EFFECTIVE LIFECYCLE MANAGEMENT OF HEALTHCARE APPLICATIONS

Utilizing a Portfolio Framework

Susan M. Houston, MBA, RN-BC, PMP, CPHIMS, FHIMSS

Ryan D. Kennedy, BS

EFFECTIVE LIFECYCLE MANAGEMENT OF HEALTHCARE APPLICATIONS

Utilizing a Portfolio Framework

EFFECTIVE LIFECYCLE MANAGEMENT OF HEALTHCARE APPLICATIONS

Utilizing a Portfolio Framework

Susan M. Houston, MBA, RN-BC, PMP, CPHIMS, FHIMSS

Ryan D. Kennedy, BS

Cover Art by Nix Houston

Contents

Acknowledgments

I would like to thank my family for their ongoing support and understanding. For my husband, you are my rock and you are always there with words of wisdom or sarcasm, whichever is appropriate. For my children, Nicole, Nick, Matt, and Dana, thank you for always being there whenever I needed encouragement. To Ryan, you are a good friend and always a pleasure to work with.

—Susan M. Houston

First and foremost, I need to thank my amazing wife, Megan. Your support has been unwavering, from reviewing draft chapters to removing all distractions for hours (and even days) at a time. I never could have done this without you. To my children, Alex and Katie, you are everything to me – and now we have an exciting new bedtime story to read! To my mentor and friend, Sue, you have guided my career. Thank you for giving me every possible opportunity to succeed.

—Ryan D. Kennedy

We would both like to thank Nix Houston for the wonderful cover art.

Authors

Susan M. Houston is a senior consultant in healthcare IT after retiring as the Chief of the Portfolio Office within the Department of Clinical Research Informatics at the National Institutes of Health Clinical Center. Her background includes clinical nursing, informatics, and project and portfolio management. Houston has presented workshops and lectures at several local, regional, national, and international conferences and meetings. She has authored a variety of articles and books on project management and informatics. She is a member of the Project Management Institute (PMI), American Nursing Informatics Association (ANIA), and the Healthcare Information and Management Systems Society (HIMSS), while serving on various committees.

Ryan D. Kennedy is the Acting Chief of the Project and Portfolio Office at the National Institutes of Health Clinical Center and an instructor for an IT Project Management course at the University of Maryland, School of Nursing. Kennedy's project work has contributed to the implementation of new clinical systems, infrastructure upgrades, and improved workflow processes. In addition, he has presented workshops and lectures at several national healthcare and project management conferences and has been a contributing author on multiple publications related to project and configuration management. His background includes 20 years of experience in configuration management and project management, spanning across multiple government agencies and hospital systems.

Chapter 1

Introduction

> The goal of running high performance IT is to make improvement
> in business fundamentals and achieve high-performance business
> results in the long term.
>
> **– Pearl Zhu**
> *100 IT Charms: Running Versatile IT to get Digital Ready*

With healthcare organization's portfolio of software applications rapidly growing, having a defined framework for how to manage them is essential. This book will describe how utilizing a portfolio management framework throughout the application lifecycle will provide the structure to ensure that all new applications are properly evaluated and once implemented, remain relevant while continuing to meet organizational requirements.

In 2003, there were only about 31% of hospitals using electronic health record systems (EHRs), and in 2017, the number has increased to nearly 99% (Landi, 2017). Office-based physician adoption of EHRs was about 21% in 2004 and expanded to about 86% in 2017 [Health Information Technology (IT) Dashboard]. These statistics only show the adoption of EHRs, while the availability and use of specialty applications have also grown at a fast pace in the same timeframe. While an organization may have a few large "organization-wide" systems such as the EHR, lab, or radiology systems, they also have a large quantity of other clinical, administrative, and research systems. Some larger organizations now have hundreds of software applications to support and manage. While the IT staff is busy implementing new, they also have to maintain the old. Utilizing a standard, repeatable framework will help to manage the large portfolio of software applications.

