CIS 571 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. If you have difficulty understanding a task assignment, please ask your instructor for assistance. Your instructor might be willing to look at the task solution and provide a hint to help you proceed.

Part 1: Project Integration Management

Third Avenue Software is a relatively young company that develops mobile applications for phones. The company is still trying to find its corporate identity and permanent footing; it has released several moderately successful products but is still looking for a best-seller. Likewise, the company is still trying to determine which internal systems work best for its employees. Project management is among these systems. The company has used a few agile principles in previous projects with some success; its new project will use agile and Scrum whenever possible.

Many of Third Avenue's products thus far have been designed to serve niche markets, so the company's cofounders instructed their marketing staff and programmers to identify markets that have more universal customer appeal. A couple of programmers quickly turned their focus to the field of health care, which affects everyone directly or indirectly. The programmers drafted an idea for an app that could serve as a "one-stop shop" for customers' health-care information and needs. The app's name is to be determined, but it will contain the following features and information. Because Third Avenue knows from experience with agile projects that software complexity ratings can be useful for later time and cost estimates, management asked the programmers to include initial complexity estimates for each major feature set. These numbers are shown in parentheses and use a scale of 1 to 8:

- A fitness tracker that allows customers to record and track their blood pressure readings, cholesterol levels, exercise regimen, calorie intake, and other related information (3).
- A medication tracker in which customers can enter their medications and schedules for taking those meds. This "electronic pillbox" will contain a calendar that displays the customer's medication schedule and an alarm that sounds whenever it's time to take one of the medications (3).
- A physicians list that is essentially an electronic address book for the customer's health-care company, doctors, nurses, and physician's assistants. The list will include controls that allow customers to quickly incorporate existing entries from other contact lists in their phones (2).
- An emergencies list for storing vital phone numbers and addresses. This list will provide quick access to local in-network hospitals, urgent care clinics, and children or friends who

can be relied upon to provide transportation in an emergency. As soon as the customer enters and saves an address, an interactive GPS map becomes available in a new window, with voice and text directions (6).

- An emergency information list in which customers store important information about themselves, such as medical conditions (e.g., the customer is diabetic), allergies, adverse reactions to drugs, and other personal information that a physician, nurse, or other concerned party might find useful in an emergency (2).
- A resource feature that lists links to other popular online health sites, such as WebMD. The customer will have the option to add links to the list (1).
- A payment feature that tracks the customer's medical expenses and allows customers to make medical payments through their phones (4).

The budget for the project is $350,000, and Third Avenue management would like to see a finished application available in four months.

Scrum will be the preferred approach to managing the project's development because Third Avenue wants a working version of the application quickly but does not yet know the full scope of the project. This working version will be released for review and testing well before the planned official release in four months. Remember that agile projects involve numerous iterations and software versions before the final release. These versions should be responsive to the concerns expressed by all stakeholders.

For example, programmers assigned to the app's development might be needed to provide support for other company projects, and more functionality might be added to the app after various stakeholders have had an opportunity to evaluate the first working version.

Usability

Usability will be extremely important, as customers will tend to be older than those who download and buy the majority of mobile apps. For example, the app will require a *prominent* control for increasing the text display size. Such controls are available in a phone's Settings feature, but many older users tend not to explore such "hidden" settings.

The features mentioned above need to be immediately available and easily accessible when the app is launched.

Another usability issue is crucial: How does the app balance customer privacy against the need to share some of the customer's information in an emergency? For example, the emergency information list might be of no use in a medical emergency if the customer's phone access is blocked by a password that only she knows.

Taken as a whole, programmers give usability issues a complexity rating of 4 on a scale of 1 to 8.

Monetizing the App

Another unknown is the question of how to monetize the app most effectively—for example, the app will use ads, but how? Pop-up ads are an annoyance to many people; will they be tolerated by users or will they be immediately rejected? Will the app offer premium services, and if so, what are they? Will a subscription paywall be viable after an initial period of free use?

- *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. Review the seven processes of project integration management and identify which processes are needed to begin planning the project from an agile perspective. Briefly explain your reasoning for including and excluding processes. The processes are listed below and explained in more detail in the Module 4 Reading.

Seven processes of project integration management
1. Develop the project charter
2. Develop the project management plan
3. Direct and manage project work
4. Manage project knowledge
5. Monitor and control project work
6. Perform integrated change control
7. Close the project or phase

2. Begin developing a project charter for the health-care app project. Assume that the project will take four months to finish and have a budget of $350,000. Use the project charter template provided in this text and the sample project charter in Table 4-1 if you need assistance. Project personnel have not been determined yet, so do not be concerned for now with this area of the charter.
3. Third Avenue first needs to identify a good project manager. Remembering your study of agile concepts in the text, by what title is the project manager known when using a Scrum approach? What skills and qualities must this person possess in order to lead the project effectively? How do these skills and qualities differ in a Scrum approach versus that of a more traditional project management style?
4. Next, the person identified in Task 3 must form a team and establish a project framework within which the team will create a successful app. Describe at a high level how the team and framework will function, using as many relevant terms and concepts from Scrum as possible.

