Smart and adaptive interfaces for INCLUSIVE work environment

Grant Agreement N°723373

Deliverable 1.4 – Summary of all verification and validation processes

Contractual delivery date: 2017-09-30
Actual delivery date: 2017-09-29
Responsible Partner: UNIMORE
Author(s): UNIMORE: Valeria Villani, Lorenzo Sabattini, Cesare Fantuzzi; TUM: Frieder Loch; SCM: Luigi Caiffa; KHS: Stefan Magerstedt; RTWHA: Julia Czerniak; SoftFactory: Rosario Maida; Grobotics: Panagiotis Lagios; SILVERLINE: Engin Talas.
WP n°: 1
WP leader: UNIMORE
Project Coordinator: Prof. Cesare Fantuzzi
Project Coordinator Organisation: UNIMORE
Dissemination level: Public

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N723377.
The author(s) is/are solely responsible for its content, it does not represent the opinion of the European Commission and the Commission is not responsible for any use that might be made of data appearing therein.
Executive Summary

This deliverable reports the plan for the tests that will be carried out to validate the INCLUSIVE system. Specifically, two kinds of tests have been considered: i) in-lab procedures for assessing the effectiveness of each module of the INCLUSIVE system, namely *measure, adapt, and teach* modules and middleware, and ii) procedures for testing the prototypal INCLUSIVE system.

On the one hand, in the first set of tests, the functionality of each module will be considered, e.g., tests will verify that the measure module can measure the users’ features identified in Deliverable D2.2, or that the middleware can effectively manage the communication and exchange of information in the system architecture.

On the other hand, the second set of tests will involve volunteers from the groups of target users identified in Deliverable D1.1 and the possible benefits coming to them from the use of the INCLUSIVE system will be assessed.
Table of Contents

Executive Summary .............................................................................................................. iii
Table of Contents ................................................................................................................ iv
1 Introduction .......................................................................................................................... 1
2 In-lab tests ........................................................................................................................... 2
  2.1 Measure module .............................................................................................................. 2
  2.2 Adapt module .................................................................................................................. 2
  2.3 Teach module ................................................................................................................ 3
    2.3.1 Participants and tasks ............................................................................................... 3
    2.3.2 Test of the off-line training system (developed in Task 4.2) .................................... 3
    2.3.3 Test of the on-line training system (developed in Task 4.3) .................................... 3
  2.4 Middleware .................................................................................................................... 3
    2.4.1 Test Unit Architecture ............................................................................................ 4
    2.4.2 ThingWorx test environment ..................................................................................... 7
    2.4.3 Additional test procedures ........................................................................................ 7
3 Plan for the tests on the prototype of the INCLUSIVE system ........................................... 9
  3.1 Use case 1: woodworking machine by SCM ................................................................. 11
  3.2 Use case 2: robotic cell of GIZELIS applied to SILVERLINE process .......................... 12
  3.3 Use case 3: production line by KHS ............................................................................ 13
4 Conclusion .......................................................................................................................... 14
1 Introduction

The INCLUSIVE project aims at developing methodologies for smart anthropocentric human-machine and human-robot systems that adapt to the human capabilities, skills and emotions and provide support when needed. In particular, three main groups of operators are considered, namely elderly, disabled, and inexperienced operators, since they are the most vulnerable ones suffering from the fast advances of technology in modern industry, as discussed in Deliverable D1.1. To achieve this goal, the INCLUSIVE system relies on three main modules, namely measure, adapt, and teach, which have been described in Deliverable D1.1. To allow a smoother development of such modules, the architecture of the system prescribes that they are developed separately and integrated by a middleware layer in the final prototype of the INCLUSIVE system, as described in Deliverables D1.3 and D5.1.

WP1 of the project focuses on the definition of the requirements and the overall architecture of the system. In this regard, this deliverable deals with the preliminary definition of the plans for assessing the effectiveness of the system, with respect to the desired behaviour defined by the specifications and requirements in Deliverables D1.1 (technical requirements) and D1.2 (safety and ethical, legal and social requirements). Specifically, tests are needed to firstly validate each module of the INCLUSIVE system (measure, adapt, and teach modules and middleware) and secondly assess the overall effectiveness of the proposed methodology by considering real end users interacting with the demonstrators developed in WP7.

