I-19-QUALITY CONCEPTS

day, 31 May 2022 1:17 AM

⇒ What is Quality ?

⇒ David Garvin of HBS suggests that "quality is a complex and multifaceted concept" that can be described from five different points of view –

1) Transcendental View : Quality is something you immediately recognize, but cannot explicitly define.
2) User View : sees quality in terms of an end user's specific goals. If it meets goals, product exhibits quality.
3) Manufacturer's View : Defines quality in terms of the original specification of the product, If the product conforms to the spec, it exhibits quality.
4) Product View : Quality can be tied to inherent characteristics (functions and features) of a product.
5) Value-based View : Measures quality based on how much a customer is willing to pay for a product.

⇒ Quality of Design refers to the characteristics that designers specify for a product.
In software development, quality of design encompasses the degree to which the design meets the functions and features specified in the requirements model.

⇒ Quality of conformance focuses on the degree to which the implementation follows the design and the resulting system meets its requirements and performance goals.

User Satisfaction = compliant product + good quality + delivery within budget and schedule.

⇒ ISO 9126 Quality Factors [RUF ME P] ⇒ R U free, morning evening for Pyaar

⇒ The ISO 9126 standard was developed in an attempt to identify the key quality attributes for computer software
Six Key Quality Attributes :-

1)- Reliability : The amount of time that the software is available for use as indicated by the following subattributes – maturity, fault tolerance, recoverability.

2)- Usability : The degree to which the software is easy to use as indicated by the following subattributes – understandability

Reliability : The amount of time that the software is available for use as indicated by the following subattributes - maturity, fault tolerance, recoverability.

2)- **Usability :** The degree to which the software is easy to use as indicated by the following subattributes - understandability, learnability, operability.

3)- **Functionality :** The degree to which that software satisfies stated needs as indicated by the following subattributed - suitability, accuracy, interoperability, compliance and security.

4)- **Maintainability :** The case with which repair may be made to the software as indicated by following subattributes - analyzability, changebility, stability, testability.

5) **Efficiency :** The degree to which the software makes optimal use of system resources as indicated by the following subattributes - time behaviour, resource behaviour

6)- **Portability :** The case with which the software can be transposed from one environment to another as indicated by the following subattributes - adaptability, installability, conformance, replaceability.

⇒ Achieving Software Quality [SPOO]

⇒ Software quality is a result of good project management and solid software engineering practice. Management and practice are applied within the context of four broad activities that help a software team achieve high software quality :- SE methods, project management techniques, Quality control actions and software quality assurance.

1)- Software Engineering Methods

⇒ If you expect to build a high quality software, you must understand the problem to be solved.

⇒ You must be capable of creating a design that conforms to the problem while at the same time exhibiting characteristics that lead to software that exhibits the quality dimensions.

Software quality is a result of good project management and solid software engineering practice. Management and practice are applied within the context of four broad activities that help a software team achieve high software quality :- SE methods, project management techniques, Quality control actions and software quality assurance.

1)- Software Engineering Methods

⇒ If you expect to build a high quality software, you must understand the problem to be solved.
⇒ You must be capable of creating a design that conforms to the problem while at the same time exhibiting characteristics that lead to software that exhibits the quality dimensions.

2)- Project Management Techniques

The implications are clear :-

a) If a project manager uses estimation to verify that delivery dates are achievable.
b)- Schedule dependencies are understood and the team resists the temptation to use shortcuts.
c)- Risk planning is conducted so problems do not breed chaos

⇓

Then, Software quality will be affected in a positive way.

3)- Quality Control

⇒ Quality control encompasses a set of software engineering actions that help to ensure that each work product meets its quality goals.
⇒ Models are reviewed to ensure that they are complete and consistent.
⇒ Code may be inspected in order to uncover and correct errors before testing commences.
⇒ A series of testing is applied to uncover errors in processing logic, data manipulation and interface communication.
⇒ A combination of measurement and feedback allows a software team to tune the process when any of these work products fail to meet quality goals.

4)- Quality Assurance

⇒ Quality Assurance establishes the infrastructure that supports solid software engineering methods, national project management and quality control actions — all pivotal if you intend to build high-quality software.
⇒ In addition, quality assurance consists of a set of auditing and reporting functions that access the effectiveness and completeness of quality control actions.
⇒ The goal of quality assurance is to provide management and technical staff with the data necessary to be informed about product quality, thereby gaining insight and confidence that actions to achieve product quality are working.

