D1.8 Requirements for Scenarios and Prototypes
Third Version of D1.4/D1.6

Editor: István Koren
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Contributor(s)</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>31.10.2018</td>
<td>István Koren</td>
<td>ToC</td>
</tr>
<tr>
<td>0.2</td>
<td>23.10.2018</td>
<td>István Koren</td>
<td>Content</td>
</tr>
<tr>
<td>0.3.</td>
<td>30.11.2018</td>
<td>Ralf Klamma</td>
<td>Final Request for Contributions</td>
</tr>
<tr>
<td>0.4.</td>
<td>6.12.2018</td>
<td>Marco Fruscione, Chris Brox, Liliana Ravagnolo</td>
<td>Requirements from Testbeds</td>
</tr>
<tr>
<td>1</td>
<td>6.12.2018</td>
<td>Ralf Klamma</td>
<td>Final Version</td>
</tr>
</tbody>
</table>

**Disclaimer:** All information included in this document is subject to change without notice. The Members of the WEKIT Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the WEKIT Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
D1.8 Requirements for Scenarios and Prototypes

WP 1 | D1.8

Editor:
István Koren (RWTH)

Authors:
Ralf Klamma, István Koren (RWTH), Liliana Ravagnolo (ALTEC), Marco Fruscione (Ebit), Chris Brox (LT)

Reviewers:

<table>
<thead>
<tr>
<th>Deliverable number</th>
<th>D1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
<tr>
<td>Date</td>
<td>6.12.2018</td>
</tr>
<tr>
<td>Due date</td>
<td>M36</td>
</tr>
</tbody>
</table>
Table of Contents

Revision History .......................................................................................................................................................... 2

Executive Summary .................................................................................................................................................. 5

1. Introduction .......................................................................................................................................................... 6

2. Methodology ........................................................................................................................................................ 7
   2.1 TECHNOLOGY COLLECTION ......................................................................................................................... 8
   2.2 REQUIREMENTS ENGINEERING .................................................................................................................... 8
   2.1. TECHNOLOGY ASSESSMENT: HOUSE OF QUALITY .................................................................................... 10
   2.3 DEVELOPER WORKFLOW INTEGRATION ..................................................................................................... 13

3. Requirements for Prototypes .................................................................................................................................. 14
   2.4 USER INTERFACE AND USABILITY ................................................................................................................ 15

4. Requirements for Scenarios .................................................................................................................................. 16
   4.1 CASE 1 – AERONAUTICS ............................................................................................................................. 16
   4.2 CASE 2 - ENGINEERING (MEDICAL) ............................................................................................................. 18
   Physical Space Environment Interaction ............................................................................................................. 18
   4.3 CASE 3 - SPACE .............................................................................................................................................. 20

5. Conclusion ............................................................................................................................................................ 22

6. References ............................................................................................................................................................. 23
Executive Summary

As specified in our Description of Action (DoA), WP1 defines the framework and requirements to be used for all technical developments (WP2, WP3, WP4, WP5). Furthermore, the framework sets the foundation for piloting phases and their evaluation (WP6) and for exploring next steps (WP8). Within WP1, T1.4 Requirements for Scenarios and Technological Platform is responsible to set up the Requirements Bazaar methodology and toolkit to continuously collect, update, negotiate and correct requirements among the stakeholders involved. T1.4 is reported in D1.4 which is updated in M6, M21 (delayed to M23) and M36. The first version of this deliverable presented the methodology in detail and listed the outcome of the first idea collection of the WEKIT Community and described WEKIT’s three industrial scenarios. Based on these inputs, WP2 developed the first prototypes that were put into practice by WP6. Version D1.6 of this deliverable, presented the outcomes of the follow-up activities of the trials concerning the collection of new requirements. Version D1.8 of this deliverable presents the requirements extracted from the second trials performed in the industrial scenarios. These are documented in the Requirements Bazaar.
1. Introduction

This document is the last deliverable associated to T1.4 Requirements for Scenarios and Technological Platform. Based on the Requirements Bazaar methodology and toolkit, requirements were continuously collected, updated, negotiated, and corrected among the stakeholders involved. It is the third one in a series of deliverables providing input for scenarios (WP6) and prototypes (WP2).

