



Centers for Medicare & Medicaid Services

## CMS XLC Guidelines for Agile Projects

v 1.0

09/03/2014

## Record of Changes

Version	Date	Author/Owner	Description of Change
1.0	09/03/2014	XLC Steering Committee	Formal release of baseline version 1.0

# Table of Contents

- 1. INTRODUCTION ..... 1**
- 2. PURPOSE..... 1**
- 3. SCOPE..... 2**
- 4. INTENDED AUDIENCE..... 2**
- 5. ASSUMPTIONS/CONSTRAINTS ..... 2**
- 6. DEFINITION AND OVERVIEW ..... 4**
  - 6.1. XLC Overview .....4
  - 6.2. XLC Artifacts .....5
  - 6.3. Agile Overview .....6
  - 6.4. Scrum Overview .....6
- 7. XLC GUIDELINES FOR AGILE PROJECTS ..... 8**
  - 7.1. Agile Onboarding Phase .....10
    - 7.1.1. Onboarding Phase Synopsis .....10
    - 7.1.2. Onboarding Guideline Example .....12
  - 7.2. Release Planning Phase .....14
    - 7.2.1. Release Planning Phase Synopsis .....14
    - 7.2.2. Release Planning Guideline Example .....18
  - 7.3. Sprint Planning Phase .....24
    - 7.3.1. Sprint Planning Phase Synopsis .....25
    - 7.3.2. Sprint Planning Guideline Example .....25
  - 7.4. Sprints Phase .....28
    - 7.4.1. Sprint Phase Synopsis .....28
    - 7.4.2. Sprint Guideline Example .....31
  - 7.5. Implementation Phase .....33
    - 7.5.1. Implementation Phase Synopsis .....33
    - 7.5.2. Implementation Guideline Example .....34
  - 7.6. Operations and Maintenance Phase .....35
    - 7.6.1. Operations and Maintenance Phase Synopsis .....35
    - 7.6.2. Operations and Maintenance Guideline Example .....36
  - 7.7. Disposition Phase .....37
    - 7.7.1. Disposition Phase Synopsis .....37
    - 7.7.2. Disposition Guideline Example .....37
- 8. INFORMATION AND ASSISTANCE..... 38**

## List of Figures

Figure 1: CMS XLC.....	5
Figure 2: Scrum Framework.....	7
Figure 3: CMS XLC Framework for Agile IT Projects.....	9
Figure 4: Onