

analysis is performed by people who have the best understanding of the selected defect or problem being studied.

### 4.4.3 Ready, Set, Go!

Before the technology solution goes live and is available to the users, the project team needs to make sure that all involved parties are on the same page. For this, communication throughout the project is critical.

**TIP** Engage users early and often. There is no substitute for real users.

**Steve:** All the way through a project, you want to take every opportunity you can to validate with the customer and supplier that you're on track. But with all the preparation, it still comes down to the successful launch and full deployment of the new capability. Your customer satisfaction, a large chunk of your supplier's payment, and your professional ego are wrapped up in the pivotal events around going live with your product. It's like you've gone through all the rehearsals, and now it's time to perform.

Consistently delivering a successful premiere—seamlessly inserting new or enhanced technology solutions into your environment—is not without its perils. For instance, you don't want your project to be the one in which UAT came to mean “user angry at technology” instead of “user acceptance testing.”

**TIP** Throughout the project, frequently expose the technology solution to its intended environment.

**George:** Surprises during user acceptance testing aren't pretty. Let me tell you a story. It was in our validation lab for one of our largest automotive clients. We had pushbuttons, industrial controllers, and screens mounted on a table instead of being mounted on a fork truck. The pushbutton and the screen are about three meters apart—the user can walk between the two just like they would in their work environment. So my team tested the system. They'd push the button and then they'd walk over to the screen to see if, for instance, a material request would show up. Then they'd clear the material request, walk back over to the pushbutton, push it again, get another material request, walk over to the screen, and so on and so forth. No problems.

So in comes the user, a woman from one of the customer's sites. She pushes the button, walks over to the screen, sees that the message came through, clears the screen, walks back to the button, and then hammers

it mercilessly—bang, bang, bang, bang, bang—just as fast as she can. The button gives. It breaks. There's nothing coming through on the screen anymore, the light starts flashing, basically “smoke” comes out of the system.

And my guy says to her, “What did you do *that* for? Who would ever do something like that?” And she says, “That's going to happen. Someone's going to get frustrated, the trucks aren't going to come, so someone is just going to start hammering this button.”

My guys had never tested this situation, which was very embarrassing for me. Within ten minutes the user could come in and break the system. It's probably one of the worst UAT stories I could tell. It took us a while to live that one down.

Unfortunately, these UAT surprises are common. Steve adds his favorite one.

**Steve:** Before I started here, I worked for a cement factory. My team went through this elaborate, sophisticated effort to create a new system to monitor the ingredients to make the best cement there is. They were so confident that this would turn out to be a home run.

So when they had the great unveiling in the cement factory, the first worker that walked up to the system—the system had a fancy touch screen—tries to start the ingredient analysis. But he can't use the touch screen! How come? Because a safety regulation requires all workers to wear big gloves—think giant oven mitts. He couldn't manage to hit even one button on the touch screen. And of course, using the touch screen was the only way a user could steer the system.

A key ingredient for successful validation of an acquired product is to have the true customer and users involved throughout the project life cycle. This is especially challenging on projects that have long life cycles.

**TIP** Define skill and capability profiles of the users who will participate in user acceptance testing.

**George:** You need one kind of user, and that's the right kind. But what happens a lot of times in UAT, the acquisition team or their management might say, “Oh, that's seven months from now. I don't know who'll be available in seven months.” So the acquirer's side doesn't plan who to send to UAT because they don't know what else they're going to do. So, the week before UAT comes around, and they look around and they say, “Hey, John. You and Susie standing by the water cooler, you're going to UAT next week.”

**Steve:** Well, I've never done that, but I can see how it happens.

**George:** It never needs to happen. You write up a profile that captures the agreed-upon skills and capabilities of a potential test user. Okay? You can figure out, do they meet that profile, before folks are asked to participate in UAT. Otherwise, if you're doing UAT that requires domain knowledge—and in my experience, many products require this for successful UAT—you get inaccurate feedback without the right users, and you'll spend a lot of money before you get it right.

George pauses for a moment and then continues.

