Software Process Improvement

15

Abstract

This chapter discusses software process improvement. It begins with a discussion of a software process and discusses the benefits that may be gained from a software process improvement initiative. Various models that support software process improvement are discussed, and these include the Capability Maturity Model Integration (CMMI), ISO 9000, Personal Software Process (PSP) and Team Software Process (TSP).

15.1 Introduction

The success of business today is highly influenced by the functionality and quality of the software that it uses. It is essential that the software is safe, reliable, of a high quality and fit for purpose. Companies may develop their own software internally, or they may acquire software solutions off-the-shelf or from bespoke software development. Software development companies need to deliver high-quality and reliable software consistently on time to their customers.

Cost is a key driver in most organizations, and it is essential that software is produced as cheaply and efficiently as possible, and that waste is reduced or eliminated in the software development process. In a nutshell, companies need to produce software that is *better, faster and cheaper* than their competitors in order to survive in the marketplace. Another words, companies need to continuously work

smarter to improve their businesses, and to deliver superior solutions to their customers.

Software process improvement initiatives are aligned to business goals and play a key role in helping companies achieve their strategic goals. It is invaluable in the implementation of best practice in organizations and allows companies to focus on fire prevention rather than firefighting. It allows companies to problem solve key issues to eliminate quality problems, and to critically examine their current processes to determine the extent to which they meet its needs, as well as identifying how the processes may be improved and identifying where waste can be minimized or eliminated.

It allows companies to identify the root causes of problems (e.g. using the *five why tool*) and to determine appropriate solutions to the problems. The benefits of successful process improvement include the consistent delivery of high-quality software, improved financial results and increased customer satisfaction.

Software process improvement initiatives lead to a focus on the process and on ways to improve it. Many problems are caused by a defective process rather than people, and a focus on the process helps to avoid the blame culture that arises when blame is apportioned to individuals rather than the process. The focus on the process leads to a culture of openness in discussing problems and their solutions, and in instilling process ownership among the process practitioners.

Software process improvement (SPI) allows companies to mature their software engineering processes and to achieve their business goals more effectively. It helps software companies to improve performance and to deliver high-quality software on time and on budget, as well, reducing the cost of development and improving customer satisfaction. It has become an indispensable tool for software engineers and managers to achieve their goals, and it provides a return on investment to the organization.

15.2   What Is a Software Process?

A software development process is the process used by software engineers to design and develop computer software. It may be an undocumented ad hoc process as devised by the team for a particular project, or it may be a standardized and documented process used by various teams on similar projects. The process is seen as the glue that ties people, technology and procedures coherently together.

The processes employed in software development include processes to determine the requirements, processes for the design and development of the software, processes to verify that the software is fit for purpose and processes to maintain the software.

A *software process* is a set of activities, methods, practices and transformations that people use to develop and maintain software and the associated work products.

Definition 15.1 (Software Process)
A *process* is a set of practices or tasks performed to achieve a given purpose. It may include tools, methods, material and people.

An organization will typically have many processes in place for doing its work, and the objective of process improvement is to improve these to meet business goals more effectively.

The Software Engineering Institute (SEI) believes that there is a close relationship between the quality of the delivered software and the quality and maturity of the underlying processes employed to create the software. The SEI adopted and applied the principles of process improvement used in the manufacturing field to develop process maturity models such as the Capability Maturity Model (CMM) and its successor the Capability Maturity Model Integration (CMMI). These maturity models are invaluable in maturing the software processes in software-intensive organizations.

The process is an abstraction of the way in which work is done in the organization, and it is seen as the glue (Fig. 15.1) that ties people, procedures and tools together.

