

1. Functional Requirements

1.1. Conduct a Game

The first and most important major requirement is to be able to actually conduct the strategic awareness game. This means participants must be able to join and play a game together, while a trainer observes and facilitates the game.

1.1.1. Participate in a Game

Requirements 1 and 2 are self-explanatory – players must be able to play games. If a multi-player game is implemented, it is likely trivial to implement a single-player version.

| ID | Requirement | Priority |
|----------------|--|-----------|
| 1 Group Games | The system MUST ALLOW <i>participants</i> to collaboratively ¹ play a game as part of a group. | Essential |
| 2 Single Games | The system MAY ALLOW <i>participants</i> to play a game alone. | Medium |

Personalized software systems are perceived as having a higher quality. Furthermore, while in a collaborative game every participant discusses and solves the same inject, the participants may have different prior knowledge. Providing personalized hints based on the target group of a participant, may lead to a more productive discussion, rather than explaining key concepts to single participants.

| ID | Requirement | Priority |
|-----------------|---|----------|
| 3 Target Groups | The system SHOULD ALLOW <i>participants</i> to describe the target group they belong to, using keywords. Keywords can be used to describe: a) - The industry of the participant, b) - The prior experience of the participant, c) - The position of the participant within their organization. | Medium |
| 4 Hints | IF the <i>participants</i> have used keywords to describe themselves, AND these keywords indicate that the participants are not familiar with some concepts, THEN The system SHOULD SHOW hints with additional information to explain these concepts to the participants. | Medium |

Naturally, in order to play a game, participants must solve injects. Because this is a collaborative game, all participants must see the same inject.

| ID | Requirement | Priority |
|-------------------|--|----------------------------------|
| 5 Solving Injects | The system MUST ALLOW <i>participants</i> to solve injects. An inject solution may be: <ul style="list-style-type: none">• Advancing an informative inject.• Selecting one choice of multiple choices.• Inserting a textual value. | Essential Essential Medium |
| 6 Inject Sequence | The system MUST SHOW the same sequence of injects to all participants of a game. a) IF a group of <i>participants</i> selects different choices when solving one inject, THEN the system MUST determine which choice is evaluated to advance the game. | Essential Essential Medium |

¹ Collaborative game play: All participants have the same goal. In practice, they play through exactly the same scenario.

| | | |
|----|---|--|
| b) | The system MAY ALLOW <i>participants</i> to revise their choice, before determining which choice to evaluate. | |
|----|---|--|

Variables may hold a value over the course of multiple injects. Seeing the value of these variables may influence some decisions (e.g. knowing that there is still a high budget available, participants might choose a different path as opposed to having a low budget).

| ID | Requirement | Priority |
|---------------------|--|----------|
| 7 Visible Variables | IF the game has variables AND some of these variables are visible to the participants THEN The system MUST ALLOW <i>participants</i> to view these variables. | Medium |

In the spirit of collaborative gameplay, participants should be able to rejoin, if they have left the game or disconnected from the game for whatever reason. This should not be possible for closed games, however.

| ID | Requirement | Priority |
|-----------------|--|-----------|
| 8 Rejoin a Game | IF a <i>participant</i> was previously disconnected from a game, THEN The system MUST ALLOW players to rejoin. | Essential |
| 9 Closed Games | The system SHOULD PROHIBIT <i>participants</i> from joining closed games. | Medium |

1.1.2. Facilitate a Game

For trainers to facilitate a game, they have to be able to open a game first, thus making it available for participants. Trainers might also want to manage the participants that have joined an open game, such as limiting the number of possible participants or removing select participants.