This deliverable gives an overview of the requirements for scenarios and prototypes collected in the last year of WEKIT by the project partners. The general requirements process started at the WEKIT kick-off meeting in Milan, Italy, and continued on the Web-based Requirements Bazaar, which is a social requirements engineering platform developed for this purpose and successfully used in several major projects in the past. In particular, we opened up the platform and swapped out parts of it to the WEKIT Community website. In this way, we involved end users, designers and developers from the beginning in our decision processes. All the collected input has been processed using our House of Quality instrument. Considering the relevance of the collected requirements, the project partners designed and developed the AR-based technological WEKIT.one platform. The requirements elicitation process continued throughout the trials and this document provides an update on them.

In the following, we first describe the methodology leading to the results presented in this deliverable. We then list the current state of the requirements. It is important to note, that the current status in terms of prioritization and completeness is always recorded in Requirements Bazaar and is subject to change. Section 4 discusses possible changes and adaptions of the three industrial cases of WEKIT, based on the second trial runs. The deliverable concludes with a discussion on the process and gives an outlook how the remaining collected requirements will be used in the future.
2. Methodology

The requirements engineering methodology has been already described in the first version of this deliverable in M6 (D1.4) and updated in 1.6. In the following we take up the earlier reports and interweave them with actual activities carried out since D1.4 and D1.6. D1.8 is based on the lessons learnt in the second runs of the industrial trials.

Meeting the WEKIT challenges requires a sound methodological basis to develop the technological platform. The selected methodology relies on three activities. These are requirements engineering based on the activities involving end users; collecting and describing technologies that are available to support WEKIT scenarios and use cases; and employing an assessment instrument. The latter shall facilitate decision making by comparing products in terms of how well they support the elicited requirements. The requirements should play the role of a vehicle to transfer the framework pedagogical ideas into a clear set of technical requirements.

In order to document and compare the different technologies and to obtain a traceable approach to infrastructural decision making, the technology survey task was approached using the methodology (illustrated in Figure 1) that relies on three activities:

1. **Technology Collection**: The objective of this activity was to collect technological features that are potentially relevant and related to WEKIT using a desk research approach. See Section 2.1.
2. **Requirements Engineering**: The objective of this activity was to elicit, consolidate, and prioritize user requirements from different end-user sources and design activities in WEKIT following an open development approach. The requirements are the main ingredient to building a technological platform that serves the WEKIT objectives, as described in the other two WP1 deliverables - *D1.1 User Industry Needs and D1.3 WEKIT Framework and Training Methodology*.
3. **Technology Assessment**: Based on the artifacts obtained in the requirements engineering and technology survey activities, House of Quality (HoQ) (Hauser & Clausing, 1988) was adopted as an instrument for obtaining and assessing technical requirements to be met by technology products using the collected technologies and the prioritized user requirements.

![Figure 1. Technology Survey Activities and Outputs.](image-url)
In order to involve suitable stakeholders, a use case exercise was performed at the beginning of
the project with the WEKIT pilot partners Lufttransport, ALTEC, and Ebit. Moreover, we collected
input from the WEKIT Community of stakeholders by means of several activities and tools that
are explained in Section Error! Reference source not found.. The project partners were
collecting ideas to help developers push the boundaries in what will be possible with the
WEKIT.one technology platform - providing wearable experiences for knowledge intensive
training. During the idea collection phase (leading up to a project report in May 2016 and a more
extensive scenarios report in November 2016), the partners discuss these ideas - with respect to
whether they are possible, whether they address significant challenges that are currently
unsolved, and whether they bring out the best in what is technically feasible.

2.1 Technology Collection

The technological features (see Section 3) that may be relevant for WEKIT had been identified in
Task 1.3, which deals with the research and development of the WEKIT Framework and
Methodology, described in Deliverable 1.3.