In the following, Paragraph 2 reports the plan for in-lab tests, which will consist in assessing the functionality of each part of the INCLUSIVE system (measure, adapt, and teach modules and middleware), without considering, at that stage, the integration in the final system and the involvement of users from the use cases. Then, Paragraph 3 describes the plan for testing the complete INCLUSIVE system. This set of tests will be carried out by using the prototypal demonstrators that will be integrated for the three use cases considered in the project. For the time being, a rough plan for tests is provided, and it will be refined during the next year of activities. Finally, Paragraph 4 follows with some concluding remarks.
2 In-lab tests

In-lab tests will be performed at the end of the development of three modules of the INCLUSIVE system, namely measure, teach and adapt, and the middleware:

- WP2: Human capabilities measurement
  - Task 2.6: In-lab prototype development, from M16 to M20;
- WP3: Adapt interfaces to human capabilities
  - Task 3.4: Integration of adaptation modules to industrial use cases targets and in-lab tests, from M18 to M24;
- WP4: Teaching and training for unskilled users
  - Task 4.5: Integration of the training module to industrial use cases targets and in-lab tests, from M21 to M24;
- WP5: Adaptive automation middleware
  - Task 5.4: Integration with machines from use cases and tests, from M22 to M24.

The aim of these tests is to verify the functionality of each element of the INCLUSIVE system, irrespective of its effectiveness, which will be tested in the second set of tests described in Paragraph 3. As a consequence, in-lab tests do not require the enrolment of subjects from the use cases. Conversely, researchers and students from the responsible universities will be involved for testing the measure, teach and adapt modules, whereas tests for the middleware will consists mainly in software tests to verify the communication among the elements of the INCLUSIVE system.

2.1 Measure module

The measure module is modular and directly connected to the HMI. As described in Deliverable D2.2, measurements of human capabilities will be a priori, real-time and longitudinal:

- A priori measurements are an “offline” assessment, consisting in questionnaires regarding demographic questions, or tests for perceptive, cognitive or motoric capabilities. Moreover, skills and constitutional characteristics will be queried by questioning.
- Real-time measurements consist in measuring physiological indicators for mental strain, such as pupil diameter, blinking rate, skin conductance, cerebral activity, body temperature, hormonal balance and heart rate.
- With longitudinal measurements, performance indicators will be tracked, e.g. time for decisions, executions steps for the task, mistakes, and redundancies.

Preliminary in-lab tests aim at verifying that the measure module is able to detect mental strain and monitor user’s interaction with the system. The tools used for the a priori measurements, which have been extensively described in Deliverable D2.2, are well known tests used in the literature and in practice to measure perceptive, cognitive or motoric capabilities. The ultimate aim of in-lab tests for the measure module is that the user profiles, which have been described in Deliverable D2.1, can be derived from the planned measurements. To this end, tests will be carried out considering subjects with known profile and assessing whether the measure module succeeds in recognizing the correct profile. For the time being, the involvement of about 30 subjects is planned for preliminary in-lab tests of the Measure module.

2.2 Adapt module

The adapt module consists in the development of a methodology to adapt the HMI according to user profiles. Specifically, some adaptation rules will be devised to adapt the interaction to the user’s measured perception and cognition capabilities and to provide the most suited interaction technology according to the user’s features. Thus, the aim of the in-lab tests for this module is to assess whether the proposed adaptation rules match the needs of subjects belonging to different user profiles.
In particular, mockups of the HMIs implementing the adaptation rules will be used. Approximately 10 to 20 subjects from UNIMORE who simulate or approximate the user profiles considered in WP2 will be enrolled (e.g., deafness might be simulated by using earplugs). The improvement in the interaction provided by the adaptation rules will be measured by replicating in small scale and as long as possible the testing scenarios on prototypes described hereafter in Paragraph 3.

2.3 **Teach module**

The aim of the in-lab tests is the validation of the technical functions and the identification of software bugs of the offline and online training system. Therefore, users will be asked to carry out representative tasks with the developed training system.