Naturally, trainers also have to be able to begin a game, so that all participants can play through.

| ID | Requirement | Priority |
|------------------------|--|-----------|
| 10 Opening Games | The system MUST ALLOW <i>trainers</i> to open games. To open a game, trainers MUST SELECT a scenario that is to be played. | Essential |
| 11 Invite trainers | The system SHOULD ALLOW <i>trainers</i> to allow other trainers to facilitate the game. All trainers of a game MUST HAVE the same permissions. | Low |
| 12 View Participants | The system SHOULD ALLOW <i>trainers</i> to see how many participants have joined an open game. | Medium |
| 13 Remove Participants | The system MAY ALLOW <i>trainers</i> to remove any number of participants from an open game. | Low |
| 14 Private Games | The system MAY ALLOW <i>trainers</i> to set games to “private”, such that only certain participants can join. Examples of how this MAY be achieved are: <ul style="list-style-type: none"> Defining a PIN-code or passphrase, which participants have to enter to be able to join a game. Requiring authentication from participants. | Low |
| 15 Participant Limit | The system MAY ALLOW <i>trainers</i> to define an upper limit to the number of participants that can join an open game. | Low |
| 16 Begin Games | The system MUST ALLOW <i>trainers</i> to begin games. | Essential |

To properly facilitate a game, such as by providing guidance to participants who are struggling, trainers must have an overview of the state of the game. Depending on the situation, trainers may choose to adapt the difficulty by changing game variables or activating/deactivating injects. Trainers should also have the option to abort games, in case the training session is interrupted or the game difficulty is inappropriate to the target audience.

| ID | Requirement | Priority |
|---|---|-----------|
| 17 Game Information a) b) c) d) | The system MUST ALLOW one or more <i>trainers</i> to see the relevant information of a game in progress: - Which inject participants are currently working on. - Which variables exist in the game. - What value each of the variables has. - Any other injects and stories. | Essential |
| 18 Observers | The system SHOULD ALLOW one or more <i>observers</i> to see the relevant information for a game in progress. This is the same information that a <i>trainer</i> can see. | Medium |
| 19 Change Game Variables | The system SHOULD ALLOW <i>trainers</i> to change variable values for games in progress. | Medium |
| 20 Deactivate/activate injects | The system MAY ALLOW <i>trainers</i> to toggle whether injects are shown during a game. | Low |
| 21 Abort Games | The system SHOULD ALLOW <i>trainers</i> to abort games. Aborted games are <i>closed</i> . | Medium |

1.2. Develop a Scenario

To play a game, learning scenarios must first be created. For convenience and long-term usability, it should also be possible to edit and delete scenarios. The requirements below reflect the different elements that a learning scenario has – they correspond to the priority for conducting a game as described in Section 1.1.

| ID | Requirement | Priority |
|--|---|-------------------------------------|
| 22 Create scenarios | The system MUST ALLOW <i>scenario designers</i> to create learning scenarios. | Essential |
| 23 Mandatory Scenario Properties a) b) c) | The system MUST ENFORCE that a scenario has all required properties: - Title, - Description, - One or more Stories. | Essential Essential Essential |
| 24 Optional Scenario Properties a) b) c) | The system SHOULD ALLOW that a scenario has the following properties: - Scenario Variables, - Target Group, - Learning Objectives. | Medium Medium Low |
| 25 Scenario Variables a) b) c) | The system SHOULD ALLOW <i>scenario designers</i> to define <i>variables</i> for the scenario. IF so, The system MUST ENFORCE that variables have all required information: - Name, - Datatype (NUMERIC, TEXTUAL), - Starting Value at the beginning of a game. | Medium |
| 26 Stories | The system MUST ENFORCE that a scenario has one or more <i>stories</i> . | Essential |

