RE Mini Project

V150914SFR

Scenario:

You get appointed to do requirements engineering for the system X. While the system X is interesting overall, it is unattractive or burdensome for at least some users. You take initiative and propose an app that simplifies the system from the users' perspective or enhances the value of the system X for these users. You do requirements engineering to understand the users' problem in detail, to suggest the app that improves the system X for the users, and build evidence that your app actually will deliver the improvements.

Note that a user will always evaluate an application in comparison to all alternatives the user is aware of. Thus, ensure that your app is better than any of these alternatives.

Note that you will be a bad representative of a user. You are too much involved in the project, thus are clearly biased. To ensure success, involve real people as user representatives that are not students or lecturers at FHNW.

Goals:

- Be able to elicit requirements and specify an attractive, realistic system with state-of-the-art techniques.

Tasks:

1. W1: Build a group of five students. Exceptions have to be granted by Samuel.
2. W2: Selection of system X accepted by Samuel.
3. W3: Picture- and Text-based documentation of the system X and of alternative systems that a user may use as a replacement of system X.
4. W4: Stakeholder documentation and list of ideas of how the system X and its alternatives may be improved from the users' perspective.
5. W5: Video-based documentation of how the system X is used by the users and the problems users experience with it.
6. W7: Prototype of the proposed app.
7. W10: Functional specification of the proposed app with appropriate diagrams. The use cases must be specified in detail as they will be used for the ensuing validation.
8. W11: Video-based documentation of how the proposed app will be used by the users and the feedback about advantages, disadvantages, and risks that the users gave.
9. W12 (obs: large effort!): Adaptations of the requirements specification to account for the user feedback.
10. W13: Quality checking of the requirements document. Ask other teams to help you in the quality checking and offer the same to the other teams. Deliver here a list of findings that represent the most critical items that remain to be improved in your specification document.
11. W14: Submission of the requirements document and of supporting materials (see deliverables).

Constraints for the selection of the system X:

- Select an existing system that may be replaced or enhanced by a mobile app. The system may be anything. It may be an existing app, a desktop software, a cloud software, a hardware-based system such as a ticket mache, a system that does not contain any software such as a bycicle, etc.
- The mobile app must be simplifying the system from a user's perspective or increase the value of the system for the user (i.e. make the product smaller, while solving a bigger problem). Simple addition of functionality is uninteresting and not allowed.

- The system should be relevant for people like you and your colleagues (ie. you belong to the users targeted by the system). At least two team members must know the system well, and any colleague student and Samuels must be able to use it.

Deliverables

- Requirements specification document that gives an overview of the complete app. The document…
  - must give an overview of the whole new system that replaces the old system.
  - must specify the simplifying, value-adding feature(s) in detail.
  - must be written with enough explaining text that a new reader understands what the various texts, diagrams, graphics, and photos mean. Test this with colleagues.
- Videos from your requirements discussions with the real users.

Rubrics used for grading. Evaluated is how well you achieved that the specification is realistic, corresponds to good practice, and is understandable.

- Understandability for a new reader: the specification document [builds a clear argument of how the proposed app improves the system's value for the stakeholders | contains enough clarification to understand the various texts, diagrams, graphics, and photos | is clearly a requirements document | contains parts that are difficult to interpret | contains many parts with unclear meaning or role for the proposed app]
- Conformance to document standards: the specification document [represents an improvement to the proposed template | complies with the proposed template | contains all necessary parts | does not comply with any specification document standard | is unstructured or fragmentary].
- Conformance to language standards: the specification [uses well-defined language and formalisms efficiently | uses UML and NL patterns where applicable | is precise, concise, and verifiable | is unprecise at many places | is not understandable]
- Internal consistency: the specification [makes evident how the system fulfils vision and stakeholder needs | contains all necessary traces | allows inferring how the system fulfils vision and stakeholder needs | contains parts that are not integrated with other parts | contains unnecssary redundancies, inconsistencies, or ambiguities]
- Correctness: the specification of existing parts of the system [corresponds fully to the real-world system | corresponds to the real-world system but contains omissions that are well indicated | corresponds partially to the real-world system | contains vague parts that cannot be mapped to the real-world system | mostly does not correspond to the real-world system or cannot judge]
- Attractiveness: the specified innovation (the improvements produced by the app) [is attractive for the users it targets and clearly improves the uers' situation in comparison to existing alternatives | is attractive the users it targets | is a realistic addition | targets users not interested in the system | exists already in competitive systems, is unrealistic, or does not represent an improvement to the system]
- Comment: **target 25 pages for your requirements specification. The number of pages is indicative only, however. Important is that you provide an overview of the complete system and in detail the improvemens to be implemented in the app.**
- Comment: **any lowest-level rubric can lead to a Fail even if the specification is good according to other rubrics**.

The points are [+2 | +1 | 0 | -1 | -2] for each rubric. The rubrics with a high number of points assume that the criteria indicated in the rubrics with less points have been addressed. The average number of points will be translated into a grade as follows:

- Average of 0 points: grade 4.0.
- Average of 1.5 points: grade 6.0.
- Average of -2 points: grade 1.0.
- The averages in-between are proportionally translated into grades.

Note: insufficient contribution of a team member will imply that the concerned team member is required to do the assignment alone. The separated team member will need to deliver a specification for another mobile app.

Note: in case of conflicts between sub-teams, the overall team is split into teams that work alone. Samuel will decide about changing the app that is to be specified by each concerned team.

Notes, tips, tricks:

- A "feature" is a group of requirements that from a user, marketing, or strategic reason should be implemented together.
- Discuss in your team how you distribute the work. Each member takes about 1/4 of the total workload. Ensure that each team member was involved in each task, e.g. by investing the overlapping work percentages in reviews of work results.
- Monitor the time needed to specify X (e.g. a use case). Predict the total effort and scope the specification so that the assignment becomes feasible.