5. After identifying a manager, team members, and project framework, Third Avenue needs to research the market to determine what competing apps might exist and how they operate. Your task here is to locate a similar mobile app or online program and then get a feel for its content and users. Use a targeted Web search to find the app or program and then spend a half-hour or so reading about it to get an idea of what the Third Avenue application should be able to do. Describe your findings in a bulleted list. Is something important missing from the preceding list of features for the health-care app?

6. Once the team has studied the app or program in Task 5, an initial meeting is necessary to discuss the features and content needed for the software's first software iteration and to assign tasks to team members. The team also needs to establish schedules for project milestones and subsequent meetings. List your ideas for conducting the initial meeting and for creating an initial high-level schedule, using as many relevant terms and concepts from Scrum as possible.

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 table top 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.

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 a number of "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 back 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 a number of 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 Scrum approaches are 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.

Part 4: Project Cost and Quality Management

As with other methods of software development, the application's size is a major indicator of how much it might cost to develop. In agile development, cost estimates are often made based on size measurements such as story points.

You will recall that each user story for the health-care app is assigned a number of story points based on estimates of how much work is needed to complete each major task. Based on prior experience with agile projects, the three-person accounting staff at Third Avenue Software has determined an average dollar production cost for a story point: $1200.

However, the staff accountants are not completely confident in this average dollar value because Third Avenue's experience with agile projects is not extensive. The accountants would like the team to confirm their calculations, if possible.

As you learned in Module 7, earned value management (EVM) is a more traditional project management method for determining whether a project is meeting time and cost goals. EVM requires calculation of three values for each major activity in the project:

- The planned value (PV), which is the authorized budget assigned to scheduled work
- The actual cost (AC), which is the realized cost for the work performed on an activity during a specified period of time
- The earned value (EV), a measurement of project work done, framed in terms of the approved budget for the work

Several more values and formulas are involved in EVM. For more information, refer to the "Earned Value Management" section of the Module 7 Reading.

Managers at Third Avenue are also eager to see evidence that the Quality Assurance staff are making progress in their ability to test the health-care app. They have asked the team to provide at least a basic framework of test specifications.

- *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. Review the user stories, technical stories, and story points you developed in earlier parts of this case and then use the information to estimate the cost of the health-care app project. As you learned in the introduction to Part 4, Third Avenue accountants estimate that the average production cost of a story point is $1200. Based on the values you calculate, do the project costs appear to be within the overall budget of $350,000? Do the costs seem to be within the MVP budget of $120,000? If the costs are not within budget, adjust the story points you developed in Part 3 to make sure your revised numbers are within budget. You might also have to reconsider the feature set within the MVP if it exceeds its budget.
2. Assume that you have completed one month of the health-care app project and have some reliable EVM data for cost accounting. Remember that the budget at completion (BAC) is $350,000 for the four-month project. You have received the following figures from the Third Avenue accountants:

   PV $105,000

   EV $122,000

   AC $105,000

   Using this information, answer the following questions.

   - What is the cost variance, schedule variance, cost performance index (CPI), and schedule performance index (SPI) for the project?
   - Use the CPI to determine the estimate at completion (EAC) for the project.

- Based on your answers, does the project appear to be on schedule and within budget?

3. The health-care app will undergo rigorous software testing by the Third Avenue Quality Assurance staff, using test plans that might fill entire binders. At this early point in the process, however, such plans are still in development. Develop a short list of quality requirements for testing at least five of the important app features and/or usability issues described thus far in this running case. In your list, briefly describe each requirement.

   For example, here is a short (and far from complete) list of quality requirements for the emergencies list feature, which stores vital phone numbers and addresses as well as provides quick access to local in-network hospitals, urgent care clinics, and children or friends who can be relied upon to provide transportation in an emergency:

   - Do all of the data-entry fields accept text without problems? For example, long entries in a field can sometimes be clipped or truncated prematurely, which is frustrating to users.
   - Is it possible to enter nonsensical data in a field? For example, users should not be allowed to type text characters into a field that records phone numbers.
   - Does the software automatically move the user from one entry field to the next? Does the cursor always appear in the next entry field as a means of visual feedback for the user?
   - As soon as the user enters and saves an address, an interactive GPS map should become available in a new window, with voice and text directions. Does the window appear? Does it appear immediately?
   - When you slide your finger to create the effect of movement within the map, does the software redraw in a timely manner?

4. Modules 2 and 8 of the text discussed Kanban, a method sometimes used with Scrum. Kanban uses five core properties:

   - *Visualize the workflow*. Cards are a common visualization method.
   - *Limit work in progress*. New project work is done when it can be incorporated into the next software iteration and when there is available capacity. In this way, problem areas are quickly revealed for resolution.
   - *Measure and manage flow*. A key here is to analyze problem areas and then implement changes to correct the problems as quickly as possible.
   - *Make process policies explicit*. Everyone on the project team must understand the processes and any problems with them.
   - *Use models to recognize improvement opportunities*. The models themselves are less important than the important agile concept of continuous improvement.