2.3.1 **Participants and tasks**

The participants will be trained on scenarios that are representative for the use cases of the project (e.g., changeover tasks, troubleshooting tasks). The tests can be carried out in the labs at TUM where the technical infrastructure and demonstrators that can serve as testbeds for the trained procedures are available. About 30 subjects will be recruited for the in-lab tests of the *Teach* module.

2.3.2 **Test of the off-line training system (developed in Task 4.2)**

The offline training system runs separate from the INCLUSIVE infrastructure. Techniques to measure strain that were developed in WP2 will be used. An interface to the INCLUSIVE system to write information about the training results for the use by other components will be evaluated if necessary.

The effectiveness of the training will be measured by evaluating whether the participants are able to perform the trained task (e.g., number of errors, time to execute task) better than with a baseline system that does not provide support to the user. Furthermore, it will be evaluated whether the training system adapts as planned within a training (e.g., reduction of complexity if strain or errors are detected) or across multiple training sessions (e.g., advice to repeat a lesson in case of too many errors).

2.3.3 **Test of the on-line training system (developed in Task 4.3)**

The test of the on-line training system should validate whether errors in the execution of a task are detected. Hence, the experiment will be designed to deliberately cause errors or strain that the tracking of the execution has to detect. Measurement techniques (e.g., eye tracking, analysis of speech) will be used to detect problems supplemented by the logging of the users’ actions on the interface of the machine. It will be evaluated whether the suggested training units are appropriate for the problems that the users encountered.

The effectiveness of the provided training should be evaluated thereafter, by testing the performance of the user before and after the suggested on-line training units.

2.4 **Middleware**

In INCLUSIVE the adaptive automation middleware represents the connectivity layer for machine/robot-sensors, HMI and the three INCLUSIVE Modules (*measure*, *adapt* and *teach*).

By designing and developing the INCLUSIVE middleware, one important factor to consider is the set-up of a testing environment in order to:

- test that program units and their interfaces are brought together (through Unit tests, WP5),
- test the entire system to verify that system requirements including specified functions, quality, and performance characteristics are met (System tests, WP5 and WP7), and finally
• test the system by actual end users to ensure that it is ready for delivery and use (Acceptance tests, WP7).

In general, software testing involves the execution of a software component or system component to evaluate one or more properties of interest. These properties indicate the extent to which the component or system under test:

• meets the requirements that guided its design and development,
• responds correctly to all kinds of inputs,
• performs its functions within an acceptable time,
• is sufficiently usable,
• can be installed and run in its intended environments, and
• achieves the general result its stakeholders desire.

To fulfil these operational needs, the INCLUSIVE middleware make uses of a Test Unit, represented (mainly) by Graphical User Interfaces (GUIs), through which a bidirectional exchange of information between middleware and INCLUSIVE components can be established and monitored (Figure 1).

![Figure 1: Schematic view of the middleware Test Unit, comprising GUls with widgets and Services to proof the connectivity of the middleware with both Progea-HMI as well as with device-controller in terms of: established connection, communication speed, completeness of exchanged data, proof of input/output values (e.g., out of range), alert, etc.](image)

2.4.1 Test Unit Architecture

The middleware Test Unit, which is shown in Figure 2, is designed to perform tests and simulations at different levels:

1. (thingWorx) middleware controller with the INCLUSIVE modules;
2. (kepware) middleware interface for the communication with HMI and PLC/Controller;
3. (MSSQL Unit) database as repository of all user profiles/technical data for comparison with measured data

Techniques that exploit “capture and replay” tools will be used in the tests: these techniques capture events from the tested application and use them to generate test cases (over a PTC Integrity Modeler) that replay the actions performed by the user. Thus, developers will be able to perform lightweight testing
processes (Simulations) that only take into account the required elements, actions and properties of the GUI to be tested, avoiding the rest of unneeded test.

Following such approach criteria of scalability, tolerance to modifications, correctness, robustness and usability of HMI- and PLC/Controller-components can be easily tested.

Additionally, tasks as:

- analysis and troubleshooting of defects (serious departure from the specific behaviour) and
- surprises (user-recognized departure from the specific behaviour, but not explicitly indicated in the specification)

will be referenced by the middleware Test Unit, too.

