
CSQE Body of Knowledge areas:
Program and Project Management

The ability of an organization to accurately estimate tasks, build realistic schedules, and then meet those schedules is critical. Yet few organizations have demonstrated the ability to do this consistently. As a result, many software development and appraisal groups have little or no credibility when it comes to estimating and scheduling. Organizations may be deficient because of a lack of training, inability to manage commitments made to customers, poorly defined requirements, or lack of management support. Some organizations believe project management tools can improve their estimating and scheduling abilities. Unfortunately, tools cannot solve this problem—identifying and implementing best practices can. This article describes a technique based on team communication that has proved to be a useful method for estimating and scheduling the work required to develop and evaluate software products.

Key words: estimating, project management, scheduling, team communication, Wideband Delphi Method, Yellow Sticky Note Method

Creating Accurate Estimates and Realistic Schedules

STEVEN R. RAKITIN

Software Quality Consulting, Inc.

INTRODUCTION

The ability of an organization to produce accurate estimates, build realistic schedules, and then meet those schedules is critical. What happens when a software project is behind schedule? Often, features are dropped at the last minute, activities such as inspections are eliminated, and testing time is reduced. The result is a product that is delivered late, with fewer features than were promised, and often with far too many defects. When this happens, everyone loses:

- The company's *customers* lose because they may receive the software later than promised, without key features, and with more defects than were expected.
- The company's *employees* lose since no one wants to be associated with projects that are deemed failures.
- The *company* loses as its reputation suffers from not meeting customer commitments.

The objective of this article is twofold: 1) to provide insight into why estimates and schedules are usually wrong, and 2) to describe a technique that has been proven to provide more accurate estimates and realistic schedules.

WHY ARE ESTIMATES OFTEN WRONG?

From participating in and observing dozens of projects at dozens of organizations, the author has found several common

Creating Accurate Estimates and Realistic Schedules

factors that result in inaccurate estimates and unrealistic schedules. These include:

- *Companies play ridiculous negotiating games.* Project managers typically find themselves in a position where they must “negotiate” the project schedule with management. Frequently, this negotiation occurs at the beginning of a project, before all the requirements have been defined and all the variables identified, and it usually continues throughout the project, as key milestones are missed. These negotiating games, many of which have been identified by Thomsett (1996), are eloquently explained by Yourdon (1997). A few examples of these games are shown in Figure 1.
- *Companies over-commit and under-deliver.* Many organizations fail to manage commitments made to customers and frequently over-commit. When an organization over-commits, it cannot deliver what was promised when it was promised. Over-committing is caused by many factors, including competitive pressures; failure to consult with the development, quality assurance, and technical writing groups; over-eager sales people; and a lack of understanding of the implications of commitments. When the project manager realizes that the organization has over-committed, he or she is often forced to take drastic measures in an attempt to meet these unrealistic commitments and schedules. Features are cut, design reviews and inspections are cancelled, and testing is curtailed.
- *Projects start with a predetermined release date and without well-defined requirements.* All too often, a project begins with a predetermined release date that has been communicated to customers. The release date is often set before the requirements are clearly defined. It is not possible to commit to a realistic delivery date before the requirements are defined. This would be similar to building a house before the architectural drawings were done. Without these drawings, how can builders give you a realistic completion date? They cannot. Yet, people frequently ask software engineers, appraisal personnel, and technical writers to build, evaluate, and document a software product *without* well-defined requirements.

FIGURE 1 Yourdon's Negotiating Games [1]

Doubling and Adding Some	In this game, the project manager comes up with an estimate for the schedule and then doubles it. For good measure, a few extra weeks or months are then added in.
Reverse Doubling	Most managers are aware of the “doubling and adding some” game. They take the initial estimates from project managers and immediately cut them in half.
Spanish Inquisition	In this game, the project manager walks into a meeting unaware that he/she will be asked to provide management with an on-the-spot, instant estimate. Usually, the schedule has already been determined and the unwitting project manager is coerced into accepting it.
Low Bid	When outsourcing software, competitors often are encouraged to match or beat the competitor's schedule in order to win the contract. Of course, the competitor's schedule is not realistic, so the project manager must agree to match someone else's folly in order to get the contract.
“Guess the number I'm thinking of...”	Management has decided what an “acceptable” figure is for the schedule but doesn't reveal it. The project manager meets with management and attempts to guess what it is by starting with a realistic estimate and whittling it down until it reaches management's “acceptable” figure.

