

Assignment 1: Crowd Balancing Platform (preliminary)

By working as a team of architects, you and your team are responsible for the design of the architecture for the Crowd Balancing Platform (see the case description for more details).

Your software architecture design will be delivered in two assignments, each detailed in a separate assignment description.

Throughout the project, you will gradually construct a complete architecture description. To aid you in structuring your work, and treating your work as a living document throughout the project, Canvas contains a documentation template based on the Views and Beyond documentation approach from Bass et al., Chapter 18. The template provides an outline of sections that would typically be included in a (complete) architecture description. You are welcome, but in no way obliged, to use this template. Don't feel restricted by the template's outline; you're always free to (re)structure your incremental assignments as you see fit.

This assignment refers to the case “Crowd Balancing Platform”. Your document will be constructed according to the steps described below and reporting on the key characteristics of your software architecture design.

- **Step 1, Stakeholder profiles:** [see material Week 1, Lecture 1] [Doc. Template Section 2]
First, select at least as many stakeholders as the number of members in your team. Describe their stakeholder profiles. These profiles will drive your own architecture design. Since there are no actual stakeholders in this project that you can talk to, we suggest that each team member adopts at least one of these profiles, and takes on the responsibility to understand that stakeholder and represent their interests during the project.
The selected stakeholders need to cover a diverse set of interests (usability, privacy, etc.) from the stakeholders is needed. Aim to cover as much different perspectives on the new system as possible. In the ‘architecture debate’ sessions, you will be able to challenge your own stakeholder profiles against the ones of the other teams of architects.
Ensure to include for each stakeholder role: the role name (or job title), a description of the stakeholder/role, and the stakeholder’s expectations or demands regarding the system (the requirements or concerns). The second half of the profiles in the documentation template (business goals, architecturally significant requirements) will be addressed in step 3.
- **Step 2, Architecture sketch:** [see material Week 1, Lecture 1] [Doc. Template Section 3]
Then, draw a sketch of the architecture based on your current insights into the selected stakeholder roles, the case description, your experience with earlier architectures that worked out well, etcetera.
In the ‘architecture debate’ sessions, the format in which you present the architecture sketch is up to you - text, a graphical diagram/view, or any other format you think brings the message across.
This architecture sketch can be a starting point for future architectural decisions. As a minimum, provide one diagram showing the overall structure of your solution. Explain what the constituents of the diagram mean, and why you chose this solution.
- **Step 3, Business goals and ASRs:** [see material Week 1, Lecture 2] [Doc. Template Section 2]
In this step you should further detail the business goals of the stakeholders you selected, and the architecturally significant requirements (ASRs) that stem from these goals.

Express the business goals as business goal scenarios, and the ASRs as quality attribute scenarios. Make sure you keep track of the sources of these scenarios, to ease maintenance of the profiles throughout the project.

Throughout the project, you are required to keep the profiles up to date and to extend them when necessary (e.g. when new stakeholders and/or additional goals and requirements emerge). You should also identify and manage conflicts between goals and requirements you identified.

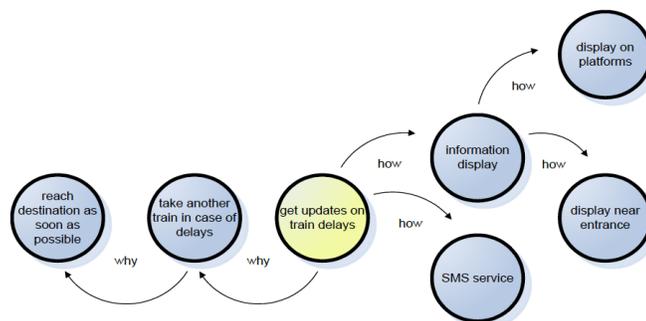
See Chapter 16 (Bass et al.) for more details about business goals scenarios and quality attribute scenarios.