Tests/Simulation of
INCLUSIVE Modules

Tests/Simulation of
HMI/PLC

Figure 2: The middleware Test Unit architecture. With “unit” are referred ThingWorx developed GUIs within the middleware, that are responsible for communication and for monitoring processes as data-exchange between components (HMI, Databases, machine-controller, modules, etc.).

Once the application to be tested is launched, the developer interacts with the Unit-GUI which generate GUI events (i.e. based on the information gained from the three inclusive modules) that are automatically captured and stored into a test case. This test case can be modelled and used in simulation processes (Figure 3).

The generated test cases can be completed by adding new actions or meta-events in order to insert messages, verification points, or anything that can help to refine it. The process of capturing, executing, and analyzing executions is an example of Observation-based Testing.
Figure 3: An example of test/simulation of test cases by using A) state machine diagram and B) structure diagram representation of the test (from Use Case to Test Case). Modelling is performed with PTC Integrity Modeler.

Figure 3A shows a test simulation scenario, by using the state machine diagram functionality of the PTC Integrity modeler. This tool allows to mimic representational states of test cases and can be programmed to represent real machine components with known responses to inputs. A proof of the resulting output with expected responses (for instance from value in database) will give a “yes” or “no” test response. This technique allows for carrying out test procedures without the need to connect to physical machine components.

A schematic demonstration on how the Test Unit could be implemented for simulation tests is shown in Figure 3B, with reference to the use case by KHS. The PLC element of the KHS Innoket Neo labelling machine can be started/stopped/controlled by the KHS-proprietary HMI, which will be integrated with extensions developed within the INCLUSIVE project to (for example) read and monitor the stress level of an operator during some operation on the machine (for instance by starting a fault recovery procedure).

The measure module integrated in the middleware can measure (dynamically) values of hearth rate of the operator and pass these values to the middleware that forward them to the HMI-INCLUSIVE extension. Here these values are compared with a priori defined values in the MS SQL Database to determine the stress level of the operator and dynamically conform (upon INCLUSIVE HMI-criteria) the complexity of flow of information presented to the operator.
2.4.2 ThingWorx test environment

For test purposes, an additional demonstration example for an adaptive HMI customized for the three INCLUSIVE use cases, using the ThingWorx Platform will be developed. This will be used to:

- test the middleware against the HMIs (mimic of Progea-HMI),
- obtain information during development phase of the middleware.

The communication between the “dummy” test environment, middleware controller and middleware interface is OPC UA based, for further information please refer to Deliverable D5.1.

The ThingWorx developed HMIs will reflect those developed by Progea for each of the three use cases (see Deliverable D.1.3) in order to carry out dynamic real-time testing (including simulation) of the middleware controller along the different development phases.

2.4.3 Additional test procedures

Beside the Test Unit designed within the middleware, some standard and free available tools will be implemented to carry out some specific tasks.

2.4.3.1 Test of the OPC UA interface between middleware and HMI

Open standards such as the OPC specifications are created with the aim of permitting interoperability between products from different manufacturers.

![Diagram: OPC Server test environment for Progea Automation Platform.NExT™, KEPServerEX® and ThingWorx®.](image)

Tests to monitor OPC UA activity in INCLUSIVE (i.e. connectivity to HMI over the Progea Automation Platform.NExT™, connectivity to external IT Systems (like MES, ERP), performance ...) will be implemented by using UaExpert®, a full-featured OPC UA client. The UaExpert is designed as a general purpose test client supporting OPC UA features like Data Access, Alarms & Conditions, Historical Access and calling of UA Methods. It uses the sophisticated GUI library QT from Nokia (formerly Trolltech) forming the basic framework which is extendable by Plugins.

Test cases will be designed to verify the behaviour of the OPC UA-based Kepware by using both valid parameters when making calls as well as invalid parameters. It is important to verify that not only server connectivity and performance runs as expected, under normal conditions, but also to monitor if a client is behaving as it should be. The results are finally recorded in a file and can be issued as a summary.
2.4.3.2 Test of the interface between middleware and MS SQL DB (ODBC)

Microsoft ODBC Test is a utility created by Microsoft that accompanies the Microsoft ODBC Software Development Kit. It is a more advanced tool which allows to call most ODBC API functions. There are 32-bit and 64 bit, ANSI and Unicode versions available of this utility. Through this tool the connectivity with middleware controller can be easily tested and monitored at any time.

