

Using the use case model in system development

Each of the UML models concentrates on modelling its own particular aspects of the system while ignoring others: they provide complementary views of the system. The use case model gives the user coherent and detailed documentation of what the system does or will do. However, in addition to this, it also provides direct support for other parts of the software development process.

Checking the system using the use case model. The use case model provides an excellent basis for system testing. As each use case models a significant task, the use cases provide ready made testing units. The developer can take each use case in turn and check that the system does what it is meant to, i.e. provides the functionality specified. The scenarios provide instances of the normal sequence of events in the use case and the main alternatives; exactly what is required for testing purposes. The developer must check that the system can handle each scenario. Checking the design of the system can be done by walking through each use case to verify that it can be realized. When the system has been implemented, system tests can be derived from the use cases to check that the code produces the required functionality.

Estimating using use cases. Project managers find use cases useful for planning and estimating the development process. Once the system has been broken down into a set of use cases and the nature of each use case understood, the project manager will have a good idea of how long each will take to develop, how much risk is attached to each and how vital each is to the success of the project. This helps the project manager to produce an informed project plan; the system can be developed use case by use case. If the project manager knows that the system is not going to be produced as quickly as the customer would like, he can make sure that the most vital use cases are tackled first.

Basis for interaction diagrams. Use cases are closely related to interaction diagrams, i.e. sequence diagrams and collaboration diagrams (see Chapter 6). A use case provides a description of a particular task; the corresponding interaction diagram shows how that use case will work in terms of messaging between objects (see Chapter 6). Like the scenario, an interaction diagram shows what happens in a specific instance; loosely speaking the scenario gives a step-by-step account of what happens on the user's side of the computer screen and the interaction diagram gives a step-by-step account of what happens on the other side of the screen.