2.2 Requirements Engineering

In parallel to the technology collection activity described in the previous section, there were
several initiatives in the project to elicit user requirements with end-user involvement. Software
architectures are built based on functional and non-functional requirements. In WEKIT, we
elicited two sets of the user requirements.

Early in the project 15 main use cases have been specified by the consortium members (see
Section 3.1). The requirements for them were elaborated during the technical meeting, which
took place in May 2016 in Oxford. These use cases can be found also in the WEKIT Community
Portal.

Additional comments from the WEKIT Community have been collected at the Kick-off Meeting in
January 2016 in Milano, then at the AR Hackathon in April 2016 in Aachen, as well as the JTEL
Summer School in June 2016 in Tallinn. Since the first report on requirements and scenarios,
further events where the requirements collection of WEKIT has been explicitly presented have
been the ORPHEE Alpine Rendez-Vous in Font Romeu, France, the Immersive Learning
Conference iLRN 2017 in Coimbra, Portugal, as well as the general meeting before the EC-TEL
conference 2017 in Tallinn. In 2018, the requirements have been presented at the third and final
community event at AWE Europe 2018 in Munich, Germany.

The functional requirements obtained were ingested into Requirements Bazaar
(http://requirements-bazaar.org) (Klamma, Jarke, Hannemann, & Renzel, 2011; Law, Chatterjee,
Renzel, & Klamma, 2012; Renzel, Behrendt, Klamma, & Jarke, 2013), a tool developed by RWTH
Aachen University in the context of the ROLE project. The Requirements Bazaar is a browser-
based social software platform (see screenshot in Figure 3) for Social Requirements Engineering
(SRE) addressing the challenge of a feedback cycle between users and developers in a social
networking manner. Stakeholders from diverse Communities of Practice (CoPs) are brought together with service providers (developers) into an open, traceable process of collaborative requirements elicitation, negotiation, prioritization and realization (Figure 2). A vital communication between all stakeholders of an open source project is essential in this regard (Hannemann & Klamma, 2013; Hannemann, Klamma, & Jarke, 2012). The Bazaar aims at supporting all stakeholders in reaching their particular goals with a common base: CoPs in expressing their particular needs and negotiating realizations in an intuitive, community-aware manner; service providers in prioritizing requirements realizations for maximized impact.

The Requirements Bazaar was used in the requirements engineering step for a collective voting process, in order to achieve a ranking of the elicited functional requirements, as an input for the quality function deployment (see Section 2.3). Partners were asked to express their opinion through casting a vote on the most important requirements. The vote consisted of a like on a certain requirement. Through this collective process, all the existing requirements were rated, enabling the prioritization of requirements. The ranking was constructed by sorting the requirements list according to the scores obtained after the voting procedure. A portion of the obtained prioritized list can be seen in Figure 3; the complete list is presented in Section 3.

Figure 2. Schema of Requirements Bazaar
Moreover, partners were also encouraged to comment on the requirements, in order to allow further refinement of the available requirements descriptions.

![Requirements Bazaar](image)

**Figure 3. Screenshot of Requirements Bazaar**

The resulting weighted list of requirements was end-user input to the House of Quality approach described in the next section.

### 2.1. Technology Assessment: House of Quality

We need to be able to make controlled technological decisions, which are informed by the actual needs of end-users. Therefore we chose to deploy a well-established methodology that allows us to map technical features offered by new and existing components with end-user requirements in the context of use, which will typically be defined by one or more design teams. The general methodology we chose is called Quality Function Deployment (QFD), and the particular instrument we adopted to map features and requirements is called House of Quality (HoQ) (Hauser & Clausing, 1988).