A process is often represented by a process map which details the flow of activities and tasks. The process map will typically include the inputs to each activity and the output from an activity. Often, the output from one activity will become an input to the next activity. A simple example of a process map for creating the system requirements specification is described in Fig. 15.2. The input to the activity to create the system requirements specification will typically be the business (user) requirements, whereas the output is the system requirements specification document itself.
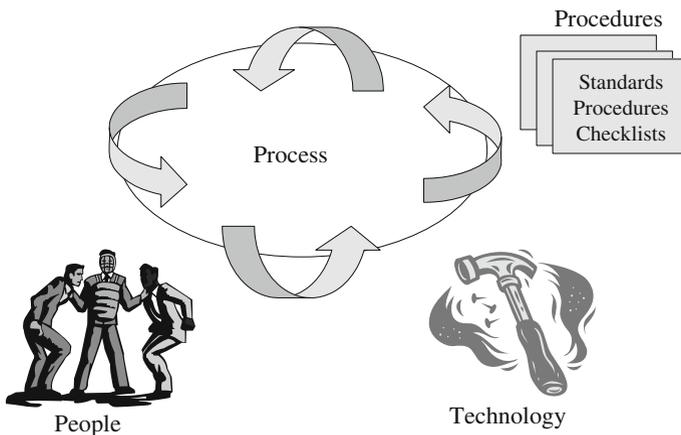

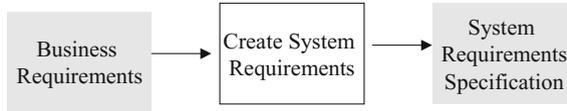**Fig. 15.1** Process as glue for people, procedures and tools

Fig. 15.2 Sample process map

As a process matures, it is defined in more detail and documented. It will have clearly defined entry and exit criteria, inputs and outputs, an explicit description of the tasks, verification of the process and consistent implementation throughout the organization.

15.3   What Is Software Process Improvement?

The origins of the software process improvement field go back to Walter Shewhart's work on statistical process control in the 1930s. Shewhart's work was later refined by Deming and Juran, and they argued that high-quality processes are essential to the delivery of a high-quality product. Deming and Juran argued that the quality of the end product is largely determined by the processes used to produce and support, and that therefore there needs to be an emphasis on the process as well as on the product.

These quality gurus argued that product quality will improve as variability in process performance is reduced [1], and their approach was effective in transforming manufacturing companies with quality problems to companies that would consistently deliver high-quality products. Further, the improvements to quality led to cost reductions and higher productivity, as less time was spent in reworking defective products.

The work of Deming and Juran was later applied to the software quality field by Watts Humphrey and others at the SEI leading to the birth of the software process improvement field. Software process improvement is concerned with practical action to improve the software processes in the organization to improve performance, and to ensure that business goals are achieved more effectively. For example, the business goals may be to deliver projects faster with higher quality.

Definition 15.2 (Software Process Improvement)
A program of activities is designed to improve the performance and maturity of the organization's software processes and the results of such a program.

Software process improvement initiatives (Fig. 15.3) support the organization in achieving its key business goals more effectively, where the business goals could be delivering software faster to the market, improving quality and reducing or eliminating waste. The objective is to work smarter and to build software better, faster and cheaper than competitors. Software process improvement makes business sense, and it provides a return on investment.
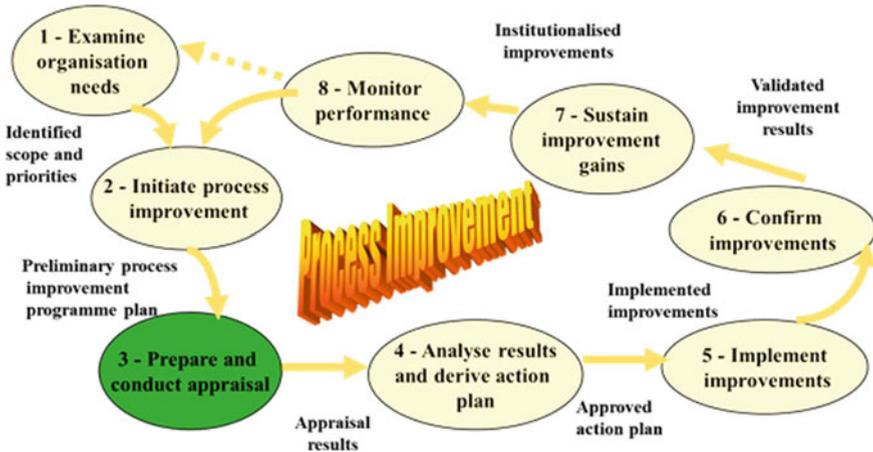
Fig. 15.3 Steps in process improvement

There are international standards and models available to support software process improvement. These include the CMMI model, the ISO 90001 standard and ISO 15504 (popularly known as SPICE). The SEI developed the CMMI model, and it includes best practice for processes in software and systems engineering. The ISO 9001 standard is a quality management system that may be employed in hardware, software development or service companies. The ISO 15504 standard is an international standard for software process improvement and process assessment, and it is popular in the automotive sector.

Software process improvement is concerned with defining the right processes and following them consistently. It involves training all staff on the new processes, refining the processes and continuously improving the processes. The need for a process improvement initiative often arises due to the realization that the organization is weak in some areas in software engineering, and that it needs to improve to achieve its business goals more effectively. The starting point of any improvement initiative is an examination of the business needs of the organization, and these may include goals such as delivering high-quality products on time or delivering products faster to the market.

15.4 Benefits of Software Process Improvement

It is a challenge to deliver high-quality software consistently on time and on budget. There are problems with budget and schedule overruns, late delivery of the software, spiralling costs, quality problems with the delivered software, customer complaints and staff morale.

Software process improvement can assist in dealing with these problems. There are costs involved, but it provides a return on the investment made. Specifically, the benefits from software process improvement include as follows:

- Improvements to quality
- Reductions in the cost of poor quality
- Improvements in productivity
- Reductions to the cost of software development
- Improvements in on-time delivery
- Improved consistency in budget and schedule delivery
- Improvements to customer satisfaction
- Improvements to employee morale

The SEI maintains data on the benefits that organizations have achieved from using the CMMI. These include improvements in several categories such as cost, schedule, productivity, quality, customer satisfaction and the return on investment.

Table 15.1 presents results in software process improvement collaborations of twenty-five organizations taken from conference presentations, published papers and individual [2].

For example, *Northrop Grumman Defense Systems* met every milestone (25 in a row) with high quality and customer satisfaction; *Lockheed Martin* reported an 80% increase in software productivity over a five-year period when it achieved CMM level 5 and obtained further increases in productivity as it moved to CMMI level 5. *Siemens (India)* reported an improved defect removal rate from over 50% before testing to over 70% before testing and a post-release defect rate of 0.35 defects per KLOC. *Accenture* reported a 5:1 return on investment from software process improvement activities.

15.5   Software Process Improvement Models

A process model[1] such as the CMMI defines best practice for software processes in an organization. It describes what the processes should do rather than how they should be done, and this allows the organization to use its professional judgment in the implementation of processes to meet its needs. The process model will need to be interpreted and tailored to the particular organization.

A process model provides a place to start an improvement initiative, and it provides a common language and shared vision for improvement. It provides a framework to prioritize actions and it allows the benefits of the experience of other organizations to be shared. The popular process models used in software process improvement include as follows:

---

[1]There is the well-known adage "All models are wrong, some are useful".

Table 15.1 Benefits of software process improvement (CMMI)

Improvements	Median	#Data points	Low	High
Cost	20%	21	3%	87%
Schedule	37%	19	2%	90%
Productivity	62%	17	9%	255%
Quality	50%	20	7%	132%
Customer satisfaction	14%	6	−4%	55%
ROI	4.7:1	16	2:1	27:1

– Capability Maturity Model Integration (CMMI)
– ISO 9001 Standard
– ISO 15504
– PSP and TSP
– Six sigma
– Root cause analysis (RCA)
– Balanced score card

The CMMI was developed by the SEI, and it is the successor to the older software CMM which was released in the early 1990s. The latter is specific to the software field, and it was influenced by Watts Humphrey's work at IBM [3]. The CMMI is a suite of products used for improving processes, and it includes models, appraisal methods and training material. The CMMI models address three areas of interest:

– CMMI for Development (CMMI-DEV)
– CMMI for Services (CMMI-SVC)
– CMMI for Acquisition (CMMI-ACQ)

The CMMI Development Model is discussed in Chap. 16, and it provides a structured approach to improvement, which allows the organization to set its improvement goals and priorities. The CMMI framework allows organizations to improve their maturity by improvements to their underlying processes. It provides a clearly defined road map for improvement, and it allows the organization to improve at its own pace. Its approach is evolutionary rather than revolutionary, and it recognizes that a balance is required between project needs and process improvement needs. It allows the processes to evolve from ad hoc immature activities to disciplined mature processes.

