**5.3 QUALITY ASSESSMENT AND IMPROVEMENT**

Various parallel and post-QA activities are carried out to close the quality engineering loop. The primary purpose of these activities is to provide quality assessment and feedback so that various management decisions, such as product release, can be made and possible quality and process improvement initiatives can be carried out.

The major activities in this category include:

* *Measurement:* Besides defect measurements collected during defect handling, which is typically carried out as part of the normal QA activities, various other measurements are typically needed for us to track the QA activities as well as for project management and various other purposes. These measurements provide the data input to subsequent analysis and modeling activities that provide feedback and useful information to manage software project and quality.
* *Analysis and modeling:* These activities analyze measurement data from software projects and fit them to analytical models that provide quantitative assessment of selected quality characteristics or sub-characteristics. Such models can help us obtain an objective assessment of the current product quality, accurate prediction of the future quality, and some models can also help us identify problematic areas.
* *Providing feedback and identifying improvement potentials:* Results from the above analysis and modeling activities can provide feedback to the quality engineering process to help us make project scheduling, resource allocation, and other management decisions. When problematic areas are identified by related models, appropriate remedial actions can be applied for quality and process improvement.
* *Follow-up activities:* Besides the immediate use of analysis and modeling results described above, various follow-up activities can be carried out to affect the long term quality and organizational performance. For example, if major changes are suggested for the quality engineering process or the software development process, they typically need to wait until the current process is finished to avoid unnecessary disturbance and risk to the current project.

**5.4 QUALITY ENGINEERING IN SOFTWARE PROCESSES**

The quality engineering process forms an integral part of the overall software engineering process, where other concerns, such as cost and schedule, are also considered and managed. Individual QA activities can be carried out and integrated into the software process. When we broaden our scope to quality engineering, it also covers pre

QA quality planning as well as the post-QA measurement and analysis activities carried out parallel to and after QA activities to provide feedback and other useful information. All these activities and the quality engineering process can be integrated into the overall software process as well.

Pre-QA quality planning can be an integral part of any project planning. For example, in the waterfall process, this is typically carried out in the phase for market analysis, requirement gathering, and product specification. Such activities also provide us with valuable information about quality expectations by target customers and users in the specific market segment a software vendor is prepared to compete in. Quality goals can be planned and set accordingly. Project planning typically includes decisions on languages, tools, and technologies to be used for the intended software product. It should be expanded to include 1) choices of specific QA strategies and 2) measurement and models to be used for monitoring the project progress and for providing feedback.