| | | |
|--|---|---|
| 27 Story Order | The system MUST ENFORCE that the stories of a scenario are in sequential order. | Essential |
| 28 Story Properties a) b) | The system MUST ENFORCE that a story has the following properties: - Title, - One or more injects. | Essential |
| 29 Injects | The system MUST ALLOW <i>scenario designers</i> to add one or more injects to a story. | Essential |
| 30 Entry Points | The system MUST ENFORCE that a story has exactly one inject as an <i>entry point</i> . | Essential |
| 31 Inject Properties a) b) c) d) e) f) | The system MUST ENFORCE that an inject has the following properties: - Title, - Textual Description, - Zero or one reference to - Any number of transitions, - An image, - Hints for different target groups. | Essential Essential Essential Essential Medium Low |
| 32 Transition | The system MUST ALLOW <i>scenario designers</i> to define transitions from one inject to another. | Essential |
| 33 Validate Storyline | The system MUST ENFORCE that all injects within a story can be reached from at least one other inject via transitions. ² | Essential |
| 34 Mandatory Transition Properties | The system MUST ENFORCE that a transition has the following properties: - Title (also called “label”), - Target inject. | Essential |
| 35 Optional Transition Properties | The system SHOULD ALLOW that a transition has the following properties: - VariableChange, - Conditions. | Medium |
| 36 Variable Changes | The system MUST ALLOW <i>scenario designers</i> to define that a transition will change the variable values of the game. | Medium |
| 37 Transition Conditions a) b) c) d) e) | The system MAY ALLOW a transition to reference another inject, IF a predefined condition is met. IF conditions are allowed, THEN The system MUST ENFORCE that they have the following properties: - Scenario variable, - Comparison Operator, - Comparison Value (threshold), - Alternative Inject. IF conditions are allowed, THEN The system MAY ALLOW chaining multiple conditions through logical operators. | Low |
| 38 Edit Scenarios a) b) c) | The system SHOULD ALLOW <i>scenario designers</i> to edit scenarios. Editing includes the following: - Add a new story to the scenario, - Remove an existing story from the scenario, - Change the order of stories in the scenario, | Medium |

² This implies that there can be no livelocks (circular dependencies) and no unreachable (dangling) injects.

| | | |
|----------------------------|--|--------|
| d) e) f) g) h) | <ul style="list-style-type: none"> - Add an inject to one of the stories, - Remove an inject from one of the stories, - Add a transition from one inject to another, - Remove a transition from one inject to another, - Change any of the values for the core properties mentioned in the previous requirements. | |
| i) | IF a scenario is being edited, THEN the system MUST ENSURE that other requirements to the system are still satisfied. | |
| 39 Delete Scenarios | The system SHOULD ALLOW <i>scenario designers</i> to delete of scenarios. | Medium |
| 40 Selective Deletion | The system MIGHT NOT ALLOW <i>scenario designers</i> the selective deletion of previous versions of a scenario. | Low |

In the future, participant groups may come from different nationalities – catering to the different native languages of participants might improve the learning effect.

| ID | Requirement | Priority |
|---------------------------|---|----------|
| 41 Multilingual Scenarios | The system MIGHT NOT ALLOW <i>scenario designers</i> to offer the same scenario in different languages. | Low |

1.3. Provide Feedback

Rapid feedback allows participants to understand the consequences of the decisions made during the course of the game. This is a major factor for learning. Feedback is also a major factor for motivation theory, particularly flow theory and should therefore be incorporated as much as possible during the game.

Feedback can be provided at the end of the game, so that all participants and the trainer can collectively evaluate the impact of single decisions.

| ID | Requirement | Priority |
|--|--|----------|
| 42 View game history (participant) a) b) c) | <p>The system SHOULD SHOW <i>participants</i> the personal history of a previously played game.</p> <p>The personal history of a game consists of:</p> <ul style="list-style-type: none"> - All of the solutions for injects which this participant has submitted in the course of this game. - Timestamps for the aforementioned solutions. - Timestamps denoting the start and end time of the game. | Medium |
| 43 View game history (trainer) a) b) c) d) | <p>The system SHOULD SHOW <i>trainers</i> the full history of a game which they have previously closed.</p> <p>The full history of a game consists of:</p> <ul style="list-style-type: none"> - A count of the number of players of this game. - All of the solutions for injects which have been submitted in the course of this game. - Timestamps for the aforementioned solutions. - Timestamps denoting the start and end time of the game. | Low |
| 44 View game history (observer) | IF an <i>observer</i> has observed a game in the past, THEN The system SHOULD ALLOW the observer to view the general history of this games any time in the future. | Low |

