

Agile Risks/Agile Rewards

With lean proactive techniques for identification, analysis, prioritization, planning and monitoring, you can make risk management as agile as it is effective. BY PRESTON G. SMITH AND ROMAN PICHLER

MOST AGILE DEVELOPMENT METHODS claim to be risk driven. Accordingly, project risk should be your foremost concern when deciding how to proceed with a project and assign product features to particular iterations—remember, the riskiest ones should usually come first. Despite this tenet, many books available on agile methods—XP or Scrum, for instance—

into a risk-driven agile iteration lasting only seven to 30 days? This may sound like an impossible task, but it's not. Here's how to compress the essence of more conventional risk-management methods into the timescale of agile development.

The Basics

First, we should be clear about the nature of risk. A risk has three essential components: uncertainty, loss and finite duration. There's always uncertainty as to whether a risk will occur. If an item is certain to occur, we instead call it an *issue*. Issues are just as important as risks, but we manage them differently. Because a risk may not occur, we have a powerful means of managing it; namely, preventing it from occurring. We lack this option for issues.

Risk always involves loss: loss of time, loss of money, loss of market share or some other loss that the project will suffer if the risk occurs.

The combination of the likelihood that the risk will occur and the subsequent loss involved determines the risk's severity.

Finally, most risks have a temporal component that can be expressed as time or as a condition that causes the risk to expire. Consider, for example, the risk that "the new hardware platform won't be ready in time for software integration testing." Once the new platform is ready, the risk vanishes.

Effective risk management involves:

1. Identifying the risk.
2. Analyzing each risk to determine its severity.
3. Prioritizing the identified risks based on their severity.
4. Creating action plans (responses) to deal with the high-priority risks.
5. Continuous monitoring and follow-up to ensure that your action plans are mitigating the risks.

Conventional project risk management techniques all provide variations on these five steps, but each step is necessary—in some form—for effective risk management. The rub is that for agile risk management, we must complete these steps in far less time and effort than the conventional process allows. As with



At the beginning of the risk management session, each Siemens team member wrote the most critical risks perceived on index cards.

have remarkably little to say about how a development team determines the risks it faces, prioritizes them, or takes action to negate their effects.

The project management literature, on the other hand, offers countless books on managing such project risks, but many of these risk management practices are far from agile. Thus, the dilemma: How can you effectively tailor conventional risk-management approaches meant for years-long projects

Preston G. Smith, a consultant and trainer, helps managers accelerate their new product development. He's coauthor of *Developing Products in Half the Time* (Wiley, 1997; 2nd edition) and *Proactive Risk Management* (Productivity Press, 2002). **Roman Pichler** works as a development consultant at Siemens, Corporate Technology. He's currently helping Siemens Communications pilot an agile approach consisting of the Unified Process, Scrum and XP.

most processes, however, we can shorten a step dramatically once we understand its purpose, but it's unwise to skip an essential step.

Most project teams try to manage the risk itself, but we're more successful when we dig under the risk to find the facts in the project environment in which the risk lies. We call these underlying facts *drivers*. Because risks tend to be subjective, obtaining team agreement regarding the risk can be difficult; for example, Susan may think the risk is disastrous, while Kevin believes it's tolerable. We can reach agreement faster on the underlying facts.

Drivers are also useful when we formulate action plans for mitigating the risk. Drivers are usually root causes, so once we understand them, they naturally suggest effective action plans. For instance, consider the following risk that was resolved by exposing a driver:

► **Risk:** Marketing can't decide whether the Help notes should be in landscape or portrait layout.



Each team member presented his risks to the team, before pinning his risk cards on the wall.

- **Driver:** Marketing has typically vacillated on such issues before until well into integration testing.
- **Action plan:** Provide a radio button option for landscape or portrait Help notes.

Another effective risk management technique is to divide each risk into its risk event and associated impact. For example, a risk event might be "Frequency buckets required by customer may not align with those in our database," with an associated impact of "Extra programming to reallocate frequencies." By separating the risk event from its

impact, we can formulate prevention plans—which keep the risk from happening—that are separate from contingency plans, which reduce the damage if the risk does occur.

Because these techniques are essential to effective risk management, they apply to conventional and agile projects alike. The key to making the risk management process agile is to exploit the strengths of the agile method, such as the use of a dedicated team, colocation, close cooperation with the customer, and fast

feedback on changes. For example, in the conventional approach, the team goes through a lengthy analysis step (step two of the five-step process listed above). They normally complete this step for all identified risks so that they can use the analysis results to prioritize their risks quantitatively, based on a calculated risk severity, in step three.

In the agile approach, we take advantage of the team's strength and knowledge by having them analyze and prioritize the current risks, using *only their perceptions* to determine the severity of each risk. In a flash, this

Sample Agile Project Risks

In managing project risks on various projects, we found that risks tend to occur in the following areas. For a different project in a different environment, this list will be less pertinent, but you can create one like it for your project by categorizing risks as they occur.

Product Definition

- ▷ Product managers lack experience in creating user stories.
- ▷ User interface requirements currently unclear.
- ▷ Regional markets not yet decided.

Development

- ▷ Technology issues such as remote communication, data access and component frameworks.
- ▷ Product managers located remotely.
- ▷ Some team members not assigned full time.
- ▷ Communication issues due to language barriers.

Quality

- ▷ Variations in applying the unit test framework.
- ▷ Providing functionality competes with delivering high quality.

Environment

- ▷ Inadequate modeling tool for applying agile software engineering practices, such as test-driven development.
- ▷ Software production unstable due to differences between developer-build and production-build environments.
- ▷ Test hardware not available in time.

Sales and Distribution

- ▷ Launch programs not yet defined.
- ▷ Sales staff training measures open.
- ▷ User and service documentation may not be translated into all target languages in time.