2.4.3.3 Test of the interface between middleware and MS SQL DB (JDBC)

JDBC includes four components (JDBC API, JDBC Driver Manager, JDBC-ODBC Bridge and JDBC Test Suite). The JDBC Test Suite driver helps to determine that programmatic access to relational data from the Java™ programming language work correctly, i.e. applications can execute SQL statements, retrieve results, and propagate changes back to an underlying data source.

2.4.3.4 Test of the interface between middleware and REST API

There are many open core technology REST API Test Tools, easy to use without writing any scripts (SoapUI Pro, Jmeter, Advanced RESR client, Fiddler, DevHttpClient Plugin for chrome). These tools support extensive testing of RESTful web services and their resources, representations, etc. ensuring that APIs perform as intended and meet the business requirements in a single unified framework.
3 Plan for the tests on the prototype of the INCLUSIVE system

Although the prototypal INCLUSIVE system will be built in WP7 by developing three demonstrators for the considered use cases, at this early stage it is important to consider the tests that will assess the effectiveness of the methodology devised in the project, with respect to the system requirements. Figure 5 depicts the process leading to the preliminary definition of the plan for the tests on the final demonstrators of the INCLUSIVE system, which will be carried out in WP7.

![Figure 5: Process leading to the plan for the tests on the final demonstrators of the INCLUSIVE system.]

<table>
<thead>
<tr>
<th>Technical requirements</th>
<th>ELSI requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>[T-R1] The interface adapts to the level of skills of the operator</td>
<td>[ELSI-R1] The system prevents inducing strain itself</td>
</tr>
<tr>
<td>[T-R2] The system can be used by low educated operators</td>
<td>[ELSI-R2] The system considers anonymized personal data</td>
</tr>
<tr>
<td>[T-R3] The system can be used by physically and cognitively impaired operators</td>
<td>[ELSI-R3] The system uses collected data not for any disadvantage for the employer</td>
</tr>
<tr>
<td>[T-R4] The system can be used by people with low computer skills</td>
<td>[ELSI-R4] The system depicts relevant user requirements and prevents discrimination</td>
</tr>
<tr>
<td>[T-R5] The system enforces the correct procedures</td>
<td>[ELSI-R5] The system should meet all relevant safety criteria</td>
</tr>
<tr>
<td>[T-R6] The operator feels satisfied from the interaction experience</td>
<td>[ELSI-R6] The system should not distract the operator</td>
</tr>
<tr>
<td>[T-R7] Interaction with the system generates a low level of stress for the operators</td>
<td>[ELSI-R7] The system should not cause injuries by means of inductive measuring technology</td>
</tr>
</tbody>
</table>

Table 1: Technical and ELSI requirements for the INCLUSIVE system, as discussed in Deliverables D1.1 and D1.2, respectively.

First of all, for the sake of clearness, Table 1 reports the technical and ELSI (Ethical, Legal and Social Implications) requirements for the INCLUSIVE system, as derived in Deliverables D1.1 and D1.2. These are the starting point for the definition of the test scenarios of the INCLUSIVE project. Thus, with reference to the process shown in Figure 5, starting from the technical and ELSI requirements, it was analysed which quantities need to be measured in order to verify that the INCLUSIVE system satisfies the requirements and, thus, is beneficial to the users. Table 2 lists the measurements planned to verify the compliance of the INCLUSIVE system with the requirements.
<table>
<thead>
<tr>
<th>Measurements</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• time required with/without the INCLUSIVE system;</td>
<td>T-R1, T-R2, T-R3, T-R4</td>
</tr>
<tr>
<td>• longitudinal assessment from WP2;</td>
<td></td>
</tr>
<tr>
<td>• number of errors with/without the INCLUSIVE system;</td>
<td></td>
</tr>
<tr>
<td>• effectiveness of the approach by means of usability subjective testing (e.g., questionnaires): investigate if the guided procedures are perceived as helpful or unnecessary</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• time required with/without the INCLUSIVE system;</td>
</tr>
<tr>
<td>• longitudinal assessment from WP2;</td>
</tr>
<tr>
<td>• number of errors with/without the INCLUSIVE system</td>
</tr>
<tr>
<td>• effectiveness of the approach by means of usability subjective testing (e.g., questionnaires): investigate if the guided procedures are perceived as helpful or unnecessary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• strain with/without the INCLUSIVE system;</td>
</tr>
<tr>
<td>• eye tracking, pupil diameter and gaze analysis</td>
</tr>
<tr>
<td>• eye tracking and, if applicable, longitudinal measurements;</td>
</tr>
<tr>
<td>• eye tracking and, if applicable, longitudinal measurements;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
</table>