Scenario designers may also want to know how a scenario is being used – for example, whether a disproportionate number of games of this scenarios are aborted, whether they take an exceedingly long amount of time or whether some choices are chosen more often than others. Because changes to a scenario may also impact the statistics for the scenario, it should be possible to correlate statistics to a specific version of each scenario.

| ID | Requirement | Priority |
|--|---|----------|
| 45 View scenario statistics (observer) | IF a <i>scenario designer</i> has created a scenario in the past, THEN The system SHOULD ALLOW the <i>scenario designer</i> to view the usage statistics of this scenario. These statistics include: <ul style="list-style-type: none"> a) - How often a game of this scenario has been played, b) - How the target group for each game looked like, c) - Which paths were typically taken in each game, d) - How these statistics were impacted by changes to the scenario. | Low |
| 46 Version history | IF a scenario has been edited by a <i>scenario designer</i> , THEN The system MUST allow backwards compatibility of usage statistics. ³ | Low |
| 47 Deleting history | IF a scenario has been deleted by a <i>scenario designer</i> , THEN this will also delete all games and statistics that are associated with this scenario. | Medium |

Finally, it might be worthwhile to provide feedback immediately after each inject. This has low priority, however, because a similar effect can be achieved by adding dedicated injects.

| ID | Requirement | Priority |
|--------------------|--|----------|
| 48 Inject feedback | IF an inject has been solved by a <i>participant</i> , THEN The system MAY SHOW immediate feedback on the solution. | Low |
| 49 Group feedback | IF a game is a group game AND and inject has been solved by a <i>participant</i> , THEN The system MAY SHOW feedback on how the rest of the group has solved this inject. | Low |

1.4. Other functional requirements

To follow the principles of separation of concerns and improve privacy, it might be prudent to allow users to create user accounts. These would then have to administrated somehow – organizations might decide to use a dedicated administrator role to do so.

| ID | Requirement | Priority |
|-----------------------|---|----------|
| 50 Create own account | The system SHOULD ALLOW <i>all users</i> the creation of user accounts. | Low |
| 51 Delete own account | The system SHOULD ALLOW each <i>user</i> to delete their own account. | Low |
| 52 User roles | The system SHOULD ALLOW a <i>user</i> account to have multiple roles. | Low |

³ When a scenario is modified, this may impact future statistics. Therefore there must be a way to make transparent which change in statistics was brought about by which modification.

| | | |
|---------------------------------|--|-----|
| a) b) c) d) | Available roles include: - Participant, - Trainer, - Scenario Designer, - Administrator. | |
| 53 Add roles (administrator) | The system SHOULD ALLOW <i>administrators</i> to add roles from user accounts. | Low |
| 54 Remove roles (administrator) | The system SHOULD ALLOW <i>administrators</i> to remove roles from user accounts. | Low |

2. Non-Functional Requirements

2.1. FURPS+ Framework

For the definition of the non-functional requirements of the system, the FURPS+ framework was used. The original FURPS framework was developed by Robert Grady at Hewlett-Packard.⁴

The FURPS+ framework is an acronym for:

- Functionality,
- Usability,
- Reliability,
- Performance,
- Supportability,
- And additional requirements.

Requirements for each of these categories will be captured in the following subsections, with the exception of “functionality”, because this was extensively covered in the previous chapter.

A more recent take on the FURPS+ framework has been provided by Peter Eeles from IBM⁵, which has inspired some of the non-functional requirements described below.