I-21-ELEMENTS OF SOFTWARE QUALITY SURANCE

Tuesday, 31 May 2022 2:32 AM

→ Software quality assurance encompasses a broad range of concerns and activities that focus on the management of software quality — [RSS VET SECR]

1)- **Reviews and Audits**

Technical reviews are quality control activity performed by software engineers for software engineers.
Audits are a type of review performed by SQA personnel to ensure quality guidelines are being followed.

2)- **Standards** : The IEEE, ISO, and other standard organizations have produced a broad array of software engineering standards and related documents.

3)- **Security Management** : Every software organization should institute policies that protect data at all levels, establish firewall protection for Web Apps.

4)- **Vendor Management** : Three categories of software are acquired from external software vendors — shrink-wrapped packages (Microsoft office), a tailored shell [custom tailored], and contracted software (custom designed and constructed).
The job of the SQA organization is to ensure that high-quality software results by specifying quality practices that Vendor should follow.

5)- **Error/defect collection and analysis** : Only way to improve is to measure how you are doing.
SQA collects and analyzes error and defect data to better **understand how** errors are introduced and what software engineering activities are best suited to eliminating them.

6)- **Testing** : Software control is Quality control function to find errors.
The job of SQA is to ensure that testing is properly planned and efficiently conducted.

shrink-wrapped packages (Microsoft office), a tailored shell [custom tailored], and contracted software (custom designed and constructed).

The job of the SQA organization is to ensure that high-quality software results by specifying quality practices that vendor should follow.

5)- **Error/defect collection and analysis:** Only way to improve is to measure how you are doing.
SQA collects and analyzes error and defect data to better **understand how** errors are introduced and what software engineering activities are best suited to eliminating them.

6)- **Testing:** Software control is quality control function to find errors.
The job of SQA is to ensure that testing is properly planned and efficiently conducted.

7)- **Safety:** The impacts of hidden defects of software can be catastrophic. SQA may be responsible for accessing the impact of software failure and for initiating those steps required to reduce risks

8)- **Educations:** Key contribution to improvent is education of software engineers, managers and stakeholders.
SQA organization is a key proponent and sponsor of educational programs.

9)- **Change Management:** SQA ensures the adequate change management practices have been instituted.

10)- **Risk Management:** SQA ensures that risk management activities are properly conducted and that risk related contingency plans have been established.

35-RISK MANAGEMENT

y, 31 May 2022 3:54 AM

⇒ Reactive v/s Proactive Risk strategies

⇒ Reactive strategy Monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems. The software does nothing about risks until something goes wrong.

⇒ Proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Software team establishes plan for managing risk.

⇒ Software Risks

⇒ Risk involves two characteristics – Uncertainity : risk may or may not happen, i.e., there are no 100% probable risks.
and loss : if the risk becomes a reality, unwanted consequences or losses will occur.

⇒ When risks are analyzed, it is important to quantify the level of uncertainity and the degree of loss associated with each risk.
Different categories of risks :

1)- Project Risks

⇒ Threaten the project plan, if risks become real, project schedule will slip and costs will increase.
⇒ Potential risks identify potential budgetary, schedule, personnel, resource, stakeholder and requirements problems.

2)- Technical Risks

⇒ Threaten quality and timeliness of the software to be produced, if risks become reality, implementation may become difficult or impossible.
⇒ Technical risks ⇒ potential design, implementation, interface, verification and maintaince problems, technical uncertainity, specification ambiguity are also risk fo
⇒ Problem is harder to solve than you thought it would be.

3)- Business Risks

⇒ Threaten viability of the software to be built and often jeopardize the project or the product.
⇒ Business risks ⇒
a)- Building an excellent product no one wants
b) Building a product that no longer fits business strategy of company.
c)- Building a product that the sales force doesn't understand how to sell.
d)- Losing support of senior management
e) Losing budgetary or personnel commitment.

...ration viability of the software to be built and often jeopardize the project or the product.