**George:** Let's say you're validating a technology solution for the product development group. You're building something that supports structural engineering. You couldn't just send a random person who didn't know anything about this area, or knew enough about it but didn't understand that 4,000 metric tons per square inch is a bad answer for the force it takes to open a door.

So, when you find out that the UAT users you have don't fit the profile, you're stuck. You're looking at them and you have to say, "You're the wrong person for this assignment," which of course is insulting. So they push back: "Well, I work in this group, I know how the work gets done in this group." Well, they do, but they don't know this specific piece. So how can they help us validate the proposed solution if they don't know the work it will support?

So this is the biggest lesson learned. Ultimately you have to get the right people in the room or in the field to spend the time giving you meaningful feedback.

What can you do to avoid UAT surprises and increase your chances for a smooth go-live of the technology solution? It's vital to have a time-tested, proactive set of verification procedures and criteria to ensure that the product or component will fulfill its intended use when placed in its intended environment. You need to clearly identify the applicable verification procedures and criteria and then reference these procedures in the solicitation package and supplier agreement.

**TIP** Insist on rigorous, time-tested validation steps. Enforce them contractually without exception.

**George:** What we've created is essentially a manufacturing process for deploying technology solutions. We want to make sure we go live with a solution and live to tell about it. While we want to instill creativity when designing and developing the solution, we encourage all partici-

pants to rigorously stay within the deployment script. Everybody must know their roles and perform according to procedure. So even if you find things during deployment that will make the solution a little bit better, you have to be really careful about whether you'll change the deployment process to make an adjustment. Feature creep is bad enough, but during deployment it's downright dangerous.

**Paula:** You're so right. I can't tell you how many times we find errors in last-minute tweaking.

**George:** Now we're at a point where our deployment process is getting to be almost like how some automotive companies launch a new vehicle: Engineering will evolve the design, and then finally the design is ready for production. When they bring it to a manufacturing plant, the design is done. It's been tested, and we know it works. Then, we'll take a vehicle down the manufacturing line to go through the manufacturing process one last time just to verify that the new model will build properly and that our process works. So, eventually, deployment of technology solutions is like a vehicle launch, in that we pilot the technology solution to verify that the build process works and that the deployment process works. At that point we can confidently say that we can do the deployment process over and over again. Standardizing becomes even more important, since we've got multiple teams, dispersed globally, deploying the same system.

Steve strongly supports George's ideas.

**TIP** Maximize setup work for deploying technology solutions.

**Steve:** A tightly controlled deployment process is critical. We're better off doing a staging operation rather than doing an "out of the box" installation at the customer's site. What we do is, we source all the materials, we take it to an assembly site, rack all the hardware components, load the software, do some elementary testing. Then we'll wire it all up and connect it to the power and network at the site. This focuses the site team on two items. One, all they have to set up are those pieces we couldn't preassemble. Two, they can pay close attention to how to migrate from the old product to the new one.

**George:** That sounds like a great plan.

**Steve:** I think it's one of the cleverest things we've come up with in a long time. We used to see the same mistakes over and over again, but the common denominator was somebody new was doing something not in the process. So we want to try to have as much repetitive stability as possible.

The closer you get to going live with the new or enhanced product, the more important it is to train the potential users and support personnel to use it. Effective training requires assessment of needs, planning, instructional design, and appropriate training media (e.g., workbooks, software) as well as a repository of training process data. As an organizational process, training has these main components: a managed training development program, documented plans, personnel with appropriate mastery of specific disciplines and other areas of knowledge, and mechanisms for measuring the effectiveness of the training.

**TIP** Thoroughly test and pilot your training materials.

**TIP** For the most effective training, combine different types and delivery mechanisms.

**Paula:** At minimum, we want people to be able to do their jobs without disruption on the day the technology solution goes live. One thing about training is that our primary motto, and we're fairly rigorous on this, is that during development the supplier creates the training package. It's tested as part of development and verification of the solution.

**George:** It sounds like you've done a lot with training.

**Paula:** We've used different kinds of training, everything from one-on-one training, on-the-job training, classroom training, multimedia training, and more. We now use a lot of Web-based training. We try to have this in place a short time before go-live so that the users are ready on day one. But then again, sometimes we may supplement that with face-to-face sessions where we call the people in and walk them through the latest changes in detail. This has the advantage of allowing for a dialog between the instructor and the users. Then there are a significant number of technology solutions that we've got where the business process changes so that the user training is actually embedded in some process retraining.

