2.1.2 Software Development Lifecycle Models in Context

Software development lifecycle models must be selected and adapted to the context of project and product characteristics. An appropriate software development lifecycle model should be selected and adapted based on the project goal, the type of product being developed, business priorities (e.g., time-to-market), and identified product and project risks. For example, the development and testing of a minor internal administrative system should differ from the development and testing of a safety-critical system such as an automobile's brake control system. As another example, in some cases organizational and cultural issues may inhibit communication between team members, which can impede iterative development.

Depending on the context of the project, it may be necessary to combine or reorganize test levels and/or test activities. For example, for the integration of a commercial off-the-shelf (COTS) software product into a larger system, the purchaser may perform interoperability testing at the system integration test level

(e.g., integration to the infrastructure and other systems) and at the acceptance test level (functional and non-functional, along with user acceptance testing and operational acceptance testing). See section 2.2 for a discussion of test levels and section 2.3 for a discussion of test types.

In addition, software development lifecycle models themselves may be combined. For example, a V-model may be used for the development and testing of the backend systems and their integrations, while an Agile development model may be used to develop and test the front-end user interface (UI) and functionality. Prototyping may be used early in a project, with an incremental development model adopted once the experimental phase is complete.

Internet of Things (IoT) systems, which consist of many different objects, such as devices, products, and services, typically apply separate software development lifecycle models for each object. This presents a particular challenge for the development of Internet of Things system versions. Additionally the software development lifecycle of such objects places stronger emphasis on the later phases of the software development lifecycle after they have been introduced to operational use (e.g., operate, update, and decommission phases).