

Module Title: Software Architecture, Process and Management

Exam Diet (Dec/April/Aug): April 2014

Brief notes on answers:

1. (a) The students have seen the GPES system in the lectures but we have not explored the topic of this question using GPES as an example. I'd expect something like:
 - (i). Modifiability: I'd certainly expect this (1 mark). The justification should be on the basis of the number of stakeholders and the need for continuing modifications through the life of the system (2 marks)
 - (ii). This is more open - possibly another static property like standards compliance or interoperability (justification could be in terms of the existence of a regulator, or in terms of the need for the system to be extensible) but it could also be a more dynamic attribute like availability with an emphasis on mean time to repair since long mttr could have caused operational difficulties (again 1 mark for the attribute and 2 marks for the justification).
- (b) The scenario might look like this:
 - Source: Integration Tester
 - Stimulus: Completion of the development of a new report in the system
 - Artifact: Modules for the full system
 - Environment: Operation/Evolution
 - Response: Results from integration testing to see if all the component queries work together
 - Response Measure: Passing the integration test within one week (including fault finding and repair)

Award 5 marks for a plausible scenario.

- (c) Here I'd expect something like:
 - Limiting the structural complexity of the query so the variability of the required business logic query in each GP system is limited (1 mark for an appropriate tactic, 2 marks for the justification).
 - Adding specialist debugging interfaces to the links to the GP systems might also speed up the diagnosis of failures identified by testing (1 mark for an appropriate tactic, 2 marks for the justification).
- (d) Here I'd expect something like:
 - The involvement of so many stakeholders in the development process (i.e. the four vendor development organisations plus the NHSIC developers) pose a threat to the modifiability of the system and so this is a weakness of the architecture. (3 marks)
 - That NHSIC has control over the integration component is a benefit from a modifiability perspective because NHSIC potentially can resolve many minor differences in reporting from the different GP systems by putting more effort into the integration part of the query tool. (2 marks)

2. (a) Probably Model View Controller is the best match - the model captures the situations developing in the emergency, it has the potential for different interfaces and controllers for each user class. It allows separation of concerns between the model and the presentation to the different user classes. (2 marks for the choice and up to 4 marks for the justification of how it matches the structure of the problem.)
- (b) A wide variety of possibilities (layered, pipe and filter, multi-tier, etc.). I'd anticipate the justification would be along the lines that the pattern does not really address the requirements for the situation. For example, multi-tiers might be addressing issues in scalability, performance and availability but these have not been mentioned explicitly in the question. (2 marks for the choice and up to 4 marks for the justification of how it fails to match the structure of the problem.)
- (c) This should be an MVC diagram where the model is annotated to carry emergency situation and there are multiple view/controller pairs for the different user classes. Allocate two marks for the diagram and up to three for the annotation.
- (d) Here I'd accept a range of suggestions (1 mark for a sensible suggestion, up to three marks for discussion), for example:
 - Authentication: Ensures there is no anonymous actions in the system but this also means that people can be locked out of the system, access can be slower, people forget passwords etc so there may be disadvantages in an emergency situation. This might also allow some personalisation of the controller functionality that could be beneficial.
 - Authorisation: This would ensure that individuals were not permitted to access some of the functionality depending on their identity. This could be beneficial in disallowing some people access to critical functions but it is also the case that this would exclude people from taking over activities in the event of an individual being injured or otherwise incapacitated.
 - Encrypted communication: would ensure there listening in was difficult so malicious individuals could not track the operation. The disadvantage is it closes the system and in an emergency it may be necessary to coordinate volunteer as well as professional people and volunteers may not have access to the encryption infrastructure.
3. For each of these I want to see the students thinking about what might change on the basis of their reading.
 - (a) Here I would anticipate:
 - Some discussion of Product Line Architecture and how this might change the working process in the company since to some extent a single architecture would be adopted and this might change working practice significantly. (3 marks)
 - The other approach might be to be more explicit in the use of Software Architecture as a means to manage Quality Attributes. This might change the view of the approach to requirement gathering in the V-model approach since we might use the architecture as the means of capturing Quality Attribute requirements. (3 marks)

- (b) I think the most obvious choice is to adopt a product line approach in order to deal with the variability in the different controllers developed by the company. This would be disruptive to the development process of different development teams since it would impose more component sharing. It may also have an influence on the quality of the products because the adoption of a unified architecture would involve abandoning tried and tested approaches. There may also be potential for difficulties with regulators over the adoption of new practice. (2 marks each for a discussion of issues such as process change, quality, regulation, skills rigidities, change of toolset, ...)
- (c) Here I would expect some additional discussion of the effects of Product Line on the lifecycle (2 marks per item up to a max of 6 marks):
- Point out that the V-model has an explicit architecture phase that may need to be revised.
 - The architectures of different development groups would need to be aligned.
 - There would need to be new infrastructure to encourage/enforce sharing across development groups
 - There would need to be a different approach to fault localisation and repair since that would operate over the whole component set not a single controller or family of similar controllers.
 - This is not a closed list — other items are possible.
- (d) The answer should discuss the use of agility inside the architecture where this is quite easily feasible. I'd also expect some discussion of agile architecting where there is a need to consider the articulation of the top-down architecture (mostly product line based) and more bottom-up architecture driven from the structure of the sensors and actuators for a specific class of controllers. (2 marks each for top-down and bottom-up considerations).
- (e) There are a range of issues (1-2 marks per issue - max of 3):
- Long-term there is a potential gain because of wider operational experience with the core system components because of greater reuse.
 - Short-term this is a potential loss in quality caused by discarding components with a well-established quality record.
 - Architectural change can also be expensive because it change the environment of use of the system and so the demand on components in operation changes.