IT 304 Systems Requirements Project Two  
Implementation Plan

**Southern New Hampshire University**

**Millennia HealthCenter**

**Cara Hanley**

**April 15, 2022**

Table of Contents

[1. Introduction 3](#_Toc21700751)

[1.1 Purpose 3](#_Toc21700752)

[1.2 System overview 3](#_Toc21700753)

[2. Implementation approach and schedule 3](#_Toc21700754)

[3. Phases 3](#_Toc21700755)

[4. Tasks 3](#_Toc21700756)

[5. Resources 3](#_Toc21700757)

[5.1 Hardware and software 3](#_Toc21700758)

[5.2 Personnel 3](#_Toc21700759)

[5.3 Training 3](#_Toc21700760)

[5.4 Budget 3](#_Toc21700761)

[6. Durations and estimates 3](#_Toc21700762)

[7. Task dependencies 4](#_Toc21700763)

[8. Performance monitoring 4](#_Toc21700764)

# 1. Introduction

## 1.1 Purpose

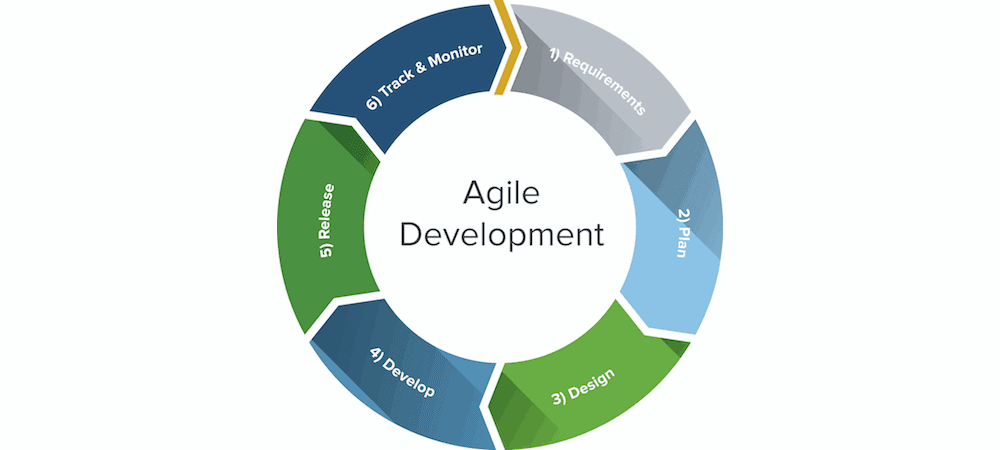
The purpose of this implementation plan is to provide a detailed framework for the installation of the new Millennia HealthCenter software system. It will outline a strategy that is broken down into specific tasks that are comprehensive and manageable, and which will ultimately help MHC realize the full benefits of the software system. This document should be used as a working document for the project team, to include the development team, engineers, QA testers, system analysts, system administrators, and IT department.

## 1.2 System overview

The purpose of this system is to provide patients, practitioners, and administrative staff within the Millenia HealthCenter (MHC) organization access to the tools and resources needed to conduct activities related to healthcare. The objectives of restructuring the current system are to: increase the accessibility to patient healthcare information, improve the quality of online patient care, facilitate the processing and scheduling of patient visits and to establish a reliable and easy-to-use platform that assists practitioners in making valuable medical decisions. The system will increase the flow of information between Millenia HealthCenter employees in different departments as well as across the globe, while maintaining data confidentially and integrity, and improving accurate data. This document is intended for the stakeholders, systems analysts, management, and the development team within MHC.

# 2. Implementation approach and schedule

This implementation plan is based on an agile software development model due to its flexibility in allowing testing and debugging in smaller increments, its use of parallel tasking, the ability to resolve any risks during iterations and to accommodate any future changes if needed. Agile development supports frequent adaptation to changes, encourages teamwork over management, incorporates engineering best practices that allow for the rapid delivery of high-quality software, and prioritizes the alignment of company goals with development (Raza, 2020).



(ForTech, 2020)

# 3. Phases

The phases outlined in the implementation schedule are as follows:

Scope & Planning

In the initial phase, the project boundaries are clearly defined, and user specifications are collected to formulate requirements specification documents for the development team to understand what they are building as well as outline the organization’s business objectives. At this stage, the project schedule is determined and resources allotted according to the defined tasks. This phase is important because it helps the team stay within boundaries to avoid scope creep and to give the project an overall direction.

Analysis & Requirements

At this stage, functional and non-functional requirements are analyzed and defined to determine developmental approaches and how they meet stakeholder needs. Software specifications are defined and process diagrams created. “In Agile development practice, the requirements are allowed to evolve overtime throughout the project life cycle” (Arzadon, 2016). This phase ensures everyone is on the same page with requirements and how those requirements align with the business objectives.