2.2. Usability

Usability refers to how easily a user can perform their respective user tasks. User tasks are hereafter understood to be any sequence of activities necessary to satisfying one of the functional requirements.

| ID | Requirement | Priority |
|-----------------------------------|---|-----------|
| 55 GUI | A <i>user</i> MUST be able to perform all user tasks through a <i>graphical interface</i> . | Essential |
| 56 Integration with other systems | All tasks of a <i>scenario designer</i> MAY be performed by other systems through an API. | Low |
| 57 No account required | <i>Participants</i> SHOULD NOT be required to have a user account to be able to participate in a game. Examples of how this MAY be achieved are: <ul style="list-style-type: none"> • Access to any game via a simple URL. | Essential |

⁴ Grady, Robert; Caswell, Deborah (1987): *Software Metrics: Establishing a Company-wide Program*. Prentice Hall.

⁵ Eeles, Peter (2005): Capturing Architectural Requirements. Retrieved from <https://web.archive.org/web/20201112020231/http://www.ibm.com/developerworks/rational/library/4706.html>. Last access on 2021-06-08.

A system must also be accessed somehow. Because the system is intended to be used for training a heterogeneous target group, participants may use a variety of devices to participate in a game. Interoperability with different devices is therefore a major concern.

| ID | Requirement | Priority |
|----------------------------------|--|-----------|
| 58 Browser based a) b) | <p>A <i>user</i> MUST be able to access the graphical interface of the system through any commonly used state of the art web browser.</p> <p>Commonly used web browsers are:</p> <ul style="list-style-type: none"> - Google Chrome, - Safari.⁶ <p>A browser is state of the art, if it is no older than two generations than the youngest version of that browser.</p> | Essential |
| 59 Other browsers | A <i>user</i> MAY be able to access the graphical interface of the system through any other web browser. | Low |
| 60 Hardware Requirements | <p>The system MUST be usable with standard periphery of a modern desktop computer.</p> <p>Standard periphery is understood to be mouse and keyboard.</p> | Medium |
| 61 Mobile use | <p>The system SHOULD be usable with mobile devices, such as smartphones and tables.</p> <p>This can be achieved using one of various ways, such as:</p> <ul style="list-style-type: none"> - responsive web design, - a mobile app. | Medium |
| 62 IoT Accessibility | The system MAY be usable with other cyberphysical systems that support a web browser and user input. | None |

Usability also requires the user to understand the system. This goes for an understanding of the system language, as well as understanding the possible user tasks.

| ID | Requirement | Priority |
|--------------------------|---|----------|
| 63 System language | The language for the graphical interface language MUST be English. | Medium |
| 64 Inter-nationalization | The graphical interface MAY support other languages (internationalization). | Low |
| 65 Easy Onboarding | <p>The system SHOULD provide support to first-time users, to accelerate the onboarding-process.</p> <p>Examples of how this MAY be achieved are:</p> <ul style="list-style-type: none"> • Tooltips that explain the different elements of the user interface. • A tutorial, which guides the user through a first-time use of their current user task (e.g. creating a scenario or joining a game). | Medium |

⁶Google Chrome: global market share of ca. 64%. Safari: global market share of ca. 19%.

Source: Liu, Shanhong (2021): Global Market Share Held by Internet Browsers 2012-2021, by month. Retrieved from <https://www.statista.com/statistics/268254/market-share-of-internet-browsers-worldwide-since-2009/>. Last access on 2021-06-08.

The system should also enforce an adequate amount of security, without inflicting the usability of the system. This includes security of user accounts, as well as privacy, integrity and confidentiality of data (the “availability” aspect of the classic CIA – confidentiality, integrity and availability – triad is listed in section 2.3 Reliability). A logical result of the CIA requirements is that it should be possible to restrict read- and write-access to scenarios and to limit read-access for statistics. At the same time, members of a team or organization might want to collaborate on scenarios and therefore need access to the same scenarios and statistics.