© 2002, ASQ

- *Tasks are estimated based on time available rather than time required.* When a project team commits to a delivery date before the requirements are defined, the project manager is forced to *schedule backward*—that is, create the schedule by starting from the release date and working backward to today. When this happens, tasks are estimated based on *how much time is available* rather than *how much time the task actually requires*. Estimating tasks based on time available rather than time required means that from the beginning of the project, the estimates are incorrect and the schedule is unrealistic.
- *Task interdependencies are not identified.* Software development projects frequently require several groups within the organization to work together. At a minimum, these groups include software engineers, quality assurance, and technical writers. Often, other groups may be involved as well. However, interdependencies between tasks are frequently ignored when scheduling backward. Why? Because the release date has already been set, and the interdependent groups have not been involved in producing the schedules.

A TYPICAL SCHEDULED-BACKWARD PROJECT

Consider what happens on a typical scheduled-backward project—a project that starts with a predetermined release date, usually resulting from one of the negotiating games listed in Figure 1, and lacks well-defined requirements.

Sooner or later, a critical task will be very late or, worse, will be unable to be completed. Then the ripple effect begins. The test plans, documentation, and coding take longer than expected because the tasks were never fully understood from the outset, dependencies between tasks were never identified, contingency plans for staffing were never implemented, and so forth.

When it becomes obvious that the project team will not meet the schedule, the project manager often abandons whatever development process the team was following in hopes this will somehow speed things up. The focus is shifted to paring down features and cranking up coding. Verification activities such as design reviews and inspections are now viewed as unnecessary and are eliminated. The time planned for validation testing is drastically cut, since testing is typically one of the last activities on the schedule. No design reviews, no inspections, less testing, and more hurrying often results in releasing an unfit product.

The product eventually gets released, usually weeks or months after the scheduled date. The project team is demoralized since it worked extremely hard to get the product released and knows that customers will not be satisfied because the product is missing key features and has far too many defects. The most amazing thing about this whole scenario is that no matter how many times this happens, management is still:

- Appalled at the high support costs
- Upset that so many defects were missed
- Quick to blame the team for doing shoddy work

Clearly, focusing *only* on time-to-market or *only* on quality or *only* on features to the *exclusion of the other two* is not desirable. Having a quality product delivered months late, and as a result, not sell can be just as bad as releasing a poor quality product on time. What is needed is a proper balance of quality, features, and schedule (Rakitin 2001).

YELLOW STICKY NOTE METHOD

The yellow sticky note method facilitates communication within the project team, and as a result, helps people develop more accurate, realistic estimates of tasks they will perform. It also includes identification of dependencies between tasks. By starting with more accurate estimates and including task dependencies, it is a more straightforward process to create a project schedule that is accurate and realistic.

This method is based on the following simple principles:

- Start with well-defined requirements.
- People who will be doing the work create estimates for the tasks they will do and then help build the schedule.
- Project team members critique each other's estimates.
- People are held accountable for meeting their commitments.
- The organization under-commits. Customers are promised less than what can realistically be delivered.
- Everyone is trained in the yellow sticky note method.
- Management has “bought into” the process and supports its use.

With these simple principles in mind, consider how to create accurate estimates and build realistic schedules.

Start with a Complete Requirements Specification

To create accurate estimates and build a realistic schedule, the project team must have a relatively complete description of what it is being asked to build, that is, a software requirements specification (SRS) (IEEE 1998). Having a relatively complete SRS is essential regardless of what method is used to create estimates and build schedules.

The author has taught this to thousands of people. In doing so, it never fails that when he mentions that a relatively complete SRS is required, people laugh. People often say, “We never start projects with an SRS. We're lucky if the SRS ever gets written!”

So the first step in learning how to create accurate estimates and schedules is to start with an SRS. At this point, it would be helpful to plan a requirements review or inspection. This way, problems with the requirements can be worked out early and problems avoided. Developers, quality assurance, and technical writers should be invited to participate and should each bring his or her unique perspective. That is:

- Developers should ask, is each requirement implementable?
- Quality assurance should ask, is each requirement verifiable?
- Technical writers should ask, is each requirement explainable in simple terms?