Design

"At this stage, user experience design is formulated and software architecture is coming together. This phase establishes how users will interact with the software and begins to answer questions raised during the analysis phase. The result of the design phase is a prototype of the software so the elements and flow can be viewed together.

Development

The development phase is where developers begin coding after reviewing the software requirements and system design specifications to eventually turn them into reality. In an agile development model, the goal is to have a working, usable program by the end of the iteration. The developers will test their code (aka iteration testing) as they move through this phase before passing it off for more unit and integration testing.

Testing

After a testing plan is established by the quality assurance team, the in-depth testing process begins. This is where important coding errors or bugs are identified so that they have a chance to be fixed before release to the client. The testing team, consisting of quality assurance, developers, engineers, and end users all have a hand in performing testing of the code submitted by the developers. The testing phase consists of functional testing such as unit, integration, system and acceptance testing, as well as non-functional testing such as performance, usability, compatibility and security testing.

Deployment

After testing validation and gaining the approval of the product owner, the production environment phase begins. The new program features and functions are installed on the servers and it is now available for users to test the pilot or beta system. Any iterations after implementation are to update the existing software, to add new features or to fix issues (Atatus, 2021).

Post Implementation Review

Finally, the Project Manager gathers the development team to discuss lessons learned and the overall progress towards fulfilling all requirements. The team discusses suggestions for improvement as they prepare to enter the next development cycle. Going over some lessons learned helps the team identify what needs to be corrected going forward so that that each iteration can be even better than the last.

# 4. Tasks

The tasks within each development phase provide a more granular explanation as to how the team is working toward project goals. Those tasks are explained as follows:

Scope & Planning

This phase initializes the project is where scope is identified and a project sponsorship secured. It is also when resources are starting to be identified and discussion of the project is brought to the team.

Analysis & Software Requirements

After a needs analysis is conducted, preliminary software specifications are drafted and a budget comes together. The schedule is then established, delivery deadlines are set and required resources are secured.

Design Tasks

During the design phase, the preliminary software specifications that the systems analyst drafted are reviewed by the design team and they proceed with the web design. The team will incorporate feedback from end users into the functional design specifications.

Development Tasks

There are seven main developmental tasks that will work to fulfill the requests from interviewed staff as well as the problems outlined in the problem statement of the business case. Those tasks are as follows:

* Cloud-based Solution

“Cloud-based solution refers to applications, storage, on-demand services, computer networks, or other resources that are accessed with an internet connection through another provider's shared cloud computing framework” (Del Vecchio, 2022). This stage of the implementation process is where the software engineers will transfer the MHC system to the cloud to make the services available across the globe, thereby creating a more unified system. A cloud solution will provide a much more scalable solution for MHC as it continues to grow. This task is one of the most important and labor-intensive tasks of implementation.

* Patient File Integration

It has been requested from MHC staff that patient files contain full medical history, lab results, x-rays and drug history that is available to view for authorized individuals. This task is where the integration will take place as developers merge data from multiple sources into one.

* Patient Identification

Patient-matching algorithms and referential matching software will be coded and implemented with a goal of improving patient identity management. This is an important step that will minimize the misidentification of patients and their procedures. The goal is to get the matching as close to 100% accuracy rate as possible.

* Permissions and Privilege

The systems administrators and database administrators will have permissions set according to assigned privilege. The patient’s assigned doctor and nurse will have full access to the file while pharmacists will only have access to the patient’s full drug history. Completion of this task will ensure MHC is HIPAA compliant and that only individuals with the need to know have access to full patient files. Users will no longer have to go through different channels to access portions of a patient file since it will all be in one location.

* Install Video Chat

During this task, developers will integrate video functionality on top of the existing MHC chat feature. This will contribute to MHC’s mission to “improve the patient’s health and well-being through innovative approaches in health information technology”.

* Update Check-In Process

The check-in process task will mainly consist of making updates to the current web application to allow for online check-in for users and make the calendar application shareable across MHC. Developers will also add an insurance card imaging function that captures insurance provider information so that this information will be readily available for receptionists during patient check-in.

* Notifications

This task will focus on adding an alert feature to the existing application that will notify pharmacists whenever the system searches for drug interactions that are not compatible with each other.

Testing

The testing phase consists of unit testing and integration testing, which in turn consists of developing test plans, identifying anomalies in the code to specifications, and then modifying and retesting it.

Deployment

During this phase, a deployment strategy and methodology are identified and deployment resources are secured. It is time to deploy the software and release to the client.