...siness risks ⇒

a)- Building an excellent product no one wants

b) Building a product that no longer fits business strategy of company.

c)- Building a product that the sales force doesn't understand how to sell.

d)- Losing support of senior management

e) Losing budgetary or personnel commitment.

⇒ General Categorization of risks :-

1)- Known Risks : that can be uncovered after careful evaluation of project plan

2) Predictable Risks : are extrapolated from past project experience Eg- staff turnover, poor communication with customer.

3)- Unpredictable Risks : they can and do occur, but are extremely difficult to identify in advance.

⇒ Risk Identification

⇒ Risk identification is the systematic attempt to specify the threats to project plan (estimate, schedule, resource loading etc).

⇒ General risks are a potential threat to every software project.

⇒ Product specific risks can be identified only by those who have domain knowledge.

⇒ To identify product specific risk, project plan and the software statement of scope are examined.

Checklist for Risk-Identification ⇒ [DPS BP ST] ⇒ DPS ne big P, small T.

1) Development Environment : Risk associated with the availability and quality of tools to be used to build the project.

2)- Product size : Risk associated with overall size of the software to be built or modified.

3)- Stakeholder characteristics : Risks associated with sophistication of stakeholders and developer's ability to communicate with stakeholder.

4)- Business Impact : Risk associated with constraints opposed by management or marketplace.

5)- Process Definition : " " " the degree to which the software process has been defined and is followed by the development organization.

6)- Staff size and experience : Risk associated with overall technical and project experience of the software engineers

7)- Technology to be built : " " " the complexity of the system to be built and the "newness" of the technology that is packaged by the syst

⇒ Answers to these questions allow you to estimate the impact of risk

⇒ Finally, a set of "risk components and drivers" are listed along with their probability of occurrence.

the complexity of the system to be built and the newness of the technology

...ers to these questions allow you to estimate the impact of risk

...ly, a set of "risk components and drivers" are listed along with their probability of occurrence.

35.3.1 Assessing Overall Project Risk

The following questions have been derived from risk data obtained by surveying experienced software project managers in different parts of the world [Kei98]. The questions are ordered by their relative importance to the success of a project.

1. Have top software and customer managers formally committed to support the project?
2. Are end users enthusiastically committed to the project and the system/product to be built?
3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end users have realistic expectations?
6. Is the project scope stable?
7. Does the software engineering team have the right mix of skills?
8. Are project requirements stable?
9. Does the project team have experience with the technology to be implemented?

10. Is the number of people on the project team adequate to do the job?
11. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

If any one of these questions is answered negatively, mitigation, monitoring, and management steps should be instituted without fail. The degree to which the project is at risk is directly proportional to the number of negative responses to these questions.

35.3.2 Risk Components and Drivers

past year

The U.S. Air Force [AFC88] has published a pamphlet that contains excellent guidelines for software risk identification and abatement. The Air Force approach requires that the project manager identify the risk drivers that affect software risk components—performance, cost, support, and schedule. In the context of this discussion, the risk components are defined in the following manner:

- *Performance risk*—The degree of uncertainty that the product will meet its requirement and be fit for its intended use.
- *Cost risk*—The degree of uncertainty that the project budget will be maintained.
- *Support risk*—The degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance.
- *Schedule risk*—The degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time.

The impact of each risk driver on the risk component is divided into one of four impact categories—negligible, marginal, critical, or catastrophic. Referring to Figure 35.1 [Boe89], a characterization of the potential consequences of errors (rows labeled 1) or a failure to achieve a desired outcome (rows labeled 2) are described. The impact category is chosen based on the characterization that best fits the description in the table.

35.4 Risk Projection

Risk projection, also called *risk estimation*, attempts to rate each risk in two

these questions.

35.4 Risk Projection

Risk projection, also called *risk estimation*, attempts to rate each risk in two ways—(1) the likelihood or probability that the risk is real and will occur and (2) the consequences of the problems associated with the risk, should it occur. You work along with other managers and technical staff to perform four risk projection steps:

1. Establish a scale that reflects the perceived likelihood of a risk.
2. Delineate the consequences of the risk.

Figure 35.1 Impact assessment
Source: [Boe89].