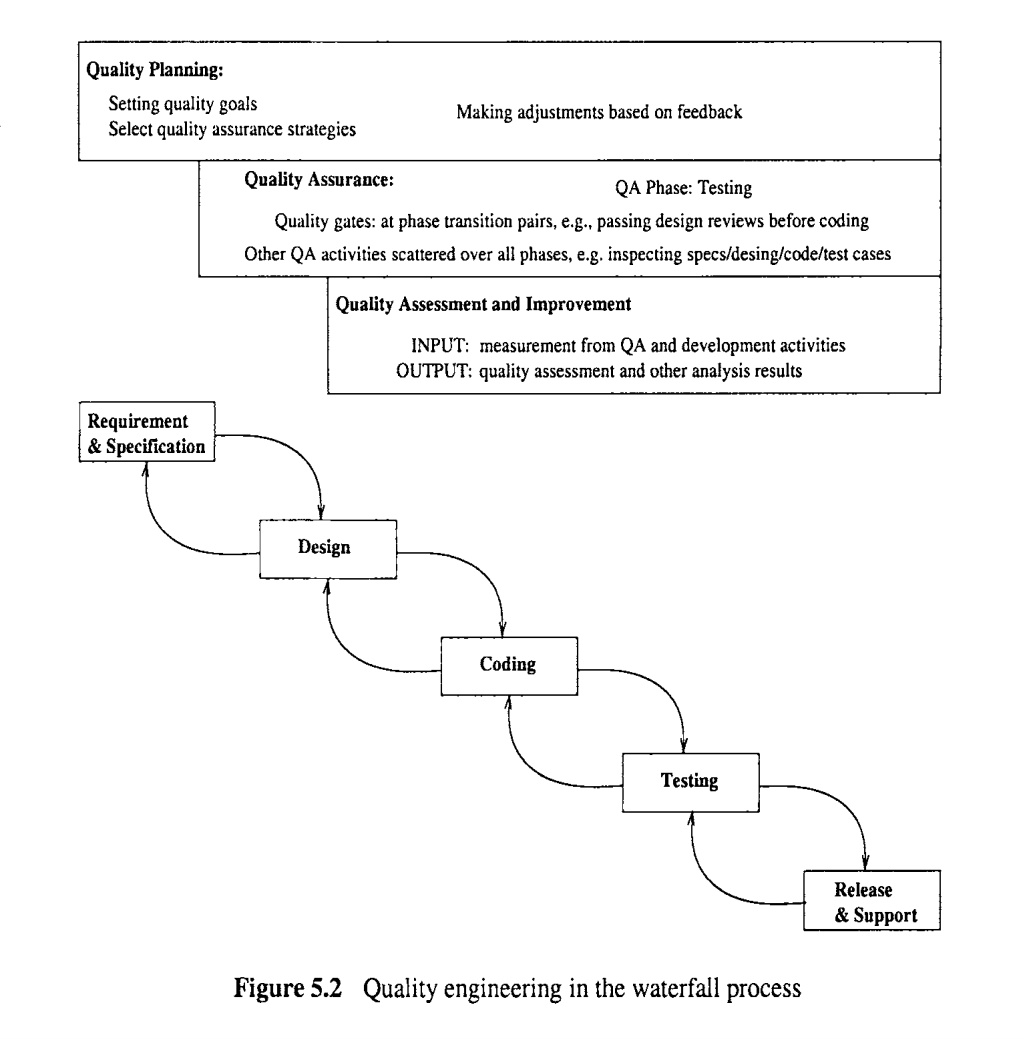
**In** alternative software processes other than waterfall, such as in incremental, iterative, spiral, and extreme programming processes, pre-QA activities play an even more active role, because they are not only carried out at the beginning of the whole project, but also at the beginning of each subpart or iteration due to the nature that each subpart includes more or less all the elements in the waterfall phases. Therefore, we need to set specific quality goals for each subpart, and choose appropriate QA activities, techniques, measurement, and models for each subpart. The overall quality goal may evolve from these sub-goals in an iterative fashion.

For normal project monitoring and management under any process, appropriate measurement activities need to be carried out to collect or extract data from the software process and related artifacts; analyses need to be performed on these data; and management decision can be made accordingly. On the one hand, the measurement activity cannot be carried out without the involvement of the software development team, either as part of the normal defect handling and project tracking activities, or as added activity to provide specific input to related analysis and modeling. Therefore, the measurement activities have to be handled “on-line” during the software development process, with some additional activities in information or measurement extraction carried out after the data collection and recording are completed.

On the other hand, much of the analysis and modeling activities could be done “off-line”, to minimize the possible disruption or disturbance to the normal software development process. However, timely feedback based on the results from such analyses and models is needed to make adjustments to the QA and to the development activities. Consequently, even such “off-line” activities need to be carried out in a timely fashion, but may be at a lower frequency. For example, in the implementation of testing tracking, measurement, reliability analysis, and feedback for IBM’s software products (Tian, 1996), dedicated quality analyst performed such analyses and modeling and provided weekly feedback to the testing team, while the data measurement and recording were carried out on a daily basis.

The specific analysis, feedback, and follow-up activities in the software quality engineering process fit well into the normal software management activities. Therefore, they can be considered as an integral part of software project management. Of course, the focus of these quality engineering activities is on the quality management, as compared to the overall project management that also includes managing project features, cost, schedule, and so on.

The integration of the quality engineering process into the waterfall software development process can be illustrated by Figure 5.2.



The horizontal activities roughly illustrate the timeline correspondence to software development activities. For example, quality planning starts right at the start of the requirement analysis phase, followed by the execution of the selected QA activities, and finally followed by the measurement and analysis activities. All these activities typically last over the whole development process, with different subactivities carried out in different phases. This is particularly true for the QA activities, with testing in the test phase, various reviews or inspections at the transition from one phase to its successor phase, and other QA activities scattered over other phases. Minor modifications are needed to integrate quality engineering activities into other development processes. However, the distribution of these activities and related effort is by no means uniform over the activities or over time, which is examined next.

