

CITS5502 Software Processes

Sample exam problems

The following are examples of the style of exam question you may expect, and (not exhaustive) points from sample answers to those example questions. In some cases, “working notes” are also provided, outlining how the principles we have covered could be applied.

Note that there are many other possible sample answers – what is important is being able to identify the relevant principles and justify a method of applying them.

Sample question 1

You are a senior developer on a team which will be developing a new content management system (CMS) for the University of Wolloomooloo. Users from different parts of the university (for instance, academic staff and administrative staff) and different faculties want to be able to customize the system in slightly different ways, and from experience, you know that academic staff in the Faculty of Engineering are likely to request late changes to the system requirements. The project manager for the project has asked you what software development process model you would recommend adopting. She informs you that the deliverables include developing (for each user group) thorough documentation, including user manuals, for the system and any customizations. In your answer recommend a model, justifying your choice, and outline one entry criterion that could apply to an activity within this model.

Working notes:

- From the problem description, we note that it sounds like whatever model we adopt should allow for flexibility and the ability to accommodate changes to requirements.
- That suggests we probably don't want to recommend (for instance) using a waterfall model.
- But we might recommend something like the RAD model (which favours minimal planning, and rapid prototyping) or an agile method.
- However, if we recommend agile, we should point out the other requirements of whatever model we choose (e.g., we will need to have suitable “customer representative” who can represent all stakeholders and spend significant time with the development team).

Furthermore, agile methodologies promote “working product” over “comprehensive documentation”, so may not be a good fit here (since thorough documentation is required).

- Therefore we might decide to recommend RAD, as not imposing those requirements.
- If there are any other factors we think are relevant we may as well mention them - e.g., given that CMSs are a pretty common sort of system, our prototyping may simply involve using an existing CMS framework, and customizing it to come up with a prototype.

Points from a sample answer:

- This project requires a software development process model which allows for flexibility and the ability to accommodate changes to requirements.
- My suggestion would be to adopt the Rapid Application Development (RAD) model.
- The RAD model focuses on minimal planning and rapid prototyping. I would suggest that an initial CMS prototype be developed to ensure it meets the University stakeholders' over-arching requirements for a CMS, and then additional prototypes which customize the CMS for particular groups of users as and when their requirements become known.
- Many CMS frameworks are available, so unless there are unusual requirements, prototyping may simply involve installing and customizing an existing framework.
- The RAD model is likely to be a much better fit to this project than the waterfall model, since the waterfall model requires that requirements be (in large part) fixed before the design and implementation phases begin.
- It may also be a better fit than an agile model; given there are a number of *different* user/stakeholders groups involved, it could be difficult to find a single "customer representative" who can adequately represent all those user groups.
- Additionally, agile processes explicitly value "working software over comprehensive documentation"; given that comprehensive documentation is required in this case, again, agile may not be the best fit.
- As an example of an entry criterion for an activity in the RAD model: one is that, before feedback is sought from a particular customer group, a prototype must have been developed for that group.

Sample question 2

You are part of the software quality assurance division for a major car manufacturer, WMW (Westfälisch Motoren Werke), and your company has recently finished developing safety-critical software for the anti-lock braking system (ABS) for one of your cars. The task of carrying out a PIR (Post Implementation Review) has been assigned to one of the work experience interns who assisted with the project. They will be sending an email questionnaire to developers who worked on the project, and summarizing the results for management.

Is this a good way of conducting a PIR? If not, what would you suggest instead? Justify your answer.

Points from a sample answer:

- No, this not a good way of conducting a PIR.
- PIRs should be carried out by a staff-member who is experienced in review techniques and independent of the project team. The intern is perhaps the worst possible

choice, since they worked on the project team (and thus are not independent) and are extremely unlikely to be experienced in review techniques.

- Furthermore, best practice suggests that the PIR involve structured interviews, not an email questionnaire as is being suggested here.
- Given that WMW has a quality assurance division, that is a much more likely place to find someone experienced in review techniques, and furthermore, it should be easier to find someone independent of the original project.
- *[Outline key steps in a PIR]*

Sample question 3

You are the project manager for one of the software development teams for BookFace, a large “social networking” company. The company plans to launch a new “chat”-style application, available on the web, for various desktop platforms (such as Windows and Mac OS X), and for a range of mobile devices. Different programming languages and technologies will be used to target the different platforms. The company is currently attempting to improve their software development processes, and the project board has suggested a number of metrics be tracked for your project:

- average lines of code produced per person-hour
- number of bugs identified by the testing team per week
- customer satisfaction

Discuss whether these are appropriate metrics to track, justifying your answer. If they are not, discuss what changes you would suggest.

Working notes:

- Since it is explicitly stated that multiple languages are used, lines of code alone is unlikely to be a useful metric. It may be possible to produce some given functionality in far fewer lines in one language (say, Python) than another (say Java). It will also be difficult to compare metrics from the present project with previous projects implemented using different languages.
- Many other possible points plus improvement suggestions could be made, based on the material covered on metrics. e.g. are the metrics “informally valid”, are they sustainable over time.

Points from a sample answer:

- “Lines of code produced” is unlikely to be a useful metric on its own, since the project is to be developed in multiple languages. Although the data is still worth recording, it might be better to track a language-neutral metric based on (for instance) cyclomatic complexity, or function points delivered. It could also be useful to record data about the *quality* of the code produced. (For example, tracking how many defects arose from particular modules, or assessing the quality based on code reviews.) That way, developers could be encouraged to write good quality code, not merely large amounts of code.
- Two of the suggested metrics are not very robust. Lines of code is not robust, in the sense that it can be manipulated – developers might choose to write code in a more verbose way, or reimplement functionality

from libraries, in order to inflate the line count.

“Number of bugs identified per week” is likewise not robust – testers or developers might deliberately introduce (minor) bugs so as to inflate the bug count.

Tracking function points delivered would be more robust than a metric based on lines of code. Recording the number of bugs found is useful data, but it might be more useful to use something like code coverage to help assess the testing team’s work.

- “Customer satisfaction” without more information on how it is defined and measured seems too vague to be a metric – good metrics should be objective, reproducible and precise, so more information is needed on how customer satisfaction is to be measured. This does not mean customer satisfaction should be ignored, just that standard procedures should be set up for measuring and tracking it. (E.g., this could consist of surveys that customers are prompted to fill in when using the software.)

Sample question 4

You are team leader of a group of developers producing a new version of CAD/CAM (Computer Aided Design/Manufacturing) software for use by the ship-building industry. One of the senior developers has told you that the version is ready to ship; the testing team lead tells you she thinks there are still defects in the software. How could you estimate the number and severity of defects still remaining in the software? Explain what information you might need and what assumptions you would need to make.

Points from a sample answer:

- Given data on what bugs have been identified, when they were identified, and what impact they had, we could propose a mathematical model of defects found per unit time period, check how well it fits the data, and use this to estimate the remaining defects in the system.
- This assumes that when integrated over time to infinity, our model produces a finite result. This seems a reasonable assumption, however – any model which does not do so seems unrealistic.
- We may need to make the assumption that defects are independent of each other (even though this is not always true), or our model may become intractable or over complex.
- We may also need to assume that the staff devoted to testing (for some amount of person-hours) are interchangeable with other staff. This might or might not be realistic; if some staff are vastly more experienced or efficient at locating defects, we might need to add to our model which staff found which defects.
- *[Other techniques that could be mentioned include, inter alia, mutation testing and deliberate defect injection.]*