Post Implementation Review

The final phase of the project, the post implementation review, is when any lessons learned throughout the project are documented and discussed with the team. The goal is to learn from mistakes as the team moves forward and takes on new projects, improving with each iteration and each project.

# 5. Resources

### 5.1 Hardware and software

The recommended software for this project includes a web-based prototyping tool for the UI/UX designers where they can upload static design files and quickly turn them into high-fidelity, interactive prototypes (Stevens, 2021). InVision and Adobe XD are two such software design tool for prototyping user experiences with the web.

Enterprise cloud adoption will involve finding a cloud vendor such as AWS, who will have the tools available to assist in the migration process. A cloud-based solution will drastically minimize the amount of hardware and software needed to update, secure and maintain the system. If a virtual private cloud set-up is decided, it is essentially a private cloud built on public cloud infrastructure. It will allow the organization to maintain a level of control without hosting and maintaining the entire infrastructure internally (De Vos, 2019). This solution requires less hardware than an on-premises system.

To remain HIPAA compliant, it’s important to have intrusion detection systems (IDS), a web application firewall to monitor web requests, and a log management software to search logs for indications of compromise.

### 5.2 Personnel

This is an enterprise-level project that will require a large team. Initially, the project manager will meet with the management team to discuss scope and planning for the new system. The project manager will then work with the computer systems analyst to go over system analysis and software requirements before obtaining approvals to move forward from the management team.

In the design phase, the computer systems analyst will review the software specifications with the UI/UX designer who will then work on web design for the applications. Together they will develop a prototype based on functional specifications and work with end users to incorporate user feedback. After obtaining approval from the project manager and the management team, they can proceed to the next phase.

The development team reviews and implements the functional and non-functional requirements and features. A cloud engineer and database administrator will work on the cloud-based solution to transfer the system entirely to the web. Software developers will then work on patient file integration while software engineers code the patient-matching algorithms. The systems and database administrators set file access for pertinent individuals based on assigned permissions and privileges. Software developers will apply the video function to the existing chat feature, update the check-in process and implement drug notifications for pharmacists. After all the development tasks are complete, the development team will perform preliminary debugging before passing it off to the QA team for thorough testing.

The QA team will develop unit test plans and integration test plans and perform other responsibilities for testing the system during the development phase. Software developers will conduct the unit testing based on these specifications and modify and re-test code as they go until unit testing is complete. Software developers will also conduct the integration testing to identify anomalies and inconsistencies in the code, modify it and then re-test, to complete the integration testing task.

The deployment team will then define a deployment strategy, secure deployment resources, train support staff on the new features and functions of the system and deploy the software. Finally, the project manager will meet with the development team to conduct a post-integration review once the software has been live for a certain period of time to analyze lessons learned to potentially apply to the next project.

### 5.3 Training

Support staff will need to be trained during the deployment phase to familiarize them with the new features and functionalities of the new system. IT support staff will need to be intricately familiar with the system in order to properly address client questions and concerns.

Since the implementation plan follows an agile methodology, it will work closely with certain end users to test the functionality and usability of the new system from a user perspective. Therefore, user training will take place in stages prior to full system completion. Training will be available to all users post implementation based on a user training package developed by the IT department. The package will include online support and a link will be sent via email of a demo explaining the new system features and functionalities.

### 5.4 Budget

The estimated cost to complete this project is about $1.5 million. This is a generous estimate for an enterprise-level project. Costs to be considered are both initial and long-term costs for AWS cloud infrastructure, the salaries of the project team and any additional equipment or software that is needed to bring the project to completion.

# 

# 6. Durations and estimates

The scope and planning phase of the project is expected to take about 12 days and as it’s starting on 4/15 will be finished in early May. Management should already have an idea of the scope of the project and sponsorship secured. The project manager will likely have experience in determining the resources needed to complete the project. Due to the scope and importance of this project, it is likely to take priority over existing work and therefore the project manager should not have difficulty procuring the resources and personnel needed to complete; however, this task will take the longest to complete in this phase.

It will take almost 22 days to conduct a software requirements and analysis as there are several tasks and multiple personnel involved. Starting this phase right after scope and planning will have a finish date of early June. A draft of preliminary software specifications will need to be written, a needs analysis conducted, as well as a budget and schedule established. Procuring required resources will take the longest to complete.

The design phase will take about 25 days due to the complexity of the work involved and be complete sometime in early July. The UI/UX Design will take the longest at approximately two weeks, but because most of the functionality and features are already in place, it will not take as long as a program built from scratch would take. The rest of the phase is spent incorporating user feedback into design specifications, which could take anywhere from one to five days depending on the volume of requests.