QFD is a methodology that aims to drive product design by customer requirements. Its instantiation is HoQ, a product development technique that follows the principles of QFD and has been originated in Japan in 1972 long before in the 1980s it was adopted by large U.S. firms such as Ford, Xerox and AT&T for their product development activities. The instrument allows identifying those parameters of a technology that are especially important taking into consideration the user requirements. In the case of the WEKIT technological platform, we use the
HoQ to get a weighted list of functional requirements based on the respective demands and needs of the design team scenarios.

HoQ establishes a matrix of requirements coming from both the customer and the engineers designing the product. Using this approach, user requirements can be transformed into a weighted list of engineering characteristics that need to be met by the candidate products. It also supports the assessment of existing technologies in terms of how well they perform when meeting the user requirements.

Figure 4 shows an exemplary scheme of each HoQ. On the left side, customer requirements (e.g. indoor navigation, experience recording, assembly guide) are entered one-per-row together with a weight calculated in user surveys. On the right existing products are rated by the end users (usually on a 0-5 scale, where 0 means “not possible with this product” and 5 “totally fulfills this requirement”), thus resulting in a market analysis. Engineering characteristics are entered on a column basis together with an improvement direction. At the bottom end of each column, improvement targets and the difficulty of reaching this target is recorded.

The House of Quality

![House of Quality Diagram]

In the next step, all engineering attributes are related with each other in the roof of the HoQ. Hereby positive or negative correlations in two graduations each are entered. That is, a plus is entered if on improvement of attribute A also attribute B is improved, and a minus is entered, if improvement of attribute A degrades attribute B at the same time.
The most important part of the HoQ methodology that also leads to the weighted engineering characteristics as output is setting the customer requirements in relation with the latter attributes. Hereby relations are rated in a numerical system with the higher number being higher related. A system is adopted that assigns strong relationships the value ‘9’, medium relationships a ‘3’ and weak relationships a ‘1’. Stronger relationships also lead to a stronger influence in the weight calculation at the bottom.

Another benefit of using HoQ is that it enables traceability of product attributes as for every weighted engineering characteristic the original user requirement can be traced in the matrix. In further steps, the output of one matrix may also be cascaded as input of a new one thus enabling traceability. For software products, further matrices may be applicable in terms of software modules within a broader infrastructure.

Figure 5. The Collaborative House of Quality Web Application

In the end, all the weights of customer requirements are charged against the product attributes according to their relationship factor. The output on the bottom is a list of weights for each product attribute that can then be incorporated in the product design. As described above, the results may be integrated into another HoQ matrix.
In WEKIT, two preliminary HoQs were built up to month 6 to test-drive the methodology. The user requirements elicited and then weighted through the voting process in Scenarios - Use Cases (component ARLEM) and the WEKIT Community Input (component Ideas) were used as input for the left part of the HoQ. The technical features from the WEKIT Framework (component Transfer Mechanisms) were instantiated for the technical part on the top of each HoQ. We found that the requirements were partially too general for enabling decision making using the Quality Function Deployment method.

To promote and explain the HoQ methodology and foster participation of the design teams, we used a collaborative Web application earlier developed in the Learning Layers project that is based on Google Drive and allows multiple persons to jointly work on a HoQ, as shown in Figure 5. The tool is also available on the Google Chrome Store:

https://chrome.google.com/webstore/detail/house-of-quality/jembpnanfpbdeklbcogaafjoelommfm

2.3 Developer Workflow Integration

The result of the House of Quality is a prioritized list of technical features to be met by the product. It is the input needed for developers. In WEKIT, the integration into the development process happens over the Bitbucket issue tracker. Bitbucket is a Web-based service used for source code and project management. It was chosen by WEKIT as the primary online repository for the project’s source code. Besides code hosting and code review management, it also features an issue tracker, which tracks the project’s feature requests and bug reports. Therefore, requirements collected in Requirements Bazaar have to be entered as issues into the Bitbucket issue tracker.
3. Requirements for Prototypes

In this section, the previous version of this requirement listed the collection of ideas and requirements collected in Requirements Bazaar. They had been processed and taken as input by WP2 for the development of the first prototypes that have been used in the trials. As planned, in the following we thus list the requirements that have emerged during the trials.

