

Who?	Does what?	By when?

Action Plan

8. The team should review the Action Plan before committing to it. Is the Action Plan doable in a few days? Are the assignments fair and balanced among the team members? Once you have agreement, you have team commitment and accountability. All members of the team know what the team expects from them, and what they expect from other members of the group.
9. The scribe should document the Team Learning Record and Action Plan and make it available to everyone on the team.

What to turn in:

Turn in a professional-looking document that includes the following:

1. The project name, project team name, and the names of your project team.
2. The brief summary of the challenge, problem, or opportunity statement that your team developed in this assignment.
3. Document the Team Learning Record and the Action Plan. You may not have had a chance to complete all of the tasks or assignments, so just include the Action Plan you developed during this learning cycle.

QUICK THINKING—LEARNING FROM FAILURE

Michael Hugos says that he often learns more from failure than success. He concedes, *“When I succeed, it just confirms what I already know—I’m a genius. When I fail, I have an opportunity to learn, if I can bring myself to take an objective look at what happened. This is hard, but then making the same mistakes over again is even harder. So failure can be a great opportunity to learn.”*

Hugos also provides some lessons learned from what he calls “one of the greatest learning experiences in his career” when he was a development leader on a systems development project that turned into a multimillion-dollar disaster. Since then, he has delivered many new systems successfully, and much of that success is due to the lessons he learned from the failure of this project. The following is a summary of what happened and some of the lessons learned from his experiences on that project.

- Although the project started out with great fanfare and enthusiasm, there were no clearly defined goals or objectives. The basic idea behind the system was to empower the sales force to grow revenues by \$1 billion. *Lesson: Be wary when projects start out with wild enthusiasm and unclear goals. This can lead to the “bandwagon effect,” where intelligent people do dumb things.*
- The first six months of the project was spent investigating technology and dreaming up ideas.

The development team put together a slide show and a short demonstration of some of the technology. Senior management liked what it saw and approved major funding for the project. *Lesson: Getting lots of ideas and money can commit a team to unrealistic expectations. A better approach maybe to focus on only a few realistic ideas that cost less money.*

- Four teams were working together on the project. One team was responsible for programming and hardware selection, while the other three worked on design specifications. Although all four teams were supposed to work together, the design teams began to duplicate each other’s work. No single person was in charge of the entire project. Team members became confused, tempers flared, and feelings were hurt. *Lesson: Teams should have clear and defined assignments. The project leader should resolve disputes to keep the projection track.*
- After six months and hundreds of pages of specifications, the design was still incomplete, but pressure mounted to start programming. Regardless, the design was handed over to the programming team who were overwhelmed by the volume and complexity of the specifications. *Lesson: Spending more time designing a system will result in greater complexity. It may be better*

to design and build smaller components of the system in short, iterative steps.

- To cope with the pressure, the programmers began to change the specifications and cut out features they didn't understand. In addition, new hardware and software releases kept coming out, so the programmers rewrote many of the programs to take advantage of the new technology releases. It took about a year to program and reprogram the system. *Lesson: System specifications must be clear and complete. Developers should stick to them and not redesign the system while building it. New features can be added in future releases.*
- Beta testing resulted in a slow system that crashed often. *Lesson: After almost two years and such high expectations, the performance of the tests seriously damaged the credibility of the project.*

- Support for the system began to fade as the programmers scrambled to fix the bugs. Senior management began to question the constantly increasing budget and cancelled the project—writing off millions of dollars. *Lesson: Dividing a large project into smaller subsystems or projects is better than trying to deliver one large system in a few years. Smaller systems are easier to debug and can show a return to the organization more quickly.*

1. Should a project team wait until the end of a project to document its lessons learned?
2. How can lessons learned be documented and made available to other project teams?

SOURCE: Michael Hugos, "Lessons Learned from a Major Project Failure," *Computerworld*. August 21, 2006.

QUICK THINKING—DOING AGILE OR BEING AGILE?

Many people believe Agile is the future. After all, Agile promises projects that are delivered on time, within budget, high quality, and satisfy the customer. In fact, success stories of IT projects has inspired Agile to be used on nontechnical projects like the development of NPR radio programs, managing churches, planning weddings, as well as running a household and raising children.

