Recommended structure of the report

- Chap.1: Introduction and motivation:
  - **"Setting the scene"**
  - Should be enough to understand the relevance and importance of your topic
  - What are the challenges or problems you are trying to solve?
  - Problem definition / research question(s), delimitation
  - Methodology (not too long, if you have a lot to present, do so in a separate chapter)
  - Expected outcome
- (Extended) state-of-the-art related to your problem area:
  - **Descriptive, neutral, objective (Not your own work!)**
  - Presentation of the specific application domain or problem area, including existing solutions that are related to your project: What can we build upon?
  - Research challenges (academic perspective)
  - Relevant theory, including course material
  - Candidate technologies (NOT tutorials, but highlighting the relevant features and characteristics for your problem)
  - Relevant standardization activities or initiatives
- The analysis may comprise several parts:
  - **Subjective, your own work!**
  - What is need to find a solution to your problem:
    - Literature study (desktop research)
    - Comparison of different technologies
    - Simulations
    - Interviews with users and experts
    - Questionnaires, focus group sessions, test of early ideas and mock-ups, …
    - Identifying different types of requirements
  - Discussion and reflection!
- The Analysis chapter often results in a list of requirements:
  - **VERY important to make a good requirement specification!**
  - For each requirement you must give the rationale behind it:
    - It should originate from your analysis, not just be something random that you think is important
  - Prioritize the requirements, e.g. MoSCoW or similar, and motivate the prioritization
  - It may often be relevant to distinguish between requirement specifications for **a "full-scale" solution** (which you don't have the resources to develop) and your **more limited prototype**
- For service development projects:
  - Describe the development methodology you have used for your solution/ prototype (not the same as the overall project methodology of Chapter 1!)
  - Document the code and include screen shots of your prototype
- Design and implementation chapter(s):
  - Include all relevant diagrams, design choices, etc.
  - Emphasize your own contributions, what is new/original, to demonstrate your engineering skills(!)
- Conclusion and future work:
  - Finish the report by "closing the loop" back to the beginning:
    - Did you solve the problem / answer the research questions?
    - To what extent?
  - What are your main contributions? Make them explicit in the conclusion!
  - Position your work in relation to a "full-blown" solution, you only made a small part of that
  - Evaluation:
    - What did you learn? Would you do the project differently another time?
  - What are your recommendations for future work?