The requirements have been collected in our Requirements Bazaar in the WEKIT.ONE project:

https://requirements-bazaar.org/projects/155

Requirements Bazaar currently has around 520 users. Since the first version of this deliverable in M7, the WEKIT project collected 6 new categories. 76 new requirements were added, and 43 new comments with 13 new attachments were created. The prioritization efforts resulted in around 300 votes.

The project is currently divided into eight categories:

<table>
<thead>
<tr>
<th>Category Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics</td>
<td>What to do with the sensor data in terms of post-processing for learning</td>
</tr>
<tr>
<td>ARLEM</td>
<td>ARLEM standard</td>
</tr>
<tr>
<td>Content Store</td>
<td>Collect here all your ideas about a content store on the WEKIT Community page</td>
</tr>
<tr>
<td>Ideas</td>
<td>Community thoughts – unabridged and unfiltered</td>
</tr>
<tr>
<td>Lower level functional</td>
<td>This is the collector for all lower level functional requirements</td>
</tr>
<tr>
<td>requirements</td>
<td></td>
</tr>
<tr>
<td>Transfer Mechanisms</td>
<td>Transfer mechanisms as identified in Framework and Methodology</td>
</tr>
<tr>
<td>Trial Execution</td>
<td>Any aspect considering trials and how they should be performed can be raised here.</td>
</tr>
<tr>
<td>User Interface and Its Usability</td>
<td>New UI ideas following trial 1</td>
</tr>
</tbody>
</table>

During the requirements collection phase after the trials, two categories turned out to be of special interest: usability and analytics. Following this distribution, they are presented in the following sections.

The collected requirements were processed as outlined in Figure 6. The objective is to process the different types of input from various resources, in order to inform WP2 (technology platform) and WP6 (pilots) about the most important requirements for them. These two cooperate closely, as WP2 develops prototypes that will be tested in WP6. WP2 also integrates the technology provided by WP3-5 and complements the methodological framework from WP1 with a modular architecture.

The following three sections represent the status of the requirements collection at time of this deliverable. We sorted them according to popularity by vote and also give a summary of the submitted comments where available. The lists represent a 1:1 view on the current content in Requirements Bazaar, but typos have been corrected for improving the reading flow.
2.4 User Interface and Usability

In the following tables, we list requirements that were collected after the first trial. Most of them have been described in the deliverables 6.4, 6.5 and 6.6, following an extraction and refinement process before being collected on Requirements Bazaar to let them be discussed further.

<table>
<thead>
<tr>
<th>#</th>
<th>Hear voice in space</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 ★</td>
<td>(3D sound - works already) This is an important feature as it guides attention to where the expert said something (as you can hear the voice in space). Check: does this also work for moving ghost tracks?</td>
</tr>
</tbody>
</table>
4. Requirements for Scenarios

WEKIT deals with three different cases - Aeronautics, Engineering, and Space. The first version of this Deliverable 1.4 provided descriptions with requirements for the individual scenarios. Version D1.6 of this deliverable provided an update on these scenarios. In the following, we again updated the scenarios in close collaboration with WP6 based on the experiences in the second run of the industrial trials.

4.1. Case 1 – Aeronautics

The aeronautical case for the second trial will be NLG wheel (nose landing gear wheel) replacement. This task is like all other tasks on the aircrafts, specific steps set by the manufacturer. You cannot remove or add steps from it, although in this training scenario we have chosen to remove the steps where you grease the NLG wheel bearings for simple logistic reasons, to shorten the time of each trial and to avoid that the participant contaminate their clothes. The Goal in using the WEKIT platform on this task would be to first improve the learning of the participant and increase the quality of what the participant is learning by for example implementing expert advice. We would also free additional manpower needed for the training, usually experts.

The NLG Wheel replacement steps:

**Removal.**

Step 1/5:
Walk to the nose landing gear area and locate the nose landing gear wheel.