Once the requirements are reviewed, they then can be grouped.

Group Requirements into Must-Haves and Wants

Once the SRS is written and reviewed, the requirements must then be grouped into *must-haves* and *wants*.

- *Must-haves*: the product is not worth introducing if it lacks these features
- *Wants*: features that customers want but could be put into a future release if necessary

Frequently, marketing people are involved in making these decisions since they are supposed to be in close contact with customers and should be aware of customers' needs.

Suppose that marketing goes through the SRS and determines that all the requirements are in the must-have group. This is not an acceptable answer, since team communication depends on some requirements being more important than others. If this happens, then use the no-tie ranking method to force a ranking of requirements. Each requirement is ranked according to importance to the customer from 1 to N, where N is the number of unique requirements. The team then determines that requirements numbered 1 through M (where $M < N$) are must-haves and all the rest are wants.

Commit to Deliver Only the Must-Haves Not the Wants

The problem that many organizations have is they over-commit—they promise customers more than can be delivered in the promised time frame. When using the

yellow sticky note method, management commits to deliver *only the must-haves not the wants*. Customers are not told about the wants. By committing to deliver only the must-haves, management is setting the customer's expectation lower so that it is more likely to be met. Setting the bar too high and consistently failing to meet it causes customers to become dissatisfied.

In developing the schedule, the project team plans to deliver a product that contains all the requirements—must-haves *and* wants. To account for those unexpected things that happen on every project and for overly optimistic estimates, the project manager works with the project team to try to get back on schedule. If that is not possible, then, as a last resort, the project manager can choose to drop one of the wants. This way, in the worst case the team delivers exactly what was promised (that is, just the must-haves), and in the best case the team delivers more than was promised (that is, must-haves and some or all of the wants). By following this approach, organizations can *under-commit* and *over-deliver*.

YELLOW STICKY NOTE RULES

There are a few simple rules that must be followed when preparing task estimates and building schedules. These are:

- *The people doing the work identify and estimate their tasks based on the SRS.*
- *Each task should be small.* A larger number of shorter tasks is preferable to a smaller number of longer tasks. Decomposing overly long tasks into smaller subtasks provides better visibility and flexibility in building and managing the schedule. Each task should take five working days or less to complete.
- *When building the schedule, use the 80 percent rule.* In a given week, most people do not have 40 hours to apply to project tasks. At most, people have 80 percent of that or 32 hours. This is because people spend time in meetings, attend training classes, talk on the phone, surf the Web, and participate in other nonproject-related activities. Some people may actually have less than 80 percent to apply to a project because they may be working on two or three projects at the same time.
- *Include vacation, holidays, and tradeshow on the schedule.*

Identifying Tasks and Creating the Initial Estimates

Once the project team has been trained, team members review the SRS. This review is conducted either individually (on smaller projects) or as a group within group members' own disciplines (on larger projects).

The purpose of this review is to identify every task that is required to develop, evaluate, and document the product and determine who will be responsible for that task. Note that tasks are identified for all requirements, both the must-haves and wants.

After the tasks are identified, the person responsible for that task estimates how long (in days) it would take him or her to complete the task *assuming he or she could work on that task uninterrupted*. The 80 percent rule is not used to determine how long a task takes but rather, when the task will complete, as discussed next.

If the task is something that the organization has never done, use the Wideband Delphi Method (Rakitin 2001) to develop a reasonable estimate. Once the task duration is estimated, each person then identifies the dependencies for starting this task. That is, what other task must be completed before this task can start.

All of this information is written onto a Post-It™ note—commonly referred to as a yellow sticky note—as illustrated in Figure 2. Different groups on the project team should use different color sticky notes so they can be visually distinguished. Others have reported using similar mechanisms, such as index cards and push pins (Phillips 2001). The important concepts are

that individuals responsible for doing the work estimate the task and that the schedule is built going forward.

Each person goes through the process of completing an appropriately colored sticky note with the information shown in Figure 2 for each task he or she has been assigned. The understanding with respect to the task estimates is that *each person is making a personal commitment to complete that task in that amount of time*. Because of this personal commitment, there is immediate buy-in to the schedule from each member of the project team. Each person understands that he or she will be held accountable for meeting his or her commitments.