Category		Performance	Support	Cost	Schedule
Catastrophic	1	Failure to meet the requirement would result in mission failure		Failure results in increased costs and schedule delays with expected values in excess of $500K	
	2	Significant degradation to nonachievement of technical performance	Nonresponsive or unsupportable software	Significant financial shortages, budget overrun likely	Unachievable IOC
Critical	1	Failure to meet the requirement would degrade system performance to a point where mission success is questionable		Failure results in operational delays and/or increased costs with expected value of $100K to $500K	
	2	Some reduction in technical performance	Minor delays in software modifications	Some shortage of financial resources, possible overruns	Possible slippage in IOC
Marginal	1	Failure to meet the requirement would result in degradation of secondary mission		Costs, impacts, and/or recoverable schedule slips with expected value of $1K to $100K	
	2	Minimal to small reduction in technical performance	Responsive software support	Sufficient financial resources	Realistic, achievable schedule
Negligible	1	Failure to meet the requirement would create inconvenience or nonoperational impact		Error results in minor cost and/or schedule impact with expected value of less than $1K	
	2	No reduction in technical performance	Easily supportable software	Possible budget underrun	Early achievable IOC

Note: (1) The potential consequence of undetected software errors or faults.
(2) The potential consequence if the desired outcome is not achieved.

3. Estimate the impact of the risk on the project and the product.
4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

The intent of these steps is to consider risks in a manner that leads to prioritization. No software team has the resources to address every possible risk with the same degree of rigor. By prioritizing risks, you can allocate resources where they will have the most impact.

35.4.1 Developing a Risk Table

A risk table provides you with a simple technique for risk projection.[2] A sample risk table is illustrated in Figure 35.2.

2 The risk table can be implemented as a spreadsheet model. This enables easy manipulation and sorting of the entries.

Figure 35.2 Sample risk table prior to sorting

Risks	Category	Probability	Impact	RMMM
Size estimate may be significantly low	PS	60%	2	
Larger number of users than planned	PS	30%	3	
Less reuse than planned	PS	70%	2	
End users resist system	BU	40%	3	
Delivery deadline will be tightened	BU	50%	2	
Funding will be lost	CU	40%	1	
Customer will change requirements	PS	80%	2	
Technology will not meet expectations	TE	30%	1	
Lack of training on tools	DE	80%	3	
Staff inexperienced	ST	30%	2	
Staff turnover will be high	ST	60%	2	

Impact values:
1—catastrophic
2—critical
3—marginal
4—negligible

ADVICE
Think hard about the software you're about to build and ask yourself, "what can go wrong?" Create your own list and ask other members of the team to do the same.

You begin by listing all risks (no matter how remote) in the first column of the table. This can be accomplished with the help of the risk item checklists referenced in Section 35.3. Each risk is categorized in the second column (e.g., PS implies a project size risk, BU implies a business risk). The probability of occurrence of each risk is entered in the next column of the table. The probability value for each risk can be estimated by team members individually. One way to accomplish this is to poll individual team members in round-robin fashion until their collective assessment of risk probability begins to converge.

Next, the impact of each risk is assessed. Each risk component is assessed using the characterization presented in Figure 35.1, and an impact category is determined. The categories for each of the four risk components—performance, support, cost, and schedule—are averaged to determine an overall impact value.

KEY POINT
A risk table is sorted by probability and impact to rank risks.

Once the first four columns of the risk table have been completed, the table is sorted by probability and by impact. High-probability, high-impact risks percolate to the top of the table, and low-probability risks drop to the bottom. This accomplishes first-order risk prioritization.

You can study the resultant sorted table and define a cutoff line. The *cutoff line* (drawn horizontally at some point in the table) implies that only risks that lie above the line will be given further attention. Risks that fall below the line are

4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

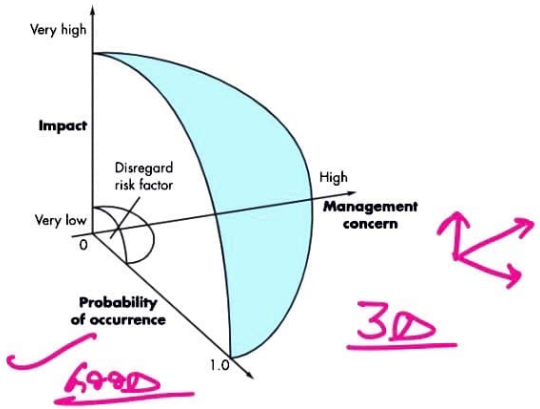
The intent of these steps is to consider risks in a manner that leads to prioritization. No software team has the resources to address every possible risk with the same degree of rigor. By prioritizing risks, you can allocate resources where they will have the most impact.