Step 2/5:
Localize the axle nut and compare the attached illustrations/3D models of the assembly with what you see to make yourself familiar with the nose wheel assembly.

Step 3/5:
Remove the cotter pin, the axle nut, the tabbed washer and spacer.

Step 4/5:
Remove the outer grease seal and the outer bearing and then remove the wheel from the axle.

Step 5/5:
Remove the inner bearing and the inner grease seal.

**Installation.**

Step 1/9:
Check the axle, the axle nut and the bearing cup for nicks, burrs or rough threads.
Step 2/9:
Carefully place the grease seal on to the axle against the inboard spacer with the legend “This side toward bearing” facing you. Then place the inboard bearing on to the axle and seat the bearing in the grease seal.

Step 3/9:
Carefully install the wheel on to the axle without damaging/nicking the axle while doing so. Make sure that the inboard bearing slides into the inner bearing cup of the wheel. Make sure that the seal is properly seated.

Step 4/9:
Install the outboard bearing on to the axle and seat the bearing in the bearing cup. Install the outboard bearing seal with the legend “This side toward bearing” facing the bearing and make sure it’s seated properly.

Step 5/9:
Install the outboard spacer and then the tabbed washer with the tab placed in the groove of the axle.

Step 6/9:
Install the axle nut on to the axle.

Step 7/9:
Tighten the axle nut to between 150 and 200 inch-pounds of torque while rotating the wheel to make sure of the proper seating of the bearing.

Step 8/9:
Back off the axle nut to zero torque, then torque the nut to 30 inch-pounds while rotating the wheel. Check to see that there is no side motion of the wheel.

Step 9/9:
Install the cotter pin. If the holes do not align, tighten the nut to the next available keying position.

The first aeronautical trial case we choose was the Pre-flight inspection’s 10 first steps, which we did a successful trial on. Our original plan was to keep the PFI (pre-flight inspection) for the second trial to see and monitor the advancement of the technology.

But after the first trial the WEKIT software has had a substantial amount of work done to it, and really evolved, and we felt that having a trial case witch nearly only consists of inspections and no hands-on work wouldn’t really do the software justice. Therefore, we felt the need to change the trial case to the NLG wheel replacement task instead, witch nearly only consists of hands-on work and would really put the WEKIT tool to the test.
We believe the NLG wheel replacement case will give a better picture and a pointer on how useful the WEKIT tool is and will be with even more development in terms of usability and the level of understanding the participant will have of the task ahead of him with the combined aircraft maintenance documentation and expert experience.

### 4.2. Case 2 - Engineering (Medical)

Starting from the outcomes of the first trial iteration, both with regards to the readiness of the prototype, and to the user feedback, it has been decided not to move to a different scenario but to consolidate and enhance the existing one during the second trial iteration. This decision does not mean to repeat the same trial, but to exploit the use case scenario with the following objectives both for trainer and trainees.

**Pedagogical requirements**

**Key goal in training:** performing interactive, personalized and self-guided training sessions.

Transferring expert experience by the most effective way is the key target of a teaching process: in the medical field the most important value of an expert is just his experience, which must be conveyed to students, so allowing them to become autonomous in the diagnostic process.

- From the students’ perspective, it is really useful to increase the amount of practical sessions, with a heterogeneous set of use cases and pathologies to analyze, and with the possibility to repeat some steps or receive specific advices or suggestions. To be really efficient, the technology readiness level of the AR tool must be high enough not to slow down the actual training process methodology.

Sometimes, during the first trial, in fact, the learners were much more focused on understanding the suggestions or looking for them in the 3d space than on the task.

**Objective for the re-enacting by the trainee:**

In the first trial use case scenario the WEKIT player and recording application were not fully integrated. At the same time, the WEKIT player UI did not allow some basic functionalities such as:

- Undo action
- Repetition
- Smooth interaction between the tools (video, images, etc.) and the real world

Such features are now expected in the new WEKIT prototype and participants could experience a new integrated version with improved functionalities and user interface design.