Table 2: Evaluation of the prototypal INCLUSIVE system: measurements planned to verify that the technical and ELSI requirements are satisfied.

It is worthwhile noting that the compliance of not all ELSI requirements can be measured and verified by tests with users. In some cases, it is verified by design. In particular, the requirements ELSI-R2, ELSI-R3 and ELSI-R4 are guaranteed by the prescriptions on ethics issues described in Deliverables D10.1 and D10.2. The requirement ELSI-R5 is guaranteed by the analysis of safety related functions and requirements presented in Deliverable D1.2 and performed in WP6. Finally, the requirement ELSI-R7 is verified since non invasive measuring technologies will be adopted, as described in Deliverable D2.2.

Moreover, starting from the working scenarios defined for each use case in Deliverable D1.1, the scenarios considered in the tests, called evaluation scenarios, are defined for the three use cases. These specify what tasks the study participants will be asked to perform while the effectiveness of the INCLUSIVE system is measured. At this stage, a general description of the evaluation scenarios is defined; practical details and implementation will be defined in WP7. As part of the evaluation scenarios, the most frequent errors that are likely to occur when using the traditional interaction systems (as defined in Deliverable D1.1) are found: the goal of the INCLUSIVE system is to improve the interaction approach and, hence, decrease the occurrence of such errors.

Finally, with regards to the test participants, users from the use cases (operators employed at the partners responsible for the use cases or their customers) will be enrolled. In particular, subjects with different expertise and capabilities will be considered, in order to assess the effectiveness of the INCLUSIVE system with the target users identified in Deliverable D1.1. The following table shows the user groups under consideration in the INCLUSIVE project.
The following paragraphs report the evaluation scenario, the possible errors and the planned test participants for each use case of the INCLUSIVE project. **The number of enrolled participants are rough estimates for the time being and will be refined in one year, that is at the beginning of WP7 (M25).**

### 3.1 Use case 1: woodworking machine by SCM

#### EVALUATION SCENARIOS

**ES1.1:** Users are asked to set up the machine for a given working process (the same for all users): specifically, they will have to tune the tools warehouse and the working area components;

**ES1.2:** Force an alarm of ordinary maintenance and ask users to solve it by AR teaching module

#### FREQUENT ERRORS WITH TRADITIONAL INTERACTION SYSTEMS

**ES1.1:** Tuning of the tools warehouse

- tune the real tools warehouse and don’t tune the virtual tools warehouse on the HMI
- tune the virtual tools warehouse on the HMI and don’t tune the real tools warehouse
- tune the real tools warehouse with a wrong tool
- tune the real tools warehouse with a tool in wrong position
- tune the virtual tools warehouse with a wrong tool
- tune the virtual tools warehouse with a tool in wrong position

**ES1.1:** Tuning of the working area components

- tune the real working area and don’t tune the virtual working area on the HMI
- tune the virtual working area on the HMI and don’t tune the real working area
- tune the real working area with a wrong holder
- tune the virtual working area with a wrong holder
- tune the component of real working area in wrong position
ES1.2: Ordinary maintenance

- Mistake the procedure
- Unable to solve the problem

**PLANNED TEST PARTICIPANTS**

Current estimate: 20 test participants.