KEY POINT A risk table is sorted by probability and impact to rank risks.

Once the first four columns of the risk table have been completed, the table is sorted by probability and by impact. High-probability, high-impact risks percolate to the top of the table, and low-probability risks drop to the bottom. This accomplishes first-order risk prioritization.

You can study the resultant sorted table and define a cutoff line. The *cutoff line* (drawn horizontally at some point in the table) implies that only risks that lie above the line will be given further attention. Risks that fall below the line are

FIGURE 35.3 Risk and management concern



reevaluated to accomplish second-order prioritization. Referring to Figure 35.3, risk impact and probability have a distinct influence on management concern. A risk factor that has a high impact but a very low probability of occurrence should not absorb a significant amount of management time. However, high-impact risks with moderate to high probability and low-impact risks with high probability should be carried forward into the risk analysis steps that follow.

All risks that lie above the cutoff line should be managed. The column labeled RMMM contains a pointer into a *risk mitigation, monitoring, and management plan* or, alternatively, a collection of risk information sheets developed for all risks that lie above the cutoff. The RMMM plan and risk information sheets are discussed in Sections 35.5 and 35.6.

Risk probability can be determined by making individual estimates and then developing a single consensus value. Although that approach is workable, more sophisticated techniques for determining risk probability have been developed (e.g., [McC09]).

Quote: "[Today,] no one has the luxury of getting to know a task so well that it holds no surprises, and surprises mean risk." Stephen Grey

35.4.2 Assessing Risk Impact

Three factors affect the consequences that are likely if a risk does occur: its nature, its scope, and its timing. The nature of the risk indicates the problems that are likely if it occurs. For example, a poorly defined external interface to customer hardware (a technical risk) will preclude early design and testing and will likely lead to system integration problems late in a project. The scope of a risk combines the severity (just how serious is it?) with its overall distribution (how

much of the project will be affected or how many stakeholders are harmed?). Finally, the timing of a risk considers when and for how long the impact will be felt. In most cases, you want the "bad news" to occur as soon as possible, but in some cases, the longer the delay, the better.

? How do we assess the consequences of a risk?

Returning once more to the risk analysis approach proposed by the U.S. Air Force [AFC88], you can apply the following steps to determine the overall consequences of a risk: (1) determine the average probability of occurrence value for each risk component; (2) using Figure 35.1, determine the impact for each component based on the criteria shown, and (3) complete the risk table and analyze the results as described in the preceding sections.

The overall *risk exposure*, RE, is determined using the following relationship [Hal98]:

$$RE = P \times C$$

where P is the probability of occurrence for a risk, and C is the cost to the project should the risk occur.

For example, assume that the software team defines a project risk in the following manner:

Risk identification. Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.

Risk probability. Eighty percent (likely).

Risk impact. Sixty reusable software components were planned. If only 70 percent can be used, 18 components would have to be developed from scratch (in addition to other custom software that has been scheduled for development). Since the average component is 100 LOC and local data indicate that the software engineering cost for each LOC is $14.00, the overall cost (impact) to develop the components would be 18 × 100 × 14 = $25,200.

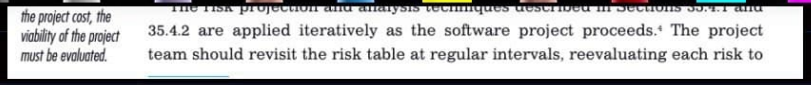
Risk exposure. RE = 0.80 × 25,200 ~ $20,200.

ADVICE Compare RE for all risks to the cost estimate for the project. If RE is greater than 50 percent of the project cost, the viability of the project must be evaluated.

Risk exposure can be computed for each risk in the risk table, once an estimate of the cost of the risk is made. The total risk exposure for all risks (above the cutoff in the risk table) can provide a means for adjusting the final cost estimate for a project. It can also be used to predict the probable increase in staff resources required at various points during the project schedule.