**Physical Space Environment Interaction**

One of the objectives of the second trial is to run and create the learning experience using different types of Ultrasound equipment as well different room locations, possibly within health organization facilities. This may be accomplished with the availability and management of advanced markers’ libraries within the WEKIT platform.

As an example of markers, it is required to be possible to use high quality-level photos of different keyboard equipment as well as a patient bed marker that would allow a better recognition of the
environment and of the relationships between objects in the 3d space and between physical objects and AR objects.

In the engineering healthcare use case scenario, the availability of IMU Sensors is especially desirable for better detection and recognition of the probe position during the teaching and learning phase of the Ultrasound examination.

- From expert doctor’s viewpoint, the support activity is time consuming and requires work shadowing sessions with one or a few students at the same time. Therefore, a tool for recording the trainer’s experience and re-enacting it several times with different students is especially useful because allows to share and put in practice the expert’s knowledge with more students.

Objective for the trainer capturing the experience:

- In the set-up phase of the first trial use case scenario, due to the prototyping status of the recording and capturing application of the WEKIT platform, the creation of the learning scenario and of the learning-capturing experience has not been performed directly by the end users. At that time the trainer defined on the paper the expert-capturing experience and the WEKIT partner implemented it.

- In the use case scenario of the second trial, one of the requirement is that the expert end-users may autonomously capture the experience with the WEKIT recorder, with the aim to better evaluate the usability in the autonomous creation of learning experience using AR tools, initially applied to simple learning procedures.

- It is expected to have and integrated recording and player solution in order for the expert to evaluate directly the learning contents created.

Generic learning goals: Supporting and enabling autonomous decision making during the diagnostic process.

- The students need ongoing training, to be updated about new diagnostic procedures or technologies applied to different clinical cases. When used to learn how to perform ultrasound examinations, the proposed steps should be adapted on the basis of the machine, the available probes and the anatomical region under analysis.
- The AR system should be able to adapt to the learning context and support the student with the proper augmented information.

Specific requirements for learning systems:

The learning procedure should provide different level of suggestions:

- Visual indications (arrows/circles, highlighted areas, etc.)
- Quick notes
- Verbal explanations, possibly on demand, that may be listened more times
- On demand, it should be possible to visualize short videos of expert’s experience or comparison videos that show the expected result of the present step
- Further clinical information to enrich the clinical value awareness of the performed step
The AR system should also be provided of sensors, tools and metrics useful to understand if the learner is in difficulties and if he needs to repeat the present step, receive more information or re-start the procedure (i.e. level of fatigue or distraction displayed in the HoloLens, measurement of time spent on a specific step, number of error, evaluated through the number of repetitions of a specific action / request / iteration...). Such an analysis is useful both during the training process and in the evaluation phase.

Training goals and requirements suggested by a medical trainer:

- Make the learner understand what (s)he has to do.
- Provide the learner all the information he needs to understand the context: he/she does not need only to learn how to perform a specific procedure, but needs to be trained to understand the clinical value of the medical images he/she is acquiring, comparing them with known case studies, in order to achieve a correct diagnosis.
- Allow a better level of practice and interaction in comparison with standard teaching activities.
- Give the possibility to update and correct recorded sequences.
- Give the possibility to create new training sequences starting from an existing one.

4.3. Case 3 - Space

Pedagogical requirements

**Key goal in training:** Changing the learning method.

Decreasing the ground training time is a key target: the aim is to shorten the time needed for training and to teach only the most important and necessary information before the space mission.

The astronaut training process needs to evolve taking into account the need to adapt to the different conditions of exploration missions. Currently, the ground training period for an ISS mission lasting 6 months is 18-24 months. A large portion of crew training aims at instructing them on procedures regarding the maintenance of items although the chances that such procedures are implemented on-orbit are quite small. If this estimation is projected on a Mars/Moon mission (lasting probably 2 years including the long distance journey to/from Mars), a training period of about 6 years seems to be necessary, but at the same time not feasible.