When the big go-live day finally arrives, acquirers can only hope that the process plays out like the Vienna Philharmonic performing Strauss at New Year's—breathtaking, flawless, and smiles on everyone's faces when it's finished.

**TIP** Open mandatory communication channels for all stakeholders during go-live of the technology solution.

**George:** Once we start the countdown to go-live, we have a whole process detailed to carry it out. Before we go live, we collect all the infor-

mation; we have very detailed cut-over plans—minute by minute—that we have done in exhaustive detail and reviewed. So a great deal of work goes into that. Part of it is to make everything very visible.

**Steve:** Tell us more.

**George:** So, we have an intense focus on the communication. It's all about communication. It really is critical—communication of the right things at the right time. Where are the key checkpoints? Where will what testing be done? All this and more, to make sure the deployment is moving as expected. One of the problems I have with people is when they say, "Well, I'm going to use the new solution for 24 hours, and then I'll find out if things are okay or not." And you just don't want to do that. You want to be able to verify and validate that things are working as you expected along the way.

**Steve:** So what's in place on the day itself?

**George:** We have all the various players, every supplier that is involved, their technical personnel, available for immediate contact—if they don't have to be on-site already—and then a senior management person who can be contacted in the middle of the night if anything is needed from their company. Then when we actually start the system, we give an update on an open phone line every hour. People can just call in and ask questions or listen to the hourly update. At a point in time in the deployment we say, "Is everything ready? Do we turn the final switch?" There is a meeting, and that's when you get a whole group of people together for that final decision to go live.

**TIP** Contractually establish that any significant error found during warranty causes the warranty period to start over.

With the technology solution ready for use, you transfer responsibility for operations and support from the development supplier to the appropriate group. This could be the same supplier or an in-house team on your side. The development supplier, however, is still very much responsible for successfully completing the warranty period for the technology solution. Steve stresses the importance of the warranty period.

**Steve:** In my experience, with so many different components working together, so many pieces and parts, the warranty period is a lot more important than it used to be. There are so many things that can go wrong. It's not cost-effective for the supplier to test to the degree that you would have to in order to prevent anything from happening after

the solution is installed. So I think the warranty is really important. You hopefully don't have any major errors or outages, but it's always possible. If something happens, we hold our suppliers accountable. They're contractually required to fix any errors that occur during their warranty period. We also insist that any significant error found resets the warranty period, among other penalties. So it's in our mutual interest to arrive at a high-quality product quickly.

After the warranty period is complete and the responsibility for the technology solution is transferred to the operational and support organizations, you review and analyze the results of the transition activities and determine whether any corrective actions must be completed before closing the project.

#### 4.4.4 Applying the CMMI-ACQ to Validating the Product

The CMMI-ACQ provides specific recommendations on how to demonstrate that an acquired product or service fulfills its intended use when placed in its intended environment (see Table 4-18). Validation demonstrates that the acquired product, as provided, will fulfill its intended use. In other words, validation ensures that the product meets the stakeholders' intentions and the customer requirements.

Preparation activities include selecting products and components for validation and establishing and maintaining the validation environment,

**Table 4-18** *Acquisition Validation Solution Goal 1: Practices and Tips*

CMMI-ACQ Process Area: Acquisition Validation	
1. Goal: Preparation for validation is conducted.	
Practices	Tips
1.1 Select products or services to be validated and the validation methods that will be used for each.	<ul style="list-style-type: none"> <li>• Throughout the project, frequently expose the technology solution to its intended environment.</li> <li>• Define skill and capability profiles of the users who participate in user acceptance testing.</li> </ul>
1.2 Establish and maintain the environment needed to support validation.	<ul style="list-style-type: none"> <li>• Maximize setup work for deploying technology solutions.</li> </ul>
1.3 Establish and maintain procedures and criteria for validation.	<ul style="list-style-type: none"> <li>• For most effective training, combine different types and delivery mechanisms.</li> </ul>