore
3. Via Ugo Bassi
4. Via Marsala
5. Piazza Malpighi
6. Via Zamboni
7. Via Mascarella
8. Via Solferino
9. Via Goito
10. Via Monte Grappa

**TASK 3:**

Use the SIMPLI-CITY information App to find out the SIMPLI-CITY web-site address

Tell the moderator, when you finish this task.

Then answer the following short question:

Overall, this task was?

Very Difficult

Very Easy

☐ ☐ ☐ ☐ ☐ ☐ ☐

15 Summary and Conclusions

The evaluation strategy followed within SIMPLI-CITY aimed to evaluate the prototypes developed within SIMPLI-CITY against the requirements specified within deliverable D2.3 “Requirements Analyses Report”, and to validate the applicability of the SIMPLI-CITY prototypes in a real-world setting.

Evaluation of the SIMPLI-CITY Prototypes Against the Specified Requirements

The Technical Evaluation of the SIMPLI-CITY prototypes showed that the requirements, which were specified with priority “Must Have” or “Should Have” were almost completely fulfilled by the prototypes developed in the course of the project. Only 1 of the 38 “Should Have” requirements, which were defined during the specification phase of the SIMPLI-CITY system at the beginning of the project, was not fulfilled. This was mainly due to the fact that the specification details changed slightly during the course of the project:

The not fulfilled requirement is “U188 Composition of services”. According to the SIMPLI-CITY developers, this requirement was not covered within the final prototypes, since now the composition of services happens in apps, and not on the level of services.

Evaluation of the Applicability of the SIMPLI-CITY Prototypes in a Real-World Setting

In order to evaluate, how easy the SIMPLI-CITY prototypes can be used by real users (i.e., software developers and road-users), several Usability Inspection activities involving experts as well as Usability Evaluation activities involving test-users were conducted during the project. The Usability Inspection and Usability Evaluation were designed as formative evaluation: in order to allow the SIMPLI-CITY developers to implement further improvements to the system based on the findings of the evaluation, Usability Inspection/Evaluation was conducted in phases starting from June 2015 to September 2015. For all components of the SIMPLI-CITY system several useful hints for improvement of their usability were obtained by the Usability Inspection/Evaluation. Many of these improvements suggested by the results of the Usability Inspection/Evaluation were implemented by the SIMPLI-CITY developers immediately. However, of course not all of these usability issues could be solved during the runtime of the project, but the remaining usability issues are well documented and constitute valuable contributions for further developments of the SIMPLI-CITY prototypes.

In order to evaluate, how well the SIMPLI-CITY prototypes are functioning under real-world conditions, Functional Evaluation activities were carried out. On the one hand Functional Evaluation looked at the performance of the SIMPLI-CITY components dedicated to support software developers, when utilised for developing real services and apps within SIMPLI-CITY, and on the other hand also the performance of the Use Case I related app in a real car in real traffic in Bologna was evaluated. The results of these Functional Evaluation activities pointed out that the prototypes of the SIMPLI-CITY components are fully functioning in real usage, i.e., when really used for developing services and apps, and when used in a real car on the road. However, of course the results of the Functional Evaluation were also reflecting the prototype state of the SIMPLI-CITY system. The suggestions of useful additional functionality from a practical point of view, which resulted from the Functional Evaluation, are valuable and helpful hints for future development of the system.

| | | | | |
|---|--------------------------|--|----------------------|--------------------|
| D7.3_Evaluation_Report_v1.0_For_Approval.docx | Document Version: 1.0 | Date: 2015-10-30 | Status: For Approval | Page: 146 / 146 |
| http://www.simpli-city.eu/ | | Copyright © SIMPLI-CITY Project Consortium. All Rights Reserved. Grant Agreement No.: 318201 | | |