| ID | Requirement | Priority |
|---------------------------------------|--|----------------|
| 66 Account Security a) b) c) | The system SHOULD ENFORCE each account to have adequate authentication measures, such as, but not limited to: - Password strength, - 2-factor-authentication, - Biometrical authentication. | Low |
| 67 Privacy | The system SHOULD NOT ALLOW <i>users</i> to view the personal details of other users. | Medium |
| 68 Confidentiality | The system SHOULD NOT ALLOW <i>users</i> to access the statistics and history of games and scenarios that other players have played or created. | Medium |
| 69 Integrity | The system SHOULD NOT ALLOW <i>users</i> to manipulate game histories or scenario statistics. | Low |
| 70 Access Control Groups | The system SHOULD ALLOW to create, manage and delete groups of users. The system SHOULD ALLOW defining whether these groups have read- and write permissions for scenarios. | Low Low |

2.3. Reliability

The system should be robust enough to allow productive use.

| ID | Requirement | Priority |
|-----------------------------------|---|----------|
| 71 Concurrent Use | The system SHOULD allow multiple games to play through the same scenario concurrently. | Medium |
| 72 Number of users | The system SHOULD allow up to 200 users (participants and trainers) for a game. | Medium |
| 73 All-time Availability | The system SHOULD be available at least 95% of the time. | Medium |
| 74 Availability during core times | The system SHOULD be available 99% of the time during core working hours. Core working hours are understood to be between 9am and 5pm from Monday to Friday. | Medium |
| 75 Robustness | IF one user task becomes defective, THEN a user SHOULD still be able to perform all other user tasks. | Medium |

2.4. Performance

Load time is a key factor in determining how satisfied users are with a web application.⁷ At the same time, the system should have enough capacity for injects and scenarios to support prolonged productive use. Storage requirements are of limited interest, however, because they can nowadays be scaled up with comparative ease.

| ID | Requirement | Priority |
|----------------------|--|----------|
| 76 Loading Time | The system SHOULD have a low loading time. As a benchmark, the loading time of a question on menti.com will be considered. | Medium |
| 77 Scenario Capacity | The system SHOULD support persisting up to 100 scenarios. | Medium |
| 78 Inject Capacity | A scenario SHOULD support up to 200 injects. | Medium |

2.5. Supportability

A software is not complete after being released to production. Instead, special attention should be taken to ensure that it can be maintained, supported and further improved over time. This implies that the architecture should be well-documented and allow some level of flexibility, the code should be understandable and well-written and the tests and requirements must be understandable, even for third-party developers.

| ID | Requirement | Priority |
|------------------------|--|-----------|
| 79 Code Quality | The source code for the system MUST be written according to relevant best practices, to ease code handover and maintenance by third parties. Relevant best practices are, among others: <ul style="list-style-type: none"> - Consistent writing style, - Few branches within the code, - Avoid duplicate code, - Keep methods shorter than 20 lines of code (LOC), - Avoiding excessive use of comments. | Medium |
| 80 Documentation | The architecture for the system MUST be comprehensively documented. This includes at least: <ul style="list-style-type: none"> a) One or more class diagrams which show the key elements of the domain layer. b) One diagram which shows the basic package architecture of the system (layers, MVC structure, etc) c) One diagram (BPMN or UML sequence or UML activity) per use case, which provides a dynamic view on how this use case will be executed at runtime. d) A textual summary of key architectural decisions and domain decisions that were taken over the course of the implementation. | Essential |
| 81 Ubiquitous Language | The key concepts (e.g. scenario, inject, story) used in the domain model of the system SHOULD be described in this document. | Medium |

⁷ Coe, Mary Ellen (2019): Milliseconds earn millions: why mobile speed can slow or grow your business. Retrieved from <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-site-speed-importance/>. Last access on 2021-06-15.

| | | |
|---------------------------------------|---|-----------|
| 82 Extensibility | The system MUST ALLOW the definition of new sorts of injects without breaking any existing functionality. | Medium |
| 83 Test Coverage | The system MUST be tested such that each line of code has been tested at least once. | Essential |
| 84 Behavior- Driven Development | Tests SHOULD follow the paradigm of behavior-driven design (BDD), such that feature tests can be understood without knowledge of the underlying code base. | Medium |