A way to structure your thoughts about a stakeholder role is to think of your requirements in terms of 'why' and 'how'. This is actually a technique that is being used in requirements engineering. With this 'goal-oriented' approach, a requirements engineer can draw a graph that connects low-level requirements to high-level goals, and analyze that graph to determine, for instance, whether there are goals that are not reached, or whether there are any conflicting requirements. For further information on goal-oriented requirements engineering, see for instance the paper *Goal-Oriented Requirements Engineering: A Guided Tour*, by Axel van Lamswearde (available on Canvas), in the Proceedings of the 5th International Symposium on Requirements Engineering, IEEE Computer Society, 2001, pp 1-13.

Suppose you are a stakeholder in the development of a train information system. Let's say you are a traveler (or a representative of the traveler, for instance a customer relationships manager). As a traveler, one of your requirements may be that you are updated if there are any train delays. If you are asked *why* you want this, your response would be somewhere along the lines of "so I can take another train if my train is delayed". The reason for this, yet another 'why', is of course that you want to reach your destination as soon as possible. Schematically, this trace from requirements to higher-level goals can be depicted as follows.

On the other hand, you might also have ideas as to *how* the requirement (i.e. be updated on delays) could be satisfied. When you state this requirement, you may envision information displays on each station. You might even think one step further, and anticipate information displays on each platform, or alternatively displays near the entrance of the station only, or maybe a combination of both. Maybe you were thinking of something entirely different, for instance a SMS messaging service that sends you a SMS whenever your train has been delayed. You may now wonder if this is technically feasible, but in your stakeholder role as a mere traveler you probably wouldn't care.

The figure below schematically depicts those 'why' and 'how' questions. You see that a simple requirement implies all kinds of hidden layers: personal or organizational goals that underlie a particular requirement, but also ideas and visions on the way such a requirement is to be implemented. If the architect teams are not careful to elicit this implicit knowledge, they run the risk to design a system that ultimately does not meet the stakeholders' needs.



- **Step 4, Viewpoint selection:** [see material Week 2, Lecture 3] [Doc. Template Section 4]
Since many stakeholders are involved in the development of a complex system, there is typically not a single way of looking at the architecture that addresses the concerns of all stakeholders. Rather, you need to have different ‘views’ that convey relevant information for different stakeholders. The same is true for the architecture of a building, which may be documented by means of the room layout, the plumbing diagrams, the electrical diagrams, etcetera. Each of those diagrams provides a different view on the same building, and not all views are equally interesting to every stakeholder.
Although, for instance, an electrical diagram will differ from building to building, there is a certain consistency in the way in which such a diagram is constructed, as well as in the concerns that it addresses and the way it is to be used or read. An ‘electrical diagram’ provides a way of looking at a building, which is defined independent of the buildings that one wants to represent with it.
In software architecture, such a specification of the conventions for constructing and using a view is called a ‘viewpoint’. A viewpoint doesn’t give you an actual view of a system (e.g. a ‘floor plan’), but it does tell you how a particular view should be constructed and read (i.e. it specifies which modeling techniques and symbols should be used to represent and connect ‘doors’, ‘windows’ and ‘walls’) and for whom such a view would be interesting (i.e. the stakeholder concerns addressed).
The ISO/IEC/IEEE 42010 standard for architectural descriptions (available on Canvas) provides, apart from a very formal definition of what a viewpoint is, a number of examples of viewpoints. The paper by Koning & van Vliet (also on Canvas) provides a viewpoint selection process.
In this assignment you should select and define a minimum of two conceptually different viewpoints on the system and explain why you have selected those viewpoints. You should ensure that the viewpoints you select cover a broad array of concerns of your stakeholders. The viewpoint definitions should provide insight in the relevant stakeholders and the concerns that are addressed through the viewpoint, the modeling techniques or descriptions used to construct a view that conforms to that viewpoint (the ‘viewpoint language’), and the rationale for your selection of this viewpoint.
Note that the views themselves should not yet be represented (this will be part of your next assignment), but you should not hesitate to start working on your views. This also helps you to assess the quality of your viewpoint definitions.

Deadline for submission on Canvas

Monday November 23, 23:59