Portfolio management is an organizational approach on how to manage a collection of projects and investments, such as applications. This approach provides a framework for strategic decisions related to maintaining the portfolio contents within the constraints of available resources, such as human, financial, and infrastructure. The Application Portfolio Office (APO), or the Project and Portfolio Office (PPO), is a new concept within healthcare with the focus on managing the lifecycle of all applications within the organization. This office allows for an organizational view of IT activities and investments to ensure strategic alignment with mission and goals. The purpose is to identify and eliminate redundant applications; manage changes; and evaluate the stability, quality, and sustainability of the applications within the portfolio. It keeps the focus on the organizational goals and objectives, and allocates resources based on the business value.

The lifecycle of an application begins with an idea followed by the initial request to purchase and implement the new software. There should be some form of governance where the request is reviewed and evaluated, leading to an approval decision. This is an important step that is not always in place, but since organizations have limited resources (human, financial, and other), the resources should be used on the requests with the highest priority for the organization, not the priority of the requestor or their department. Once approved, and resources are available, the initial implementation project begins. During this project, it is just as important to prepare for post-live operations and support, as it is to implement the software. Transitioning to operations and support occurs at the end of the initial implementation project and ensures the transfer of knowledge from the project team to the support team. The application will have activities related to adding new functionality and updating or upgrading throughout the operations and support phase. It remains in this phase until a disposition decision is made. Once it's useful life is over, consistent disposition practices should be followed to ensure all aspects of the application are disposed according to the approved disposition plan.

This book will follow an application through its lifecycle as well as provide basic project management principles. The chapters include a project management overview; the management of the new application request; governance; the implementation project, the transition to operations and support, operations and maintenance related to change management and other ongoing activities; and finally disposition. Below is the list of chapters and what they will cover (Figure 1.1).

Figure 1.1   Stages of application lifecycle discussed in this book.

Project Management Overview – outlines project management principles and project management office concepts that will be used throughout the book.

New Application Request Management – describes the management and analysis of new application requests, including input from security, privacy, architecture, procurement, and others.

Governance – defines what governance is and its role, along with the type of information that is required for the committee to make educated decisions, including details on the portfolio of current projects and resource allocation and availability.

Implementation Project – describes the activities required in the project to install and implement the application while ensuring the entire system (application and infrastructure) meets regulatory requirements and properly preparations for operations and support.

Transition to Support – provides a framework for properly transitioning an application to operations and support, which includes documentation, education, and communication.

Operations and Maintenance – identifies how changes are requested, managed, and implemented after the application is live, utilizing configuration and release management or another project along with day-to-day tasks and activities that are required to ensure the application remains up to date, compliant with requirements, and the help desk is able to provide ongoing support to all users and stakeholders.

Disposition – defines the processes of a disposition plan for the application that is no longer used, compliant or relevant, and the activities that should be completed and documented during this final stage of its lifecycle.

While this book will focus on the application lifecycle, it is important to mention the software development lifecycle (SDLC) and how the two are different. SDLC is a defined process related to the development of new software or application. While there are many different definitions of the stages involved in this process, they are basically analysis, gathering of requirements and design, development, testing, and implementation. The process is cyclical and may move through the stages multiple times, such as between development and testing. An application may go through the SDLC process multiple times within its life. There is the initial development, followed by updates where new functionality is developed and implemented or the current functionality is redesigned and improved. If you are implementing a commercial-off-the-shelf (COTS) application, the vendor has already completed the development of the system prior to moving it to general availability (GA). The vast majority of these types of applications are developed to allow the organization to customize them for their unique setting or use. This process of configuring a COTS application would follow the SDLC process during the implementation project. The configuration may include anything from the values in a dropdown for clinical documentation, details for medication order, or specific layout of a report.

Before you can begin managing your portfolio of applications, you need to identify the contents. This seems fairly straightforward since it is just a list of the applications used within the organization, but it is not as easy as it may sound. It is best to start with the definition of an application. One example is a surgical solution that has separate modules for pre-op, intra-op, anesthesia, and post-op. They are all provided by one vendor and can be purchased and implemented independently, but all work together and are hosted on the same hardware. Are they one application or four? Is the application just the application or does it include the entire system, such as database(s), servers, workstations, redundancy, and backups, as well as any other environments (development, testing, or training)? Will your list of applications include those that run in the background, such as antivirus or monitoring agents? The IT leadership should be able to provide the definition so the list of applications has the right level of specificity.

