CSC-580 PM CASE: THIRD AVENUE SOFTWARE HEALTH-CARE APP PROJECT

This case is new for the ninth edition of *Information Technology Project Management*. The case provides an opportunity to apply agile and Scrum principles to project management.

Each part of the case contains several task assignments to help you explore the use of agile and Scrum principles.

Part 3: Project Schedule Management

Remember that project schedule management is different in agile projects than in traditional project management. For example, agile projects do not always require activity durations or project schedules. However, after reviewing the scope documents prepared in Part 2 of this case, two members of management have expressed misgivings about the team's ability to complete the work by the deadline and to bring the project in under budget. Also, they are concerned that other company work might require portions of the project programmers' time. Remember that an emphasis on stakeholder interactions and collaboration is a key component of agile projects, and management is certainly a key stakeholder. Therefore, some scheduling work will be necessary.

Because of management's concerns about scheduling, they have requested that you add two members to your team:

- Aziz, a Quality Assurance tester who began his career as a programmer
- Barry, a member of Third Avenue's three-person accounting staff

As the product owner, you have done some research on agile-specific scheduling and think that the scheduling approach used by the FBI to complete its Sentinel computerized file system will work for the Third Avenue project. This scheduling approach was discussed in Module 6. In the Sentinel project, work was organized into user stories, each of which were assigned several "story points" based on how much work was needed to complete each task. Story points are an abstract measure of the amount of effort needed to convert a user story into a functioning piece of software. Story points are calculated, or sized, based on the estimated amount of work needed, task complexity, risk in doing the work, and time required to do the work.

At the start of each two-week sprint, the team decided which user stories to complete for that sprint. Completed parts of the app were then incorporated into the next iteration of the software build for customer review and approval. User stories still pending completion were kept in the product backlog awaiting future sprints. This approach helped the team focus on completing a system that met customer requirements in a timely manner. The agile approach emphasizes finishing subsets of software features for the customer in regular, short intervals as opposed to an attempt to define and schedule the entire project at the beginning.

Management has also asked the team to develop a list of project milestones and make sure these can be completed within the sprint schedule, which is once every two weeks.

- *Open a new Microsoft® Word document and complete the **Tasks** below.*
- *Save the file on your computer with your last name in the file name. (Example: part 1 tasks _Jones.doc)*
- *Click the **Choose File** button to select and upload your saved document.*

Tasks

1. Refer to the user stories and technical stories you created to describe the software requirements for the health-care app in Task 2, Part 2 of this case. For each story, assign several story points based on the programmers' original estimates of task complexity and how much work you think will be needed to complete each task. These numbers should help you refine your work in the next task.
2. Using your ideas for developing a high-level schedule from Task 6, Part 1 of this case, develop a more detailed schedule for the minimum viable product (MVP) you defined in Part 2. You can use scheduling freeware that you download from the Internet or create a schedule with pen and paper. Include milestones within this revised schedule. Recall from Module 6 that a milestone is a significant event that normally has no duration; it is a marker to assist you in identifying necessary activities. It can take multiple activities and extensive work to complete a milestone. They are also helpful for establishing schedule goals and checking your progress. In a software project, milestones can be represented by the completion of specific modules, tabs, and feature sets.
3. Now that you have a more refined schedule, consider any possible changes to the MVP you defined in Part 2. In other words, does it still seem realistic to complete the MVP in the first six weeks (three sprints) of the project? Explain your answer in two or three paragraphs.
4. Also, based on your preceding work, develop your best estimate of a schedule in which the team can plan to release all subsequent software iterations for the remainder of the project. Again, you can use scheduling freeware that you download from the Internet or create a schedule with pen and paper. Keep in mind that the success of agile and the Scrum approaches is predicated on flexibility, so this schedule is subject to change.
5. To help determine scope in a traditional project, the team would develop a work breakdown structure (WBS). In Scrum, teams instead use product and sprint backlogs to develop a high-level description of the work that needs to be done. A traditional Gantt chart can still be useful in agile projects, however. Prepare a Gantt chart for the health-care app project; you can use Figure 3-6 as a guide if necessary. You can download freeware from the Internet that enables you to create the Gantt chart, or you can create it with pen and paper. Your Gantt chart should incorporate all the new information and milestones you developed in the preceding tasks for Part 3.