The trainers have to identify the training deemed mandatory on ground and develop innovative training tools to support learning objectives that are postponed to a later stage of the mission. The goal is to shift from long and comprehensive ground training periods to situational, self-directed autonomous learning in space.

- From astronaut viewpoint, it is not possible to be completely ready and prepared for an exploration mission before departure; they have to learn things during the mission with on-site learning/training systems.
- From trainer / company viewpoint, the longer the mission endurance is, the longer and more complex is the training. It is difficult to train the astronaut for every procedures and activities, so it is necessary something that might support the astronaut during the travel bringing the trainer experience on space.

**Generic learning goals:** Supporting and enabling autonomous decision making in space.

- The astronauts cannot be followed in real time during long missions. For example, the time lag between ground control and spaceship communication would be 6-20 minutes
(one direction) during a mission to Mars. The astronauts have to make decisions without
guidance from ground.

- The AR system should be able to adapt to the astronaut, record what happen while
performing a procedure and support the astronaut with augmented information.

**Increasing task performance speed in space:** Doing things quickly, well and safely.

- Trainer / company viewpoint: Time is a critical and expensive resource in space so it is
necessary to create a tool to reduce time and costs and to make the training more effective

- Trainee (astronaut) viewpoint: Reducing time especially in emergency and life-
threatening situations is important. Increasing task performance speed in such situations
would improve safety, especially considering that astronauts will be autonomous with
only limited support provided from Ground.

Note: The technology readiness level of the AR tool must be high enough not to slow down the
actual training process methodology.

**Specific requirements for learning systems:** Recognizing the learner’s level of expertise affects
the learning process, so the AR tool should create a personal training experience according to the
expertise of the astronaut.

- For inexperienced user, all the actions should be explained with limited telemetric data
(i.e. more text, symbols and object localization).

- For expert learner, the information visualized should be more oriented to symbols and
data, avoiding too much text and explanation (more detailed instructions should be also
made available when needed)

- Providing briefing and orientation (i.e. overall picture of the task) before the actual
performance is important.

- Analysis of the behavior through sensors. The data should be made available to the
astronaut during the training experience (i.e. level of fatigue or distraction displayed in
the HoloLens) and for post analysis.

Training goals and requirements suggested by an Astronaut trainer:

- Make the astronaut understand what (s)he has to do.
- Make the training experience more engaging
- Perform activities that with the actual methodology would not be part of the training
- Operate at different knowledge levels, i.e. when repeating well-known sequences the
system should not slow down or distract the operator with unnecessary information.
- Correct or modify the instructions with the stored material in case of inconsistencies,
giving the possibility to the astronaut to add notes or corrections.

**Scenario requirements (scripts)**

- The system should be able to show telemetry data and augmented information to
facilitate the execution of the procedure
- The AR content should be visualized in a realistic environment e.g. Virtual Environment
or real mock-up.
- The system should be able to record the experience and it should be able to make it
available also with low network connectivity.
- The system should be able to access to the existing data (e.g. maintenance reports) and
show them when requested.
30 new requirements have been entered altogether in the Requirements Bazaar. Voting and prioritization of requirements is still possible.

5. Conclusion

This deliverable provides an updated collection of requirements for the WEKIT prototypes and trials. It follows up on the first (D1.4) and second (D1.6) versions of deliverables reporting about Task 1.4 Requirements for Scenarios and Technological Platform.

Requirements Bazaar, House of Quality and Bitbucket issues are the main instruments for the requirements engineering processes in WEKIT. The WEKIT Community remains the main stakeholder engagement platform to capture ideas and needs of the project from external sources. Several activities in the context of exhibitions and scientific conferences have promoted the open requirements collection. Additionally, the WEKIT partners collected different sets of requirements – for user interface and usability, analytics and trial executions. They emerged at the trial activities and have been collected afterwards. The members of the project consortium as well as of the WEKIT Community commented on or rated them for prioritization.
6. References


WEKIT project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 687669. http://wekit.eu/