Oftentimes there are multiple teams that have a list of applications based on their role in the organization. Rarely will these lists match. Gathering the separate lists and eliminating duplication will provide a good starting point. Meeting with key stakeholders from each department will provide input to what applications they currently use, no longer use, and how each meets their needs. Any application that is identified as no longer in use should move on to the disposition process, which is discussed in Chapter 8. It should not be a surprise to find applications in use that the IT department was not aware of. These are known as Shadow IT. These applications maybe something small such as running on a single workstation by a few users or larger that are hosted outside the data center and used by many. There are many reasons for Shadow IT, often it comes down to not wanting to go through the request and governance process or wanting to have complete control of "their" system.

Once there is a good list of applications, the information to track for each application depends on the purpose of the list. If it is for reporting purposes, the information to be reported should be included. If it is for support and maintenance, more information about the user groups, infrastructure, and the support documentation should be included. At a minimum, the application name, version, vendor name, departments who use it, where it is hosted (data center, department, vendor site, cloud), type of data included, and any integration should be collected. The amount of information gathered needs to be weighed between the level of detail desired and the work effort to keep the information current through any changes made during the application's lifecycle. Old, out-of-date information is worse than no information since decisions may be made on bad data.

The application list and associated information can be kept in a spreadsheet, a database, a web-based tool, or an application purchased specifically for this purpose. Something other than a spreadsheet would allow for including documents such as Service Level Agreements (SLA), Memorandums of Understanding (MOU), architecture designs, integration mapping documents, and/or help desk documentation.

The APO is responsible for managing each application as it moves through the lifecycle outlined in this book. They work with the requestors for new applications and ensure each new request is reviewed and objectively evaluated while gathering additional information so that the governance board can make an educated approval decision. If approved, APO works with the Project Management Office (PMO) to begin the implementation project once all resources are available. Once the new

application is live and in use, the office facilitates the support and maintenance work until the application is ready to be disposed. The office will then facilitate the work required to dispose of the application and all its components. It is important to note that the APO has a facilitation and control role within the organization and has a dotted line reporting relationship with other teams.

Throughout this book, a case study project will be used to help describe the concepts being discussed. The initial problem is identified below.

The current hospital procedures state that after a staff nurse has pulled a medication for administration, the nurse should manually verify the right drug, the right dose, the right route, and the right time against the electronic medical record contained in the hospital's EHR. The nurse must also verify the correct patient based on existing wristband practices. Finally, the nurse charts the medication as given in the EHR. While seemingly straightforward, the number of manual processes can lead to human error, and hospital records have shown several close calls in just the past 12 months. On average, there have been 15 errors for every 5,000 medication administered. These errors could have resulted in severe adverse reactions, which makes it essential to deploy a new application that can help reduce the likelihood of these occurrences. A Bar Code Medication Administration (BCMA) system will help confirm the five patient rights of medication administration.

Chapter 2

Project Management Overview

If you fail to plan, you plan to fail.

– Vijay Dhameliya

Before we can evaluate the process of managing applications using a portfolio framework, it's important to understand the basic principles of project management, which in turn will drive any new implementation or modification to a successful endpoint. It's important to note that the concept of project management is not new, nor is it restricted to the world of information technology (IT) or application management. In fact, you can find principles of project management in everything from the construction of ancient monuments to the planning of your best friend's wedding. Regardless of its usage, the basic definition of a project remains the same: It is a temporary endeavor that seeks to accomplish a unique set of objectives. The two key terms here are *temporary* and *unique*. A project must have a defined start and finish date; an effort that is ongoing cannot be a project. Also, each project should have a set of objectives to create a unique product or service. Requirements cannot be open-ended, and by the end of the project, you must be able to state definitively whether your objectives were met.

Some organizations may incorrectly refer to open-ended, long-term efforts as "projects," but it's important to distinguish projects from other disciplines, such as *programs* or *product management*. A program is a series of interrelated projects that serve to meet a long-term goal. For example, a large hospital system may wish to implement barcode