

## **General comment**

There were at least three commonalities amongst the very strong solutions:

- (1) good citations and referencing to relevant standards and methods (e.g., IEEE Requirements Specification, SCRUM);
- (2) an attempt at cross-referencing between the different artefacts (e.g., between requirements and architecture, between requirements and risk register, between team organisation and risk register);
- (3) a strong attempt to keep requirements/conceptual architecture separate from design - the latter is about “how” the system works, the former, which is the focus of this submission, is about “what” the system should ultimately do.

Some of these points are elaborated on below.

## **Requirements**

Processes were discussed rather vaguely - good solutions had a short abstract description of the RE process, and then followed it.

Justifications for why the RE process (for the team and project) was appropriate were missing from all but a few submission. The best answers referred to standards and suitable methods in the literature.

Many requirements specifications suffered from “design” pollution, i.e., solutions were discussed rather than focusing on what the system should be providing. The best solutions kept this to a minimum.

Non-functional requirements were poorly recorded overall (if they were considered at all).

Good solutions considered user and system requirements, and linked the two; they also linked requirements to the risk register.

## **Architecture**

Very few really good answers, because very few teams seemed to have researched architectures and conceptual modelling.

Most used UML class diagrams, which generally did not follow the advice given in lectures - they had too many classes (expanded generalisations, implementational classes) and too much detail (concrete attribute/parameter types, specific operations). The problem with doing this is that when you need to change your specification, there is a lot of reworking to do: it is much easier to extend models than rework them. Also, it makes it unlikely that the project you chose for submissions 3 and 4 will match your specification (reworking is then essential).

Many teams included some sort of functionality, and it was not always inappropriate to do so, because some teams clearly thought about their concept in behavioural terms. The best used either activity diagram, use cases, or a flow chart to outline, in general terms, what a game play

looks like. There was also one very nice duck state chart (but, unfortunately, no duck class for it to represent the behaviours of). Attempts to use other behavioural models were generally badly done or represented too low a level breakdown of activities.

The best architecture justifications referred to their game requirements, and the general structures needed for a system to meet these requirements. A few also justified components and structures, notably in relation to a known game structure. However, many teams just described what was in the diagram without any explanation of why these were appropriate components, associations and details for the problem in hand. Teams who included some sort of functionality generally forgot to give any justification or explanation of their behavioural models.

Whilst all teams submitted a separate architecture report, many mixed up the answers to part (a) and part (b) which made mark allocation hard.

Some teams produced diagrams with miniscule print (in several cases, even full screen zoom-in did not reveal detail), and several reports used a font size that was uncomfortably small (even for sections where the full page limit was not used).

### **Method selection and planning**

Quite a range of answers. The best ones identified quickly that this is a small team project and that it was suited to agile, thereby quickly dismissing other approaches. Some teams wrote lovely essays on software engineering processes and methods, which really did not answer the question as set.

Most choices of approach were good, and some went to some lengths to explain how and why the method would be used and adapted. Some nice commentary on where aspects of more than one approach might complement, or be explored.

Tools were usually covered well - though communication tools were often omitted. Most teams did not mention (here or part 4c) planning tools, even though these are used in this submission.

Some teams just wrote down every tool they might ever use and several other things that might be interpreted as tools (such as LibGDX/Java), which did not answer the question.

Planning was challenging because of the tight page limit. A few teams managed a sensible Gantt chart (or similar) and a short explanation, which was ideal. Many teams put far too much detail into the plan presented, and thus could not provide any commentary (fine to have a day-by-day plan for your team, but you don't need to present it at that level of detail). Many teams ignored the "SEPR project plan" and only gave any information on assessment 1. The sensible approach here was to give a reasonably-detailed plan for the past assessment, and perhaps the one about to start, to include a full-project plan (with decreasing detail for later stages) on the website and give us the URL, and briefly outline the key steps in later stages.

**Risks**

Risk register formats were discussed quite clearly, but justified less well; the best solutions clearly explained why the team/project thought the risk format was appropriate. The very best solutions referred to existing risk formats and examples.

Risk specifications were done, generally, quite well; the strongest solutions presented arguments for comprehensiveness and validation. Weaker solutions suffered from imprecision and ambiguity, and did not provide a short argument as to why they thought the register was comprehensive. In some cases there was too much reliance on “the customer” to mitigate risks.

**Website**

All teams provided at least an informational website. The best solutions made it easy to navigate to find the important information, and clearly thought about future enhancements/uses in the next three assessments.