EDUCATION: ~7 subjects with high school diploma, ~13 with elementary diploma

IMPAIRMENTS: efforts will be made to enrol physically and/or cognitively impaired subjects, but, for the time being, no estimate can be given on their number

COMPUTER SKILLS: ~6 subjects with low computer skills and ~14 with normal computer skills

LEVEL OF EXPERIENCE: ~5 subjects with < 1 year of work experience, ~11 with 1-3 years of work experience, ~4 with > 5 years of work experience

AGE: ~10 subjects < 30 years old, ~10 > 30 years old

**3.2 Use case 2: robotic cell of GIZELIS applied to SILVERLINE process**

**EVALUATION SCENARIOS**

ES2.1: Given a raw piece to bend, users will be in charge of simulation, tool selection, and parameters adjustment

ES2.2: Changing photo sensor

**FREQUENT ERRORS WITH TRADITIONAL INTERACTION SYSTEMS**

ES2.1: Tuning the simulation

- Tune the robot to pick the part from the gripping point of the first bend
- Tune the robot to palletize correctly with enough distance between the parts and without taking into account the gravity (curved parts might fall when placed wrong due to their centre of gravity)

ES2.1: Tuning the tools

- Tune the robot with different tool than the one is simulated with

ES2.1: Tuning the parameters

- Tune different angles than the ones in real part (possible collision)

ES2.2: Tuning the parameters

- Operator must have high knowledge of the system, read the current instruction in program and try to locate from i/o’s the faulty sensor. The systems most of the times is idle waiting for a technician to come

---

1 While evaluating the prototypal INCLUSIVE system among various user groups, it will be taken into consideration that one person may belong to different groups, e.g.: an older employee with a cognitive disability. Statistical analysis of measurement results (with and without INCLUSIVE system) will take this into account. This applies to all the three use cases.
PLANNED TEST PARTICIPANTS

Current estimate: 34 test participants.

EDUCATION: ~21 subjects with high school diploma, ~13 with elementary diploma

IMPAIRMENTS: ~10 deaf (3 with 3rd level disability, 7 with 2nd level), ~4 with upper limb disabilities (3rd level disability), ~20 non-impaired subjects

COMPUTER SKILLS: ~18 subjects with low computer skills and ~16 with normal computer skills

LEVEL OF EXPERIENCE: ~10 subjects with < 1 year of work experience, ~11 with 1-3 years of work experience, ~13 with > 5 years of work experience

AGE: ~16 subjects < 30 years old, ~18 > 30 years old

3.3 Use case 3: production line by KHS

EVALUATION SCENARIOS

ES3.1: Changeover of the format

ES3.2: Fault recovery procedure for neck ring misalignment

FREQUENT ERRORS WITH TRADITIONAL INTERACTION SYSTEMS

ES3.1: Changeover of the format

- Choosing the wrong format parts (measurement: distinct identification mark of format parts)

ES3.2: Fault recovery procedure for neck ring misalignment

- Wrong setting/position of gripper cylinder possible
- Wrong setting/position of label box referred to segments (label does not get enough glue)
- Wrong setting/position of brushes (label could be misaligned by brushing)
- Wrong setting of glue pump (too much glue on gripper cylinder, thus the label “slips” on bottle)
- A combination of the above mentioned errors is also possible

PLANNED TEST PARTICIPANTS

Current estimate: 10 test participants.

IMPAIRMENTS: ~5 subjects with varying physical and cognitive impairments, ~5 non-impaired subjects

LEVEL OF EXPERIENCE: ~5 subjects with < 3 years of work experience, ~5 with > 3 years of work experience

AGE: ~5 subjects < 30 years old, ~5 > 30 years old
4 Conclusion

This deliverable completes the activities performed in WP1, which have focused on the definition of the requirements and architecture of the INCLUSIVE system. Specifically, in this document a plan is provided for the tests that will be carried out to assess the effectiveness of the INCLUSIVE system and its performance with respect to the requirements and intended goals.

Two groups of tests have been defined. First, in-lab tests will verify that each part of the INCLUSIVE system works correctly. In particular, the measure, adapt and teach modules and the middleware will be tested. They have been described in details in Deliverables D1.3 and D5.1. Second, at the stage of system integration, the whole INCLUSIVE system will be tested on prototypes developed in WP7 on the basis of the three use cases considered in the projects. In these tests, users from the target groups of vulnerable subjects defined in Deliverable D1.1 will be enrolled to assess whether the methodologies devised in the project provide a significant advantage in the interaction of operators with complex automatic or robotic systems.