The risk projection and analysis techniques described in Sections 35.4.1 and 35.4.2 are applied iteratively as the software project proceeds.[4] The project team should revisit the risk table at regular intervals, reevaluating each risk to

⇒ Risk Mitigation, Monitoring and Management


the project cost, the viability of the project must be evaluated.

The risk projection and analysis techniques described in Sections 35.4.1 and 35.4.2 are applied iteratively as the software project proceeds.[4] The project team should revisit the risk table at regular intervals, reevaluating each risk to

⇒ Risk Mitigation, Monitoring and Management

⇒ Risk Mitigation

If the software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation.

⇒ Risk Monitoring

→ The project manager monitors factors that may provide an indication of whether the risk is becoming more or less likely.

→ In addition, product manager should monitor the effectiveness of risk mitigation steps.

⇒ Projects manager should manage work products carefully to ensure that each can stand on its own, agar ek jaaye to dusra aaram se kaam start kar paaye.

⇒ Risk Management

→ Risk management and contingency planning assumes that mitigation efforts have failed and that the risk has become a reality.

→ Agar koi chhod rha hai to, uska knowledge-transfer-mode me kaha for his last weeks

⇒ RMMM steps incur additional project cost.

→ Use Pareto's principle (80% risks from 20% identified risks).

⇒ The RMMM Plan

The RMMM Plan

RMMM Plan documents all work performed as a part of risk analysis and is used by the project manager as part of overall project plan.

Risk information sheet			
Risk ID: P02-4-32	Date: 5/9/09	Prob: 80%	Impact: high
Description: Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.			
Refinement/context: Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards. Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components. Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.			
Mitigation/monitoring: 1. Contact third party to determine conformance with design standards. 2. Press for interface standards completion; consider component structure when deciding on interface protocol. 3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.			
Management/contingency plan/trigger: *RE* computed to be $20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. Trigger: Mitigation steps unproductive as of 7/1/09.			
Current status: 5/12/09: Mitigation steps initiated.			
Originator: D. Gagne		Assigned: B. Laster	

I-37-CMMI

Tuesday, 31 May 2022 3:54 AM

Capability Maturity Model Integration [CMMI]

A comprehensive process meta-model that is predicted on a set of system and software engineering capabilities that should be present as organizations reach different levels of process capability and maturity.

→ CMMI represents a process meta-model in two different ways:

1) As a continuous model
2) As a staged model.

Each process area (e.g., project planning or requirements management) is formally assessed against specific goals and practices and is rated according to the following capability levels:

IPMDBO → IPSeNDBO?

Level 0: *Incomplete*— The process area (e.g., requirements management) is either not performed or does not achieve all goals and objectives defined by the CMMI for level 1 capability for the process area.

Level 1: *Performed*— All of the specific goals of the process area (as defined by the CMMI) have been satisfied. Work tasks required to produce defined work products are being conducted.

Level 2: *Managed*— All capability level 1 criteria have been satisfied. In addition, all work associated with the process area conforms to an organizationally defined policy; all people doing the work have access to adequate resources to get the job done; stakeholders are actively involved in the process area as required; all work tasks and work products are "monitored, controlled, and reviewed; and are evaluated for adherence to the process description" [CMM07].

Level 3: *Defined*— All capability level 2 criteria have been achieved. In addition, the process is "tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures, and other process-improvement information to the organizational process assets" [CMM07].

Level 4: *Quantitatively managed*— All capability level 3 criteria have been achieved. In addition, the process area is controlled and improved using measurement and quantitative assessment. "Quantitative objectives for quality and process performance are established and used as criteria in managing the process" [CMM07].

Level 5: *Optimized*— All capability level 4 criteria have been achieved. In addition, the process area is adapted and optimized using quantitative (statistical) means to meet changing customer needs and to continually improve the efficacy of the process area under consideration.

FIGURE 37.2
CMMI Process Area Capability Profile
Source: [Phi02].