Based on your work on the case so far, which property of Kanban has proven most useful to you? Write a two- to three-paragraph answer.

Part 5: Project Resource and Communications Management

The effort to constantly reevaluate and improve a project's artifacts, framework, and resources should be an organic feature of an effectively run agile project. This effort fits squarely within the agile principles of continuous improvement and adaptive planning. Likewise, effective communication between team participants is a concept that should be baked into the agile process from the outset. A smooth process of resource allocation and effective communication is considered a hallmark of the agile approach.

In other words, project resource management and project communications management can function as distinct knowledge areas within traditional ideas of project management. However, these distinctions should blur in a well-run agile project.

- *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. In Part 1 of this running case, you were instructed to form a team and establish a project framework within which the team would create a successful health-care app. Reevaluate your answers in Part 1 based on what you have learned since then. Identify at least one way you might adjust project resources via the product backlog, sprints, the daily Scrum, or some other aspect of the project approach.
2. Good communication is considered one of the strengths of the agile approach when compared with more traditional methods of project management. Why is this so? Explain your answer in three to four paragraphs.
3. Develop a simple progress report or status report for the project as it stands now. Progress reports and status reports address where the project stands in terms of the triple constraint—meeting scope, time, and cost goals.

If you did not use a burndown chart to create the progress report or status report in the previous task, do so now. Burndown charts are described in detail in Module 3. If project work is going well, the remaining story points should track in accordance with the ideal velocity.

Part 6: Project Risk and Procurement Management

Throughout this case, you have been introduced to various issues that create risks within the health-care app project. For example:

- Early on, management expressed misgivings about the team's ability to complete the work by the deadline and to bring the project in under budget. Basic concerns about scope, time, and cost are obvious candidates to become project risks.
- If management tries to mitigate these risks by micromanaging the project and interfering with the team's work, such action creates a risk of its own for the product owner and team.
- The health-care app should appeal to an older audience than customers who download and buy the majority of mobile apps. Therefore, the issue of usability is crucial, as older users are often not as tech-savvy as their younger counterparts. This issue could create some profound market risks if the app's usability features are not well conceived and executed.
- Do not forget that risks can be positive as well as negative. Third Avenue Software believes that it could create a best-selling app with great market appeal.

As for potential procurement issues, recall that procurement is the acquisition of goods and services from an outside source. Procurement is also known as *purchasing* and *outsourcing*. Third Avenue believes it has the resources to complete the project in-house, but management knows it must remain open to the need to bring in an outside source in case project deadlines or costs demand it.

- *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. From the course readings, why might agile environments lead to greater risks in a project than one that uses more traditional project management techniques?
2. Develop either a risk register or a probability/impact matrix to document risks to the health-care app project. Be creative; examine the previous case parts for possible clues about risks the project faces, and then develop your response. You can also develop risks of your own, based on your understanding of the health-care app project and your reading of the text. Be prepared to defend your answer based on information in Module 11.
    - As you learned in Module 11, a risk register is a document that displays results of various risk management processes in a table or a spreadsheet format. A risk register helps document potential risk events and related information.
    - A probability/impact matrix lists the relative probability of a risk occurring and the relative impact of the risk's occurrence. This simple technique helps project teams identify risks that need attention. Note that if you choose to create a

probability/impact matrix, you must include an explanation of the numbers in the matrix.

3. Review your ideas for the first and second iterations of the health-care app. You do not have to develop a risk register or probability/impact matrix for each iteration, but how did the risks change from the first iteration to the next? Identify at least two possible changes and briefly describe how and why the risks changed.
4. Prepare a justification for outsourcing part of the health-care app project or for performing all the project work in-house. You can argue either side of the issue in three to four well-reasoned paragraphs. For example, an argument *against* outsourcing is the necessity of speed, especially in agile projects. You can develop this argument based on your reading of Module 12 or develop a new justification of your own.

Part 7: Project Stakeholder Management

As you learned in the text, it is critical to involve key stakeholders on important projects and focus on their needs. In agile projects, however, increased stakeholder involvement and faster decision making are often required. To enable timely, productive discussion and decision making, agile teams communicate directly with stakeholders instead of wading through tiers of management to establish such communications.

The Third Avenue health-care app project involved the following stakeholders:

- You, the product owner. You are one of two senior programmers at Third Avenue.
- 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
- Aziz, a Quality Assurance tester who began his career as a programmer
- Barry, a member of Third Avenue's three-person accounting staff
- Members of Third Avenue management, who occasionally expressed doubts about the team's ability to complete the work by the deadline and within budget

    - *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. The preceding list contains a rather glaring omission. What group of stakeholders is not shown in the list?
2. Review the preceding list of project stakeholders and project members, and then identify two examples of how these stakeholder roles might have been different in a traditional project management approach.
3. Create a stakeholder management plan to handle the stakeholders' concerns and input for the health-care app after it has been released to the field. For example, continuing strategies are needed to monetize the software most effectively and to achieve maximum market penetration.

Create an issue log for the project's main activities. Recall that an issue log is a tool used to document, monitor, and track issues that need resolution.