The development phase will take up most of the project schedule at an estimated 71 days from early July to mid-October. Migration to a cloud-based solution will take the longest to complete (about one month) because it involves a certain set of risks, requires lots of programming and consists of developing a comprehensive migration plan. Patient file integration will take about three weeks because “to build a video chat app you can just make use of the APIs available in the market, but the one that you build from scratch can only give you more flexibility, but will take a long time to come into reality” (Manoharan, 2022). Therefore, additional time has been allocated to allow for this build.

It will take about a month in total to develop test plans and perform unit and integration testing. Testing module integration takes about a week, and integration testing will not begin until unit testing is complete.

Determining a deployment strategy, securing deployment resources and deploying the software should take a little over a week and will round out the project for a December 1st completion date. Training IT support staff should only take about two days as they will already be familiar with the overall system and will know which new functions and features were added.

Finally, the development team will meet with the project manager to go over a post-implementation review which should only take about a day to complete.

# 7. Task dependencies

In agile methodology, there are fewer tasks and phases that will be dependent on each other than in other methodologies. The scope and planning phase will need to be completed prior to the analysis and requirements phase to secure project sponsorship and identify the resources need to proceed. Once the scope and planning phase is complete, the project and software requirements analysis can proceed where formal approvals are obtained and resourced are secured.

The design phase must be completed before the development phase because during this phase is when the design specifications are created that will help in the development of the system. The design document specification also needs to be approved by the stakeholders prior to proceeding with development.

Because the project follows an agile methodology, the tasks in the development phase are independent of each other and can be completed simultaneously, personnel and resources permitting. For this project, the Gantt chart is assuming there are at least two engineers or developers working on a task at any given time. The start dates for tasks are staggered and developers will test their code throughout the development phase. The development phase will end whenever the last task is completed and tested by the development team.

The testing phase will occur after the development phase so that they can determine how well the software interacts with other software applications and to ensure it operates according to client specifications. Once the testing is thoroughly complete and any bugs are fixed, the deployment phase can begin. The tasks identified in this phase will depend on each other and must be completed in sequence.

# 8. Performance monitoring

One method to monitor system performance is in the form of user feedback and help desk tickets to understand user experience with the system during everyday usage. Even the best IT professionals cannot always meet user needs on the first try and therefore user feedback is important. Agile methodology is especially useful in incorporating user feedback because even though a feature or functionality is deployed does not mean it cannot be modified later.

In addition to user feedback, the database administrator will monitor the general state of the database and look for performance issues and workload issues. “Effective database monitoring also gives you the opportunity to enhance or optimize your database, to augment overall performance” (DNStuff, 2020). Database monitoring tracks performance of hardware and software using performance indicators, which helps to pinpoint when and where issues occur.

The system administrator will monitor the networks and perform routine maintenance to ensure no issues arise. They typically use tools that help them monitor network traffic and performance over time to look for any irregularities. They might also use tools that will apply patches to systems.

References

Arzadon, M. (2016, October 13). *Requirements Analysis in the Agile World*. LinkedIn.

<https://www.linkedin.com/pulse/requirements-analysis-agile-world-mary-antoinette-arzadon>

Atatus. (2021, June 1). *Agile Methodology*. <https://www.atatus.com/glossary/agile-methodology/>

Del Vecchio, L. (2022). *What is a Cloud Based Solution? Definition and Meaning*. Planergy.

<https://planergy.com/blog/cloud-based-solutions/>

De Vos, C. (2019, May 6). *A Full Cloud Migration Checklist*. Resolute Technology Solutions.

<https://www.resolutets.com/cloud-migration-checklist/>

DNStuff. (2020, January 7). *Ultimate Guide to Database Monitoring + Best Tools List*.

<https://www.dnsstuff.com/database-monitoring-tools>

ForTech. (2020, May 26). *What is Agile Software Development?*

<https://www.fortech.org/what-is-agile-software-development>

Indeed. (2022, April 8). *Quality Assurance Tester salary in United States*.

<https://www.indeed.com/career/quality-assurance-tester/salaries>

Manoharan, D. (2022, March 21). *Steps to Create a Live Video Chat App – Features, Tech Stack and Cost*.

Contus. <https://blog.contus.com/build-a-live-video-chat-app/>

Raza, A. (2020, July 1*). Why is Agile Methodology better?* TechEngage.

<https://techengage.com/what-is-agile-methodology/>

Stevens, E. (2021, November 23). *Essential Tools for UI Designers: A Beginner’s Guide*. CareerFoundry.

<https://careerfoundry.com/en/blog/ui-design/essential-tools-for-ui-designers-a-beginners-guide/>