The CMMI practices may be used for the development, acquisition and maintenance of products and services. A SCAMPI appraisal determines the actual process maturity of an organization, and a SCAMPI class A appraisal allows the organization to benchmark itself against other organizations.

ISO 9001 is an internationally recognized quality management standard (Fig. 15.4), and it is customer and process focused. It applies to the processes that an organization uses to create and control products and services, and it emphasizes

Fig. 15.4 ISO 9001 quality management system

continuous improvement.[2] The standard is designed to apply to any product or service that an organization supplies.

The implementation of ISO 9001 involves understanding the requirements of the standard and how the standard applies to the organization. It requires the organization to identify its quality objectives, define a quality policy, produce documented procedures and carry out independent audits to ensure that the processes and procedures are followed. An organization may be certified against the ISO 9001 standard to gain recognition on its commitment to quality and continuous improvement. The certification involves an independent assessment of the organization to verify that it has implemented the ISO 9001 requirements properly, and that the quality management system is effective. It will also verify that the processes and procedures defined are consistently followed and that appropriate records are maintained. The ISO 9004 standard provides guidance for continuous improvement.

The ISO/IEC 15504 standard (popularly known as ISO SPICE) is an international standard for process assessment. It includes guidance for process improvement and for process capability determination, as well as for performing an assessment. It uses the international standard for software and systems lifecycle processes (ISO/IEC 12207) as its process model.

The ISO 12207 standard distinguishes between several categories of software processes including the primary lifecycle processes for developing and maintaining software, supporting processes to support the software development lifecycle and organizing lifecycle processes. There is a version of SPICE termed "*Automotive SPICE*" that is popular in the automotive sector. ISO/IEC 15504 can be used in a similar way to the CMMI, and its process model (i.e. ISO 12207) may be employed

---

[2]The ISO 9004 standard provides guidance on continuous improvement.

to implement best practice in the definition of processes. Assessments may be performed to identify strengths and opportunities for improvement.

The Personal Software Process (PSP) is a disciplined data-driven software development process that is designed to help software engineers understand and to improve their PSP performance. It was developed by Watts Humphrey at the SEI, and it helps engineers to improve their estimation and planning skills and to reduce the number of defects in their work. This enables them to make commitments that they can keep and to manage the quality of their projects.

The Team Software Process (TSP) was developed by Watts Humphrey at the SEI and is a structured approach designed to help software teams understand and improve their quality and productivity. Its focus is on building an effective software development team, and it involves establishing team goals, assigning team roles as well as other teamwork activities. Team members must already be familiar with the PSP.

Six sigma (6σ) was developed by Motorola as a way to improve quality and reduce waste. Its approach is to identify and remove the causes of defects in processes by reducing process variability. It uses quality management techniques and tools such as the five whys, business process mapping, statistical techniques, and the DMAIC and DMADV methodologies. There are several roles involved in six sigma initiatives such as Champions, Black Belts and Green Belts, and each role requires knowledge and experience, and is awarded on merit subject to training and certification. Sponsorship and leadership are required from top management to ensure the success of a 6σ initiative, and 6σ was influenced by earlier quality management techniques developed by Shewhart, Deming and Juran. A 6σ project follows a defined sequence of steps and has quantified targets (e.g. financial, quality, customer satisfaction and cycle time reduction).

15.6 Process Mapping

The starting point for improving a process is first to understand the process as it is currently performed and to determine the extent to which it is effective. The process stakeholders reach a common understanding of how the process is actually performed, and the process (as currently performed) is then sketched pictorially, with the activities and their inputs and outputs recorded graphically. This graphical representation is termed as "*process map,*" and is an abstract description of the process "*as is.*"

The process map is an abstraction of the way that work is done, and it may be critically examined to determine how effective it really is and to identify weaknesses and potential improvements. This critical examination by the process practitioners leads to modifications to its definition, and the proposed definition is sketched in a new process map to yield the process "*to be.*"

Each activity has an input and an output, and these are recorded in the process map. Once the team has agreed the definition of new process, the supporting templates required become clear from an examination of the input and output of the various activities. There may be a need for standards to support the process (e.g. procedures and templates), and the procedures or guidelines will be documented to provide the details on how the process is to be carried out, and they will detail the tasks and activities, and the roles required to perform them.