Lajos Moczar has a great deal of experience with Agile, but cautions, "I've concluded that agile has not only failed like other fad methodologies before it but, in fact, is making things worse in IT." He believes that Agile has three major flaws:

1. The continuous delivery of valuable software can lead to developers being more concerned with delivery over quality. Subsequently, this leads to an ever-increasing backlog of defects. This can increase the stress and workload of the developers who may burnout instead of working at a constant and sustainable pace. Moreover, the users or customers will become increasingly dissatisfied with a poor-quality product.
2. Another Agile principle is to respond to change over following a plan, where developers and the customers define and redefine requirements. These changes can be large or small, but many people don't make this distinction because Agile

is predicated on supporting changes. As a result, many large and more costly changes are not made until late in the project. The only way to handle these major requests is to add more iterations, which can increase the project's schedule and budget, as well as the potential for more defects.

3. The third flaw arises from empowering self-organizing teams. This can lead to an "immature utopian myth" as there is still a need for responsible project management. The project team must have the right people with the right political motivations.

In addition, David Tabor suggests that distance and time can create problems for Agile projects. Agile requires close collaboration between users and developers to increase communication and to establish a strong and trusting relationship. Physical distance can create an impediment even if the project team is on different floors of the same building, but this becomes even more pronounced if the team members are in different geographical locations or when national boundaries are crossed.

While Agile projects are founded on speed and responsiveness, time, in terms of delaying the start of the project by just a few weeks, can pose significant issues or problems. If requirements were defined, they may have an expiration date because budgets and planned sprints

may become quickly obsolete. As David Tabor explains, “To use a bad analogy, agile is fresh vegetables that are better for you—but they don’t have the shelf life of the canned stuff.”

1. Come up with a good example of how Agile could be used on a nontechnical project. What advantages would Agile have over a more structured approach like Waterfall?
2. As a project manager, how could you ensure that your developers and customers were “being Agile” rather than just attempting to “do Agile?”

SOURCES:

- Johnson, P. “NPR Adopts Agile-like Method for Program Development.” *Computerworld*. August 14, 2012.
- Moczar, L. “Why Agile Isn’t Working: Bringing Common Sense to Agile Principles.” *CIO*. June 4, 2013.
- Phelps, A. “Agile, Social, Cheap: The New Way NPR is Trying to Make Radio.” *Nieman Journalism Lab*. April 27, 2012.
- Sutherland, A. C., J. Sutherland, and C. Hegarty. “Scrum in Church.” 2009 Agile Conference, Chicago, IL. August, 2009.
- Tabor, D. “Time and Distance Enemies of Agile Project Management.” *CIO*. July 11, 2013.

CASE STUDIES

Do Certifications Matter?

Many people and organizations value the Project Management Professional (PMP)[®] certification because it requires both project management knowledge and experience. According to the Project Management Institute (PMI), who administers and oversees the PMP certification, this recognition can provide increased marketability to employers and a salary up to 10 percent more than non-credentialed colleagues and peers.

The requirements for the PMP are a four-year degree (bachelor’s or the global equivalent) and at least three years of project management experience. This should include 4,500 hours leading and directing projects and 35 hours of project management education. On the other hand, a person could have a secondary diploma (high school or the global equivalent) with at least five years of project management experience with 7,500 hours leading and directing projects and 35 hours of project management education.

In addition to real-world project management experience, PMP certification also requires passing a four-hour, 200-question examination that covers the Project Management Body of Knowledge (PMBOK)[®] knowledge areas and processes. The exam is challenging and comprehensive.

The fact that you cannot take the exam without substantial and demonstrated experience makes the PMP an often sought-after credential. Many advertisements for project managers require or prefer this certification. For example, IBM’s Project Management Center of Excellence and the IBM Global Business Services’ Project Management Competency oversees project manager development programs. More than 14,000 of IBM’s

300,000 employees have attained the PMP certification, and the number is growing because clients want project managers with a PMP certification on their projects. Many clients associate certification with strong project management knowledge and may not consider a noncertified project manager.

Another reason PMP certified project managers are in demand is the perception that someone who has devoted thousands of hours to preparing the exam has the ability to keep a project on track. But the real question is whether having a certified project manager increases the likelihood of project success?

Two separate studies have linked certification with project performance. The first study, conducted by Price-Waterhouse Coopers (PWC) found that “higher-performing projects are significantly more likely to be staffed with certified project managers and 80 percent of projects classified as high-performing use a certified project manager.” The second study was conducted by PMI in 2008 and is called the Pulse of the Profession. This study reports “that having project managers without PMP certification results in a lower percent of projects coming in on time and on budget—especially when less than 10 percent of the project managers in the company are PMPs.”

However, many people are unconvinced by the findings of these two studies. It is difficult to prove that certification has a positive impact on projects because there are too many variables that can influence the outcome of a project, such as funding, resource management, end-user buy-in, and executive support. An effective project manager has the ability to deliver projects that meet stakeholder expectation on time and on budget,