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 2: Project Scope Management

As one of the two senior programmers at Third Avenue, you have been selected to run the project for developing the health-care app. You will be joined by the following colleagues on the project team:

- Eric, a junior programmer who is considered by his peers to be the author of some of the best code at the company. You have also designated Eric to be the project ScrumMaster.
- Lia, another talented young programmer
- Brianna, a marketing representative who has experience in health care from a previous job.
- Jack, the regulatory manager at Third Avenue
- Kendra, the Quality Assurance manager

Remember that project scope management is different in agile projects than in traditional project management. For example, participants in agile projects typically spend less time defining scope in early stages of a project. However, Third Avenue has high hopes for the health-care app and wants to make sure that all team members work out some basic, crucial requirements before proceeding. Also, agile projects generally require more iterations of working software than in traditional project management, so management must be willing to trust the process once the basic requirements are in place and understood.

To help develop scope, agile and Scrum approaches employ *cards*, *user stories*, and *technical stories*. User stories are often written on index cards and then arrayed on a wall or tabletop to help the agile team plan how to implement the ideas into the product. Technical stories are then developed from the user stories. Technical stories can contain one or more technical tasks that developers use to chart progress on a sprint board as work is conducted throughout a sprint. This approach facilitates group discussion, which often leads to a much better set of product specifications than the rather simple ideas expressed on the cards.

One of management's key goals is to have the team develop ideas for completing a minimum viable product (MVP) as soon as possible. An MVP is a streamlined, stripped-down version of a product that can still be released for real-world use and review. It contains a subset of features that will be included in the final version. An MVP must possess several key properties:

- It has sufficient usable features and value that users will buy it.

- These early users will see the potential benefits of the product and trust that it will only improve in later iterations.
- It provides a feedback loop that will help programmers improve the existing features and add new features with minimal delays.

Remember that the overall budget for the project is $350,000, and Third Avenue management would like to see a finished application available in four months. The MVP version, of course, must be available much more quickly—management wants it to be ready to ship in six weeks. The project team has decided that sprints will be done every two weeks, so the MVP version must be ready to ship for use and review after three sprint cycles. The budget for completing the MVP is $120,000.

- *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. Based on what you have learned in Part 2, complete the project charter you began in Part 1.
2. Part 1 of this case listed the key features needed for the app. The list is quickly summarized here:
    - A fitness tracker for recording health information, such as blood pressure and cholesterol
    - A medication tracker (electronic pillbox) with a calendar and alarm notifications
    - An electronic address book for recording contact data of doctors and other health-care professionals
    - An emergencies list for storing vital phone numbers and addresses to provide quick access to hospitals, urgent care clinics, children, and friends in an emergency. List entries will trigger interactive GPS mapping software to help locate hospitals and other health-care venues.
    - An emergency information list in which customers store important data about themselves in case it is needed in an emergency.
    - A resources feature that lists links to other popular online health sites, such as WebMD
    - A payment feature that tracks health expenses and allows customers to make related payments through their phones.
    - Usability issues

    Using this feature list, develop a set of cards, user stories, and technical stories to describe the software requirements for the health-care app. Remember from your

course readings that user stories describe what users need to do to execute a task or perform a job function, focusing on the "who," "what," and "why" of a requirement in a simple, concise way.

3. The "Collecting Requirements" section of Module 5 discusses several methods for gathering requirements, including questionnaires, surveys, stakeholder interviews, prototyping, and context diagrams. Based on your knowledge of agile and Scrum, which of these methods should the Third Avenue team use to collect requirements for the project? Write a two-paragraph response to defend your answer.
4. Develop an initial scope statement. Make sure to follow the detailed process shown in Module 5. Recall that a good scope statement requires some of the items shown in the following table.

Components of a scope statement
Information from the project charter
Product scope description
Functional and design specifications for developing software
Product user acceptance criteria
Detailed information for project deliverables
Project boundaries, constraints, and assumptions
References to supporting documents, such as product specifications or corporate policies

Based on your work in developing the software requirements and scope statement, develop a list of features that will become the MVP for the first iteration of the health-care app. For example, the programmers' initial ideas for the app include (a) an electronic address book for recording contact data of doctors and other health-care professionals; and (b) an emergencies list for storing vital phone numbers and addresses of hospitals and other emergency venues. Should these two items be combined in the MVP version? Consider such issues as you develop your list.