successful development of software systems. The tools offered by RUP allow a large part of the development process to be automated, including modelling, programming, testing, managing the project and managing change.

RUP is based on the following six 'Best Practices' which have been formulated from experience on many industrial software projects:

1 Develop software iteratively

2 Manage requirements

3 Use component-based architectures

4 Visually model software

5 Verify software quality

6 Control changes to software.

1 *Develop software iteratively*

RUP follows the phases of the generic object-oriented life cycle (inception, elaboration, construction and transition) described earlier in this chapter. It is built on the central concept of iterative development (as shown in Figure 1.3) and each of its phases defines a series of activities that may be performed once or a number of times. Each iteration is defined as a complete development loop resulting in the release of an executable product that is a subset of the final system. In this way RUP supports incremental development – the frequent release of small packages of software that gradually build up to become the final system. Iteration and incremental development encourage involvement and feedback from clients and users; they make it easier to cope with changes, and reduce the risk factors associated with any development project.

2 *Manage requirements*

RUP offers sound support for eliciting, organizing and recording requirements. Precise documentation of requirements facilitates traceability through the development process, which enhances the quality of the final system. The emphasis on the activities that take place early on in the life cycle provides a sound foundation for the later stages and results in systems that are robust, reliable and meet the needs of their users.

3 *Use component-based architectures*

RUP prescribes the early identification and development of a system structure that is at the same time robust enough to ensure system reliability, and flexible enough to accommodate