Feedback of Release 1 from Muke

Hey Guys, it appears that some of you do not yet quite get what quality requirements are and how constraints are different from other attributes. Here is something that may help.  
  
Quality requirements, in general, are in THREE categories (1) performance requirements (2) other attributes such as usability, security, availability, portability, maintainability, etc, and (3) constraints i.e. limitations imposed by e.g., regulations, budget, business operations,, design limitations, etc  
  
Students usually mistake constraints such as "The app shall only send notifications at 12pm daily" for quality attributes. A constraint in our case is any requirement that cannot be classified as quality attribute or as a functional requirement. For instance, if a business operations is only allowed to develop using Microsoft products, or that any solution implemented cannot go over 1million SEK, etc. Such things (if needed to specify in the requirements document) can only be put as constraints but neither as functional nor as quality attributes. Constraints cover limitations imposed by e.g., physical, environmental, legal, design, circumstances, etc. This is different from attributes such as usability, security, and availability.  
  
Performance requirements cover time and space bounds (timing, speed, volume, throughput)  
  
Notice that to have meaning, these quality attributes, especially performance and other attributes such as usability, availability, etc, need to be written in the context of the functions to which they correspond. Since quality requirements describe how well functions of a system are performed, it is better to attach (especially the critical ones) to some core requirements they correspond to as given in your list of core requirements in the use case diagram. For instance, if one of the use cases is to create a patient, one can define a usability requirement on how well this function must be performed by the user. If one use case is for searching for patients, one can describe related performance requirements to that, etc.  
  
Especially for attributes that can have numbers attached to them e.g., response time, percentage of system availability, etc, use the p-language to specify them. You can use the quality grid as guide to pick those you consider critical and important requirements to use the p-language on or some other technique such as what is given in the lectures or Laussen 2017 template.  
  
Not all your quality requirements must be specified with p-language.  
  
Secondly, someone asked me if "quality requirements" are the same thing as "requirements quality". Clearly, there's a difference. The former refers to non-functional requirements as I indicated above, while the latter refers to how well you specify your requirements. The second part is what you will be looking at in your validation lecture and also when you will be given another group's specification to check.  
  
Lastly, when I was reading through your specification documents, I could tell that different sections were written by different people since the quality of the document varied. For some sections I could easily underst6ans what was written while for others, things were hard to grasp. Please note that you are doing group work. While it is important that each one does their part, try to set a time when all members of the group can review the whole document before submitting so that you submit something you will ALL be happy with. In the end, the grade you get will be a group grade and that will depend on the quality of the document you submit.  
  
All the best as you work towards R2.  
  
Muke