15.7   Process Improvement Initiatives

The need for a software process improvement initiative often arises from the realization that the organization is weak in some areas in software engineering, and that it needs to improve to achieve its business goals more effectively. The starting point of any improvement initiative is an examination of the business goals of the organization, and these may include as follows:

– Delivering high-quality products on time
– Delivering products faster to the market
– Reducing the cost of software development
– Improving software quality

There is more than one approach to the implementation of an improvement programme. A small organization has fewer resources available, and team members involved in the initiative will typically be working part-time. Larger organizations may be able to assign people full time on the improvement activities. The software process improvement initiative is designed to enable the organization achieve its business goals more effectively.

Once the organization goals have been defined, the improvement initiative commences. This involves conducting an appraisal (Fig. 15.6) to determine the current strengths and weaknesses of the processes, analysing the results to for-mulate a process improvement plan, implementing the plan, piloting the improved processes and verifying that they are effective, training staff and rolling out the new processes. The improvements are monitored for effectiveness and the cycle repeats. The software process improvement philosophy is as follows:

• The improvement initiative is based on business needs.
• Improvements should be planned based on the strengths and weaknesses of the processes in the organization.
• The CMMI model (or an alternate model) is the vehicle for improvement.
• The improvements are prioritized (it is not possible to do everything at once).
• The improvement initiative needs to be planned and managed as a project.
• The results achieved need to be reviewed at the end of the period, and a new improvement cycle started for continuous improvement

- Software process improvement requires people to change their behaviour, and so organization culture (and training) needs to be considered.
- There needs to be a process champion/project manager to drive the process improvement initiative in the organization.
- Senior management need to be 100% committed to the success of the initiative.
- Staff need to be involved in the improvement initiative, and there needs to be a balance between project needs and the improvement activities.

15.8   Barriers to Success

Software process improvement initiatives are not always successful and occasionally are abandoned. Some of the reasons for failure are as follows:

– Unrealistic expectations
– Trying to do too much at once
– Lack of senior management sponsorship
– Focusing on a maturity level
– Poor project management of the initiative
– Not run as a standard project
– Insufficient involvement of staff
– Insufficient time to work on improvements
– Inadequate training on software process improvement
– Lack of pilots to validate new processes
– Inadequate training/roll-out of new processes

It is essential that a software process improvement initiative be treated as a standard project with a project manager assigned to manage the initiative. Senior management need to be 100% committed to the success of the initiative, and they need to make staff available to work on the improvement activities. It needs to be clear to all staff that the improvement initiative is a priority to the organization. All employees need to receive appropriate training on software process improvement and on the process maturity model.

The CMMI project manager needs to consider the risks of failure of the initiative and to manage them accordingly.

15.9   Setting Up an Improvement Initiative

The implementation of an improvement initiative is a project, and it needs good planning and management to ensure its success. Once an organization makes a decision to embark on such an initiative, a project manager needs to be appointed to

manage the project. The project manager will treat the implementation as a standard project, and plans are made to implement the initiative within the approved schedule and budget. The improvement initiative will often consist of several improvement cycles, with each improvement cycle implementing one or more process areas. Small improvement cycles may be employed to implement findings from an appraisal or improvement suggestions from staff.

One of the earliest activities carried out on any improvement initiative is to determine the current maturity of the organization with respect to the model. This will usually involve an appraisal conducted by one or more experienced appraisers. The findings will indicate the current strengths and weaknesses of the processes, as well as gaps with respect to the practices in the model. This initial appraisal is important, as it allows management in the organization to understand its current maturity with respect to the model and to communicate where it wants to be, as well as how it plans to get there. The initial appraisal assists in prioritizing improvements for the first improvement cycle.

The project manager will then prepare a project plan and schedule. The plan will detail the scope of the initiative, the budget, the process areas to be implemented, the teams and resources required, the initial risks identified, the key milestones, the quality and communication plan and so on. The project schedule will detail the deliverables to be produced, the resources required and the associated timeline for delivery. Project management was discussed in Chap. 2.

