in Figure 2.11.

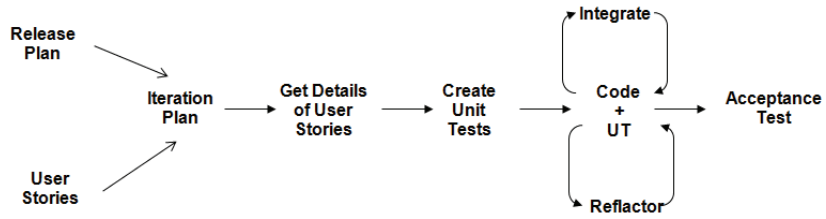

Figure 2.11: An iteration in XP.

This is a very simplified description of XP. There are many other rules in XP relating to issues like rights of programmers and customers, communication between the team members and use of metaphors, trust and visibility to all stakeholders, collective ownership of code in which any pair can change any code, team management, building quick *spike solutions* to resolve difficult technical and architectural issues or to explore some approach, how bugs are to be handled, how what can be done within an iteration is to be estimated from the progress made in the previous iteration, how meetings are to be conducted, how a day in the development should start, etc. The website www.extremeprogramming.org is a good source on these, as well as other aspects of XP.

XP, and other agile methods, are suitable for situations where the volume and pace of requirements change is high, and where requirement risks are considerable. Because of its reliance on strong communication between all the team members, it is effective when teams are collocated and of modest size, of up to about 20 members. And as it envisages strong involvement of the customer in the development, as well as in planning the delivery dates, it works well when the customer is willing to be heavily involved during the entire development, working as a team member.

2.3.7 Using Process Models in a Project

We have seen many different development process models. What is the need for the different models? As mentioned earlier, while developing (industrial strength) software, the purpose is not only to develop software to satisfy the needs of some users or clients, but we want that the project be done in low cost and cycle time, and deliver high-quality software. In addition, there could be other constraints in a project that the project may need to satisfy. Hence, given the constraints of the project, we would like to employ the process model

that is likely to maximize the chances of delivering the software, and achieve the highest Q&P. Hence, selecting a suitable development process model for a project is a key decision that a project manager has to take. Let us illustrate this by a few examples.

Suppose a small team of developers has been entrusted with the task of building a small auction site for a local university. The university administration is willing to spend some time at the start to help develop the requirements, but it is expected that their availability will be limited later. The team has been given 4 months to finish the project, and an extension of the deadline seems very improbable. It also seems that the auction site will have some features that are essential, but will also have some features that are desirable but without which the system can function reasonably well.

With these constraints, it is clear that a waterfall model is not suitable for this project, as the "all or nothing" risk that it entails is unacceptable due to the inflexible deadline. The iterative enhancement model where each iteration does a complete waterfall is also not right as it requires requirements analysis for each iteration, and the users and clients are not available later. However, the iterative delivery approach in which the complete requirements are done in the first iteration but delivery is done in iterations seems well suited, with delivery being done in two (or three) iterations (as time is short). From the requirements, the project team can decide what functionality is essential to have in a working system and include it in the first iteration. The other desirable features can be planned for the second iteration. With this approach, the chances of completing the first iteration before the final deadline increase. That is, with this model, the chances of delivering a working system increase. RUP, as it allows iterations in each phase, is also a suitable model.

Consider another example where the customers are in a highly competitive environment where requirements depend on what the competition is doing, and delivering functionality regularly is highly desirable. Furthermore, to reduce cost, the customer wants to outsource as much project work as possible to another team in another country.

For this project, clearly waterfall is not suitable as requirements are not even known at the start. Iterative enhancement also may not work as it may not be able to deliver rapidly. XP will be hard to apply as it requires that the entire team, including the customer, be collocated. For this project, the timeboxing model seems to fit the best. The whole project can employ three teams—one of analysts who will work with the customer to determine the requirements, one to do the development (which could be in some low-cost destination), and the third to do the deployment, which will be where the site is hosted. By suitably staffing the teams, the duration of each of the three phases—analysis and design, build, and deployment—can be made approximately equal. Then

the timeboxing model can be applied.

Consider another project, where a university wants to automate the registration process. It already has a database of courses and pre-requisites, and a database of student records. In this project, as the requirements are well understood (since registrations have been happening manually), the waterfall model seems to be the optimum.

2.4 Project Management Process

While the selection of the development process decides the phases and tasks to be done, it does not specify things like how long each phase should last, or how many resources should be assigned to a phase, or how a phase should be monitored. And quality and productivity in the project will also depend critically on these decisions. To meet the cost, quality, and schedule objectives, resources have to be properly allocated to each activity for the project, and progress of different activities has to be monitored and corrective actions taken when needed. All these activities are part of the project management process. Hence, a project management process is necessary to ensure that the engineering process ends up meeting the real-world objectives of cost, schedule, and quality.

The project management process specifies all activities that need to be done by the project management to ensure that cost and quality objectives are met. Its basic task is to ensure that, once a development process is chosen, it is implemented optimally. That is, the basic task is to plan the detailed implementation of the process for the particular project and then ensure that the plan is properly executed. For a large project, a proper management process is essential for success.

The activities in the management process for a project can be grouped broadly into three phases: planning, monitoring and control, and termination analysis. Project management begins with planning, which is perhaps the most critical project management activity. The goal of this phase is to develop a *plan* for software development following which the objectives of the project can be met successfully and efficiently. A software plan is usually produced before the development activity begins and is updated as development proceeds and data about progress of the project becomes available. During planning, the major activities are cost estimation, schedule and milestone determination, project staffing, quality control plans, and controlling and monitoring plans. Project planning is undoubtedly the single most important management activity, and it forms the basis for monitoring and control. We will devote one full chapter

later in the book to project planning.

Project monitoring and control phase of the management process is the longest in terms of duration; it encompasses most of the development process. It includes all activities the project management has to perform while the development is going on to ensure that project objectives are met and the development proceeds according to the developed plan (and update the plan, if needed). As cost, schedule, and quality are the major driving forces, most of the activity of this phase revolves around monitoring factors that affect these. Monitoring potential risks for the project, which might prevent the project from meeting its objectives, is another important activity during this phase. And if the information obtained by monitoring suggests that objectives may not be met, necessary actions are taken in this phase by exerting suitable control on the development activities.

Monitoring a development process requires proper information about the project. Such information is typically obtained by the management process from the development process. Consequently, the implementation of a development process model should ensure that each step in the development process produces information that the management process needs for that step. That is, the development process provides the information the management process needs. However, interpretation of the information is part of monitoring and control.

Whereas monitoring and control last the entire duration of the project, the last phase of the management process—termination analysis—is performed when the development process is over. The basic reason for performing termination analysis is to provide information about the development process and learn from the project in order to improve the process. This phase is also often called *postmortem analysis*. In iterative development, this analysis can be done after each iteration to provide feedback to improve the execution of further iterations. We will not discuss it further in the book; for an example of a postmortem report the reader is referred to [57].

The temporal relationship between the management process and the development process is shown in Figure 2.12. This is an idealized relationship showing that planning is done before development begins, and termination analysis is done after development is over. As the figure shows, during the development, from the various phases of the development process, quantitative information flows to the monitoring and control phase of the management process, which uses the information to exert control on the development process.

We will in a later chapter discuss in detail the project planning phase. As a plan also includes planning for monitoring, we will not discuss the monitoring separately but discuss it as part of the planning activity.