Repetition Questions (dt. Wiederholungsfragen) Lesson 2

Topics and Concepts (Link to Lecture, via Fact Sheets)

The second lesson of the lecture today covered the following topics (see fact sheets in script folder):

1. Methods[1]
2. Agile and Architecture[2]
3. SMART NFRs and NFR templates[3]

In the corresponding exercise[4], we applied the concepts featured in the lecture to elicit SMART NFRs.

Questions

Topic/Concept: Methods

1. What are the building blocks of software engineering and architecture design methods?
2. Which methods were listed in the lecture? List at least three.
3. Name at least two method elements from one of the methods that were introduced.

Topic/Concept: Agile Architecting

1. What is agile software development, and does it make software/application architecture obsolete?
2. How can software architects contribute to being agile and lean (hint: discuss arch. concepts in context of agile principles)?
3. Discuss potential goal conflicts between architects and developers, architects and Scrum masters, architects and product managers.

Topic/Concept: SMART NFRs

1. How was the SMART acronym introduced in the lecture and used in the exercise (what do the five letters stand for)?
2. Name two stakeholders that benefit from SMART NFRs and briefly explain why and how (by discussing one of their concerns each).
3. What do you do if NFRs have not been stated yet, or the existing NFR documentation does not meet the SMART criteria?

Topic/Concept: NFR Templates

1. Which of the SMART criteria does the *response measure* entry in the QAS table address?
2. How does the ASR assessment from lesson and exercise 1 relate to the SEI quality utility tree?
3. What would be a reason not to quantify an NFR/QA?

[1] ../2-lecture-script/lesson2/ZIO-MethodFactSheet.pdf
[2] ../2-lecture-script/lesson2/ZIO-AgileArchitectingFactSheet.pdf
[3] ../2-lecture-script/lesson2/ZIO-SmartNFRElicitationFactSheet.pdf
[4] ../3-exercises-solutions/ZIO-AppArch-ExerciseWeek2.pdf

Answers

Topic/Concept: Methods

1. *Process, notation, techniques, content*
2. *OpenUP, Disciplines Agile Delivery (DAD) that includes Agile Modeling, Attribute-Driven Design (ADD)*
3. *OpenUP has an Inception and an Elaboration phase. ADD has seven steps, for instance step "Identify candidate architectural drivers".*

Topic/Concept: Agile Architecting

1. *See Agile 101 from Agile Alliance for terminology clarification. No, the relationship is not "makes obsolete" but "requires" or "makes possible".*
2. *Write down quality requirements in a simple and accessible way, for instance in the form of user story annotations and quality stories. Focus efforts by rating their significance and prioritizing QAS according to businss value and technical risk.*
3. *The classical conflicts do not go away just because a team is agile: the functionality, time, quality triangle from project management still holds (the architect is primarily concerned with quality, but also with scope control and project execution).*

Topic/Concept: SMART NFRs

1. *Specific, measureable, agreed upon, realistic, time bound (see fact sheet)*
2. *System administrator who wants to ensure that system availability meets user's expectations; system maintainer who wants to be able to fix bugs and implement feature requests quickly without introducing new problems or compromising quality*
3. *Ask for more information, make explicit assumptions (for instance work with categories or numbers from other projects), complete the ones that score highest in the ASR assessment, or define alternate evaluation forms; leaving the NFRs blank (unspeecified) or immature is not an option because it will get you.*

Topic/Concept: NFR Templates

1. *'M'*
2. *The prioritization (n,m) in the third level of the tree matches the two criteria in item 1 of the ASR checklist (business value, technical risk).*
3. *Evaluated in another way, e.g. A/B user testing; when effort exceeds value, see risk discussion in IEEE Software paper by M. Glinz referenced in lecture and fact sheet.*