**Effort profile**

Among the three major types of activities in the quality engineering process, the execution of specific QA activities is central to dealing with defects and assuring quality for the software products. Therefore, they should and normally do consume the most resources in terms of human effort as well as utilization of computing and other related resources. However, the effort distribution among the three is not constant over time because of the process characteristics described above and the shifting focus over time. Some key factors that affect and characterize the effort profile, or the effort distribution over time, include:

* Quality planning drives and should precede the other two groups of activities. Therefore, at the beginning part of product development, quality planning should be the dominant part of quality engineering activities. Thereafter, occasional adjustments to the quality goals and selected quality strategies might be applied, but only a small share of effort is needed.

The collective effort of selected QA activities generally demonstrates the following pattern:

- There is a gradual build-up process for individual QA activities and for them collectively.

- The collective effort normally peaks off a little bit before product release, when development activities wind down and testing activities are the dominant activities.

- Around product release and thereafter, the effort tapers off, typically with a sudden drop at product release.

Of course, the specific mix of selected QA activities as well as the specific development process used would affect the shape of this effort profile as well. But the general pattern is expected to hold. Measurement and quality assessment activities start after selected QA activities are well under way. Typically, at the early part of the development process, small amounts of such activities are carried out to monitor quality progress. But they are not expected to be used to make major management decisions such as product release. These activities peak off right before or at the product release, and lower gradually after that. In the overall shape and pattern, the effort profile for these activities follows that for the collective QA activities above, but with a time delay and a heavier load at the tail-end. One common adjustment to the above pattern is the time period after product release. Immediately after product release or after a time delay for market penetration, the initial wave of operational use by customers is typically accompanied by many user-reported problems, which include both legitimate failures and user errors. Consequently, there is typically an upswing of overall QA effort. New data and models are also called for, resulting in an upswing of measurement and analysis activities as well.

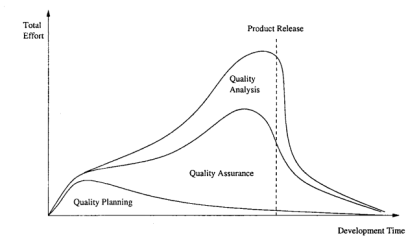
The main reason for this upswing is the difference between the environment where the product is tested under and the actual operational environment the product **is** subjected to.

The overall quality engineering effort over time is divided into three parts:

The bottom part represents the share of total effort by quality planning activities; The middle part represents the share of total effort for the execution of selected QA activities;

The upper part represents the share of total effort for the measurement and quality assessment activities.

Notice that this figure is for illustration purposes only. The exact profile based on real data would not be as smooth and would naturally show large amount of variability, with many small peaks and valleys. But the general shape and pattern should preserve.



In addition, the general shape and pattern of the profile such as in Figure 5.3 should preserve regardless of the specific development process used. Waterfall process would see more dominance of quality planning in the beginning, and dominance of testing near product release, and measurement and quality assessment activities peak right before product release.

Other development processes, such as incremental, iterative, spiral, and extreme programming processes, would be associated with curves that vary less between the peaks and valleys. QA is spread out more evenly in these processes than in the waterfall process, although it is still expected to peak a little bit before product release. Similarly, measurement and analysis activities are also spread out more evenly to monitor and assess each part or increment, with the cumulative modeling results used in product release decisions. There are also more adjustments and small-scale planning activities involved in quality planning, which also makes the corresponding profiles less variable as well.

To manage the quality assurance (QA) activities and to provide realistic opportunities of quantifiable quality improvement, we need to go beyond QA to perform the following:

*Quality planning* before specific QA activities are carried out, in the so-called pre-QA activities in software quality engineering. We need to set the overall quality goal by managing customer’s quality expectations under the project cost and budgetary constraints. We also need to select specific QA alternatives and techniques to implement as well as measurement and models to provide project monitoring and qualitative feedback.

*Qualily quantization and improvement* through measurement, analysis, feedback, and follow-up activities. These activities need to be carried out after the start of specific QA activities, in the so-called post-QA activities in software quality engineering.

The analyses would provide us with quantitative assessment of product quality, and identification of improvement opportunities. The follow-up actions would implement these quality and process improvement initiatives and help us achieve quantifiable quality improvement.