It is a good idea to identify on each sticky note whether this task is related to a must-have or a want. (An “M” or “W” in the corner works well.) When everyone is finished with this part of the process, the project manager schedules a schedule-building session.

BUILDING THE SCHEDULE GOING FORWARD

The first schedule-building session includes the entire project team and should occur at an off-site location. The project manager should coordinate this activity. The room used for this activity should have a long wall upon which some plain chart paper can be affixed. Week marks (not dates) are indicated along the top of the chart paper.

The project team brings all of its colored sticky notes to the session. The process of building the schedule going forward is based on each task having at least one dependent task. When the project team is ready, team members approach the chart paper with their sticky notes and place them on the chart in the location where the task should complete. Here is where the 80 percent rule is applied. Say one has identified a task that should take five days (40 hours) to complete if not interrupted. By applying the 80 percent rule, a five-day task requires six working days to complete. So the sticky note is placed on the sixth day after it can begin. Refer to Figure 3 for an example of what a schedule might look like.

Now this is where it gets interesting and exciting. Recall that the whole project team is in the room. Each person is now standing in front of this wall with a handful of colored sticky notes. As the sticky notes start to go up, discussions start happening. Quality assurance peo-

FIGURE 2 "Sticky note" information requirements

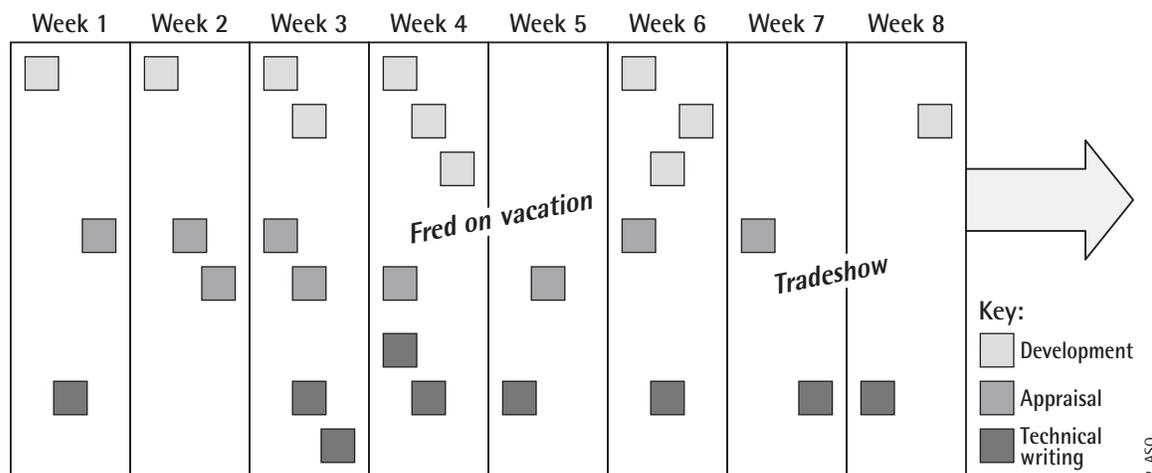
A rectangular sticky note with a black border and a light gray drop shadow. It contains four lines of text, each with a bold label followed by a colon:

- Name:
- Task:
- Duration:
- Dependencies:

© 2002, ASQ

Creating Accurate Estimates and Realistic Schedules

FIGURE 3 Building the schedule going forward



ple talk to developers, developers talk to technical writers, everyone talks to the project manager. In addition, peers review each other's estimates. For example, a developer might put up a sticky note for a task with an estimate of three days. Another, perhaps more experienced developer will critique that estimate and say, "You know, I did a task similar to that on the last project and it took me eight days not three." Thus, with peers critiquing each other's estimates, the *estimates get better*.

Further, by working together to build the schedule, the project team will often identify several tasks that no one had thought of previously. This aspect of the process serves to ensure that nothing is overlooked and that an accurate and realistic schedule is created.

When placing tasks on the chart, the team tries to place all of the tasks related to must-haves first and the tasks related to wants last. The reason for this will be apparent in the next discussion.

