

Report Guidelines for Educational Software IN4145

These guidelines are meant to help you structure and write your final report for educational software IN4145. However take note that these are **only guidelines** and therefore you may structure your report in different way or exclude chapters that are on this list if you feel that they do not fit in your report. You may also add additional chapters that are not on this list.

It is not needed to follow a strict design method (for example HOEP) as the project you will be working is relatively small and most design methods are meant for a large interdisciplinary development team. Also, it is not needed to construct class diagrams or other strict technical specification of the product that you want to create; this is not even possible if you choose to work with specific software packages (e.g. Camtasia).

The size of the report should be around 20 pages.

Structure of the report:

Title Page

- Project title
- Group number / Group member info
- Student numbers
- Datum
- Vak code

Introduction

- Motivation
- Problem definition
 - Macro level:
 - Well defined topic
 - Argumentation why educational software is a valid method of teaching in this case
 - Stakeholder (employers)
 - Do not name solutions here yet! (e.g. “this could be done with web lectures...”)
 - Constraints (hardware, time, costs etc)
 - Example: “Design educational software that would motivate and teach children aged between 4 and 8 in a playful way to talk English.”

Target group

- Definition
- Characteristics
- Possible problems / limitations
- Target group expert available?
- Other stakeholders (e.g. teachers, technical support...)

Learning material

- Meso level:
 - List
 - Specification
 - Coherence between different subjects
 - Diagrams showing relations
 - Teaching methods practiced currently
 - Teaching methods that will be used in the software

Requirements

- List of requirements that can be extracted from the information in the previous 3 chapters
 - Target group
 - Stakeholders
 - Technical
 - Educative
 - Other

Design

- Underlying educational principles
- Scenarios/Storyboards/Use cases
- (select the way that works best for you)
- Micro level:
 - Task Analysis
- Diagrams where needed
 - Metaphors (argument!)
 - Modalities (argument!)
 - Way of presenting the learning material
 - Interaction (specify!)
 - Navigation (specify!)
 - Assignments
 - Games
 - Convenience
 - Ease of use
 - Evaluation of progress (by both teacher and student)
 - Screen mock-ups / sketches for the above (where needed)
 - Specify or argument all points!
 - If your software includes a game / assignment as a separate section describe these in great detail – you can again subdivide it in task analysis, metaphor, modalities, interaction, navigation and evaluation.

Implementation

- Platform (hardware technology)
- Tools used
- Argument choices
- Positive / negative aspects of tools used

- Describe Implementation Process:
- Problems:
- How did you deal with them?
- Changes to design?
- Things left out?
- Other solutions?
- Solutions:
- Tell about use of techniques / technologies / libraries / existing software / etc
- Lessons learned
 - About implementation
 - About educational software

Evaluation

- Method
 - Experiment setup
 - Participants
 - Measure
 - Procedures
- Data analysis and Interpretation
- Conclusion / Comments / Suggestions

Conclusion

- Main conclusions
- Limitations
- Suggestions for further development

References

Appendix: Manual

Extra:

Please include a cd-rom with the educational software that you have developed (if applicable).

Conclusion

References / Supplements