The software process improvement initiative is designed to support the organization in achieving its business goals more effectively. The steps include examining organization needs, conducting an appraisal to determine the current strengths and weaknesses and analysing the results to formulate an improvement plan. The improvement plan is then implemented; the improvements monitored and confirmed as being effective, and the improvement cycle repeats. The continuous improvement cycle is described in Fig. 15.5 and Table 15.2.

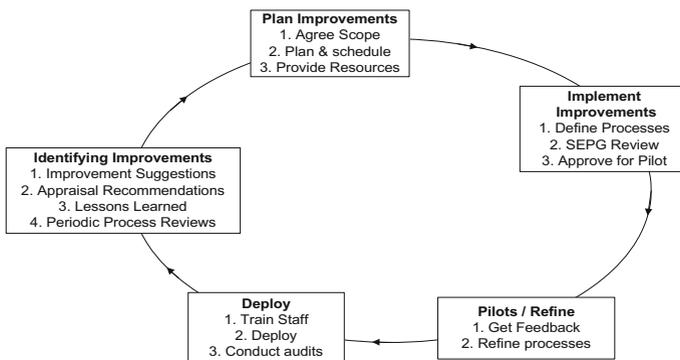The teams involved in implementation are discussed in Table 15.3.

Fig. 15.5 Continuous improvement cycle

Table 15.2 Continuous improvement cycle

Activity	Description

Identify improvements to be made	The improvements to be made during an improvement cycle come from several sources
	  – Improvement suggestions from staff
	  – Lessons learned by projects
	  – Periodic process reviews
	  – Recommendations from appraisals

Plan improvements	A project plan and schedule is prepared for a large improvement cycle (involving the implementation of several process areas). An action plan (with owners and target completion dates) is sufficient for small improvement initiatives
Implement improvements	The improvements will consist of new processes, standards, templates, procedures, guidelines checklists and tools (where appropriate) to support the process
Pilots/refine	Selected new processes and standards will often be piloted[a] prior to their deployment to ensure that they are fit for purpose
Deploy	– Staff are trained on the new processes and standards – Staff receive support during the deployment – Audits are conducted
Do it all again	Improvement is continuous and as soon as an improvement cycle is complete its effectiveness is considered, and a new improvement cycle is ready to commence

[a]The result from the pilot may be that the new process is not suitable to be deployed in the organization or that it needs to be significantly revised prior to deployment

15.10 Appraisals

Appraisals (Fig. 15.6) play an essential role in the software process improvement programme. They allow an organization to understand its current software process maturity, including the strengths and weaknesses in its processes. An initial appraisal is conducted at the start of the initiative to allow the organization to understand its current process maturity, and to plan and prioritize improvements for the first improvement cycle. Improvements are then implemented, and an appraisal is typically conducted at the end of the cycle to confirm that progress has been made in the improvement initiative.

An appraisal is an independent examination of the software engineering and management practices in the organization and is conducted using an appraisal methodology (e.g. SCAMPI). It will identify strengths and weaknesses in the processes and any gaps that exist with respect to the maturity model.

The appraisal leader kicks off the appraisal with an opening presentation, which introduces the appraisal team, and presents the activities that will be carried out during the appraisal. These will include presentations, interviews, reviews of project documentation and detailed analysis to determine the extent to which the practices in the model have been implemented.

