ENGINEERING SOFTWARE PRODUCTS

SOLUTIONS TO EXERCISES

1. Solutions to exercises

1.1 Briefly describe the fundamental difference between project-based and product-based software engineering.

The most important difference is that in project-based software engineering there is an external client for the software who is responsible for specifying their needs as 'requirements'. For products, the owner of the product is responsible for specifying its features and may change this specification in response to schedule pressure or technical difficulties.

1.2 What are three important differences between software products and software product lines.

1. There is a single version of a software product for all customers in a particular market. For software product lines, every delivered product is different and adapted to specific customer needs.
2. Software product lines are designed for evolution so that the needs of different clients can be accommodated. While evolution is important for software products, issues such as performance and usability are usually more important.
3. Product updates are more complex for software product lines as it is important to check that these do not interfere with every client version of the software.

1.3 Based on the vision example for the iLearn system, identify the 'WHAT, WHO and WHY' for that software product.

WHAT: An open-learning environment that allows teachers to configure the set of tools and resources available to teachers and students to meet the specific needs of these users.

WHO: Teachers and other educators.

WHY: Because existing learning environments are inflexible with a focus on the administration of learning rather than the learning process itself.

1.4 Why do software product managers have to be generalists with a range of skills rather than simply technical specialists?

Because they have to interface and interact with a wide range of people who may not be technical experts. Typically this will include potential users, company managers and managers of companies who are potential customers for the product. Of course, they also have to interact with the development team so must understand technical issues that may be important.

1.5 You are a software product manager for a company developing educational software products based around scientific simulations. Explain why it is important to develop a product roadmap so that final product releases are available in the first three months of the year.

The market for this software is science departments in schools and colleges. The product is a simulation system so teachers need to have time to evaluate it before deciding on

whether or not it suits their needs. Typically, schools and colleges work around an academic year, which starts in September and runs through till May or June.

Therefore, to use the product in a new academic year, it has to be installed by June of that year at the latest as teachers and instructors cannot be guaranteed to be available during the vacation. All institutions have a purchasing process and there may be several weeks between a decision to purchase and the delivery of the software. Therefore, if the software is not available in the first three calendar months of the year, it is probably impossible to evaluate, purchase, install and test the product in time for the new academic year.

1.6 Why should you develop a prototype before you start developing a new software product?

You should always develop a prototype system for four reasons:

1. If you are looking for funding for your product development, a prototype shows possible funders what your software can do. It is much easier to explain to them why your company is worth investing in if they can see customer benefits.
2. A prototype can be used for initial customer experimentation to see if the features that you propose to include are genuinely useful and likely to be used by customers.
3. You can use a prototype to investigate the feasibility of technologies (e.g. NoSQL databases) that you might use in your product.
4. Developing a prototype helps you understand the overall structure of your system. This is helpful in deciding how to decompose your final implementation into fundamental software units, such as services.