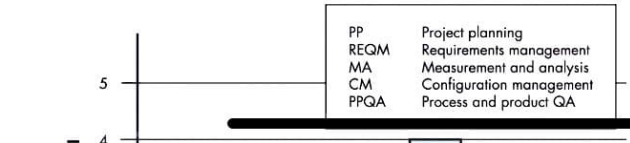
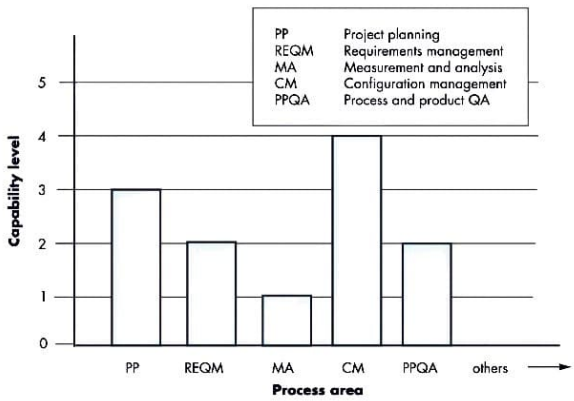


FIGURE 37.2
CMMI Process Area Capability Profile
Source: [Phi02].



The CMMI defines each process area in terms of "specific goals" and the "specific practices" required to achieve these goals. *Specific goals* establish the characteristics that must exist if the activities implied by a process area are to be effective. *Specific practices* refine a goal into a set of process-related activities.

For example, **project planning** is one of eight process areas defined by the CMMI for "project management" category.[7] The specific goals (SG) and the associated specific practices (SP) defined for **project planning** are [CMM07]:

SG 1 Establish Estimates
- SP 1.1-1 Estimate the Scope of the Project
- SP 1.2-1 Establish Estimates of Work Product and Task Attributes
- SP 1.3-1 Define Project Life Cycle
- SP 1.4-1 Determine Estimates of Effort and Cost

SG 2 Develop a Project Plan
- SP 2.1-1 Establish the Budget and Schedule
- SP 2.2-1 Identify Project Risks
- SP 2.3-1 Plan for Data Management
- SP 2.4-1 Plan for Project Resources
- SP 2.5-1 Plan for Needed Knowledge and Skills
- SP 2.6-1 Plan Stakeholder Involvement
- SP 2.7-1 Establish the Project Plan

SG 3 Obtain Commitment to the Plan
- SP 3.1-1 Review Plans That Affect the Project
- SP 3.2-1 Reconcile Work and Resource Levels
- SP 3.3-1 Obtain Plan Commitment

In addition to specific goals and practices, the CMMI also defines a set of five generic goals and related practices for each process area. Each of the five generic goals corresponds to one of the five capability levels. Hence, to achieve a particular capability level, the generic goal for that level and the generic practices that correspond to that goal must be achieved.

The staged CMMI model defines the same process areas, goals, and practices as the continuous model. The primary difference is that the staged model defines five maturity levels, rather than five capability levels. To achieve a maturity level, the specific goals and practices associated with a set of process areas must be achieved. The relationship between maturity levels and process areas is shown in Figure 37.3.

In addition to specific goals and practices, the CMMI also defines a set of five generic goals and related practices for each process area. Each of the five generic goals corresponds to one of the five capability levels. Hence, to achieve a particular capability level, the generic goal for that level and the generic practices that correspond to that goal must be achieved.

The staged CMMI model defines the same process areas, goals, and practices as the continuous model. The primary difference is that the staged model defines five maturity levels, rather than five capability levels. To achieve a maturity level, the specific goals and practices associated with a set of process areas must be achieved. The relationship between maturity levels and process areas is shown in Figure 37.3.

FIGURE 37.3

Process areas required to achieve a maturity level
Source: [Phi02].

Level	Focus	Process Areas
Optimizing	***Continuous process improvement***	**Organizational innovation and deployment** **Causal analysis and resolution**
Quantitatively managed	***Quantitative management***	**Organizational process performance** **Quantitative project management**
Defined	***Process standardization***	**Requirements development** **Technical solution** **Product integration** **Verification** **Validation** **Organizational process focus** **Organizational process definition** **Organizational training** **Integrated project management** **Integrated supplier management** **Risk management** **Decision analysis and resolution** **Organizational environment for integration** **Integrated teaming**
Managed	***Basic project management***	**Requirements management** **Project planning** **Project monitoring and control** **Supplier agreement management** **Measurement and analysis** **Process and product quality assurance** **Configuration management**
Performed		