Finally, the team looks at the schedule it is building. The fact that the team owns the schedule is a powerful motivator. Because team members are talking with each other, sticky notes are put up and pulled off many times until the team as a whole is satisfied that they have developed the best possible schedule.

Negotiate Based on Factual Information

Once the project team completes the task of building the schedule going forward, it will be able to tell manage-

ment with certainty when it can deliver the product. Frequently, management will not be pleased with this date and a negotiation session is usually required. Management will want to know how the delivery date can be pulled in. By looking at the chart with all of the colored sticky notes, the answers to this question should be obvious. Management can:

- Change the requirements (and the commitment to customers)
- Add resources

Two points need to be highlighted about this scenario. First, it is the people who will be doing the work telling management when they can deliver and, second, the negotiation can be conducted with factual information that everyone can see. For example, the chart will clearly highlight any resource bottlenecks (as evidenced by the lack of certain colored sticky notes in an area of the chart). The chart also has the estimates for each task.

During the negotiation, tradeoffs will be made between features, quality, and schedule. These negotiations can now be conducted with factual information that can result in an informed business decision being reached.

Once the negotiation is completed and everyone (management and the project team) is in agreement on the schedule, the information on the chart can be entered into the project manager's favorite project management tool.

Manage the Project to the Schedule

Once the information from the chart is entered into a project management tool, the project manager needs to manage the project to the schedule. This means that when a task is behind schedule, the result is not a schedule slip. By scheduling forward, the project manager now has the following options:

- Work with the individual to understand if he or she can recoup lost time by working weekends or extra hours or by rearranging other tasks.
- Work with management to identify additional resources.
- As a last resort, the project manager can decide to drop off a want (but not a must-have) to help keep the project on schedule.

Here then, is the reason why it is so important at the outset that the project team is able to categorize requirements as either must-haves or wants. The project manager must remain on top of the situation on a daily basis, meeting with team members and tracking progress for each task on the schedule. As the project progresses, the project manager works with the team to ensure the must-haves are delivered according to the agreed-upon schedule.

SUMMARY

The track record of the software industry with respect to meeting schedules is abysmal. To change the performance of an organization, management must recognize that:

- Continuing to use the same mechanism for estimating and scheduling will result in continued inability to meet schedules.
- The organization needs to find an estimating and scheduling process that works.
- People who will be doing the work are best able to estimate how long the work will take.
- The organization needs training in good estimating and scheduling practices.
- People doing the work need the ability to set the schedule and then need to be held accountable for meeting it.

- The way to improve estimating skills is to learn from past estimates.
- People who create accurate estimates and schedules and then meet them should be rewarded.

By adhering to this process, dozens of project teams have been able to develop better, more accurate estimates and build accurate, more realistic schedules. Using this method, the project manager has more ability to help the project team meet its commitments to the organization and to customers.

Management plays a key role in helping to create a culture based on developing accurate, realistic schedules and then meeting them. By encouraging the organization to use proven methods such as team communication, management will improve efficiency, productivity, morale, customer satisfaction, and the bottom line.

REFERENCES

- IEEE. 1998. ANSI/IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications. New York: IEEE.
- Phillips, D. 2001. Cards-on-the-Wall Session. *Software Development* (July): 53-56.
- Rakitin, S. 2001. *Software verification and validation for practitioners and managers*. Norwood, Mass.: Artech House.
- Thomsett, R. 1996. Double dummy split and other estimating games. *American Programmer* (June).
- Yourdon, E. 1997. *Death march: The complete software developer's guide to surviving 'mission impossible' projects*. Englewood Cliffs, N. J.: Prentice-Hall.

BIOGRAPHY

Steven R. Rakitin is president of Software Consulting Inc. where he helps companies establish a more predictable software development process. He has more than 25 years of experience as a software engineer and software quality manager in a broad range of industries. He has written several papers on the subject of software quality and published a book titled, *Software Verification and Validation for Practitioners and Managers*. Rakitin has a bachelor's degree from Northeastern University and a master's degree from Rensselaer Polytechnic Institute. He has earned certifications from ASQ as a software quality engineer and quality auditor. He is a member of the IEEE Computer Society, the ASQ Software Division, and is on the editorial review board for *Software Quality Professional*. He presents tutorials and workshops at conferences nationally. He can be reached at info@swqual.com.