—PS and RP

avoids hours of detailed quantitative analysis. And it's good enough: If they misjudge, their short iterations and strong feedback allow them to reprioritize the risks in the next iteration when they have fresher information from which to work.

Applying Agile Risk Management

In an agile risk-driven approach, the team performs risk management activities before starting development work within an iteration. Agile risk management practices must address two challenges: First, they must successfully integrate risk management activities into the iteration planning activities. Second, they must adapt risk management practices so that the entire team can perform them quickly.

We've successfully applied the risk management practices described here to a software development project in Siemens' telecommunications division. The project employs Scrum project management practices. Notice that these risk management practices can be adapted easily to other agile methods. In fact, adopting an agile and risk-driven approach helped us to find issues early on, such as employing an inappropriate communication mechanism, and to recover from them quickly. (For more details, see "Sample Agile Project Risks," page 51.)

Agile Risk Management Illustrated

We follow these steps during the general iteration planning process:

- ▶ Review the product backlog items (product features).
- ▶ Perform risk identification, analysis and prioritization.
- ▶ Perform risk response activities by mapping the risks identified onto the backlog.
- ▶ Select those parts of the backlog

Further Reading

- ▶ Want to know more about Scrum? Check out Ken Schwaber's book *Agile Project Management with Scrum* (Microsoft Press, 2004).
- ▶ For more information on project risk management, including detailed discussions of risk components, environments, drivers and risk management techniques, see Preston G. Smith and Guy M. Merritt's *Proactive Risk Management* (Productivity Press, 2002).
- ▶ Kent Beck and Martin Fowler's *Planning Extreme Programming* (Addison-Wesley, 2000) overviews XP and iteration planning.
- ▶ For more information on participatory decision-making, see *The Facilitator's Guide to Participatory Decision-Making* by Sam Kaner et al. (New Society Publishers, 1996).

associated with critical risks as work item candidates.

- ▶ Derive the iteration goal.

Before we start performing the risk management activities, we review the product backlog, a list of prioritized functional and nonfunctional requirements that we must implement and test to successfully deliver the product, including new or changed requirements. The entire



The team analyzed the risks by pinning cards with related risks next to each other, forming cohesive risk groups. Each risk group was given a name; for example, "Communication."

team takes part in the risk management activities, with the project manager (ScrumMaster) acting as the facilitator.

To decide which backlog items we

should work on, we identify risks by using a structured brainstorming method. Team members write down on cards the most critical risks they perceive. A risk might be "The current asynchronous communication mechanism can't handle our scalability requirements" or "Product management isn't clear on the format of the user online help." Each team member presents his card to the team and pins it on the wall. Notice that we don't necessarily list the risk impact and drivers, but we frequently mention them when we present the cards.

When all team members have presented their risks, we group them into categories—for instance, *Communication and User interface technology*—to informally analyze the risks. Then each team member votes on the risk groups by sticking points onto the groups. Team members may accumulate their votes if they feel a certain risk group is particularly critical. Once all team members have voted, the project manager tallies the votes and writes them next to each group. Then, we prioritize by selecting the three to five top risk groups as the critical risks to be addressed. Notice that we don't perform

any formal loss analysis, such as quantifying the loss associated with each risk or risk group. Instead, we tap into the team's collective knowledge to determine which risks we must deal with urgently. If a risk isn't selected, it isn't addressed in the upcoming iteration.

We either defer these risks for later action or simply accept them. Managing risks this way is sensible: If a deferred risk becomes critical, it will show up in the next iteration's risk management session.



Once the risk prioritization was finished, the ScrumMaster counted the votes for each risk group, thus making each risk's rank clear.

After we prioritize the risks, we map them onto the selected product backlog. For each high-priority risk group, we identify the product backlog items affected. If we can't find a corresponding backlog item, we create a new item that addresses the risk. If communication is a critical risk group, for instance, the mapping would identify the communication framework as an area that we must work on to address the risk. At the same time, we would identify scalability as the requirement associated with the risk. Notice that this step usually requires team discussion and participatory decision-making to ensure that the team commits to the risk response measures. For more information on participatory decision-making, see *The Facilitator's Guide to Participatory Decision-Making* by Sam Kaner et al. (New Society Publishers, 1996). Mapping risks onto the product backlog tells us which work-list items we must select to respond to the risks identified. In this way, the risks prioritize the selected product backlog items and help us to identify the ones that must be addressed in the upcoming iteration, and thus to formulate that iteration's goal.

Reflections

The risk management approach introduced here lives up to the two challenges identified: It's tightly integrated with other iteration planning activities—the risk management considerations truly drive the iteration plan. And the approach also involves the entire team and is brief—in our

experience, the risk management activities we've described typically take about 1.5 hours. We involve the entire project team in risk management activities to make the iteration goal and its connection with critical risks transparent.

It should be emphasized that the project manager plays a crucial role in keeping the team focused and the risk management activities effective. Facilitating the risk management and iteration planning activities involves a participatory decision-making approach and sound preparation. Furthermore, to enable an effective and collaborative mapping of risks onto the backlog, the project manager should ensure that all team members listen to the perceived risks and agree on their priority and resolution actions. He should create and communicate a realistic agenda ahead of time, and prepare the room and all materials required to visualize the backlog items and their associated risks.

Notice that this approach merges and integrates project management activities from different knowledge areas, such as time management and risk management. This tight integration and frequent performance enables employment of less formal methods than are typically used in conventional projects or if risk management activities also deal with interdependencies to other projects or stakeholders that aren't part of the project team.

Through a solid understanding of agile principles and values, we can adapt the risk management process to exploit the strengths of the agile approach used. By looking at a project through agile eyes, we can manage the worst 80 percent of its risks with 20 percent of the effort expended in conventional approaches. ■■■