Table 15.3 Teams in improvement programme

| Role/Team	Members	Responsibility
Project manager	Project manager	Project manage the improvement project Provide leadership on process improvement
Steering group (project board)	Senior manager(s)/ project manager	Provides management sponsorship of initiative Provides resources and funding for the initiative Uses influence to remove any roadblocks that arise with the improvement activities
SEPG team	Managers, technical and project manager	Coordinate day-to-day improvement activities Provides direction and support to improvement terms Review and approve new processes and coordinate pilots, training and roll-out of new processes
Improvement teams	Process users/project manager	Focus on specific process area(s) Review the current process "as is" and define the new process "to be" Obtain feedback on new process, conduct pilots, refine process, provide training and conduct roll-out of new process
Staff	All affected staff	Participate in improvement teams Participate in pilots Participate in training on new processes Adhere to new processes
External consultancy	External consultant	Conduct appraisal to determine initial maturity and assist in planning of first improvement cycle Provide expertise/training on the maturity model Conduct periodic process reviews Conduct appraisal at end of each improvement cycle
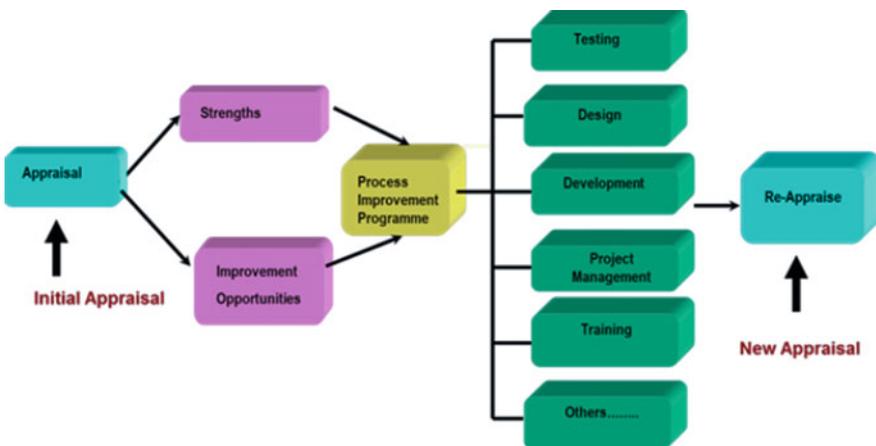
Fig. 15.6 Appraisals

Table 15.4 Phases in an Appraisal

Phase	Description
Planning and preparation	This involves identifying the sponsor's objectives and the requirements for the appraisal. A good appraisal plan is essential to its success
Conducting the appraisal	The appraisal team interviews the participants and examines data to judge the extent to which the CMMI is implemented in the organization
Reporting the results	The findings (including a presentation and an appraisal report).are reported to the sponsor

The appraisal leader will present the appraisal findings, and this may include a presentation and an appraisal report. The appraisal output summarizes the strengths and weaknesses, and ratings of the process areas will be provided (where this is part of the appraisal). The appraisal findings are valuable and will allow the project manager to plan and schedule the next improvement cycle. They allow an organization to:

- Understand its current process maturity (including strengths and weaknesses)
- Relate its strengths and weaknesses to the improvement model
- Prioritize its improvements for the next improvement cycle
- Benchmark itself against other organizations

There are three phases in an appraisal (Table 15.4).

15.11   Review Questions

1. What is a software process?
2. What is software process improvement?
3. What are the benefits of software process improvement?
4. Describe the various models available for software process improvement?
5. Draw the process map for the process of cooking your favourite meal.
6. Describe how a process improvement initiative may be run?
7. What are the main barriers to successful software process improvement initiatives and how can they be overcome?
8. Describe the three phases in an appraisal.

15.12   Summary

The success of business is highly influenced by software, and companies may develop their own software internally, or they may acquire software solutions off-the-shelf or from bespoke software development.

Software process improvement plays a key role in helping companies to improve their software engineering capability and to achieve their strategic goals. It enables organizations to implement best practice in software engineering and to achieve improved results. It allows companies to focus on fire prevention rather than fire-fighting, by critically examine their processes to determine the extent to which they are fit for purpose. It helps in identifying how the process may be improved and how waste may be eliminated.

Software process improvement initiatives lead to a focus on the process, which is important since many problems are caused by defective processes rather than by people. This leads to a culture of openness in discussing problems and instills process ownership among the process practitioners.

Software process improvement helps software companies to deliver the agreed software on time and on budget, as well as improving the quality of the delivered software, reducing the cost of development and improving customer satisfaction.

It has become an indispensable tool for software engineers and managers to achieve their goals, and it provides a return on investment to the organization. The next chapter gives an introduction to the CMMI, which has become a useful framework in maturing software engineering processes.

References

1. W. Edwards Deming, *Out of Crisis* (MIT Press, Cambridge, 1986)
2. Software Engineering Institute, in *CMMI Executive Overview*. Presentation by the SEI, 2006
3. W. Humphry, *Managing the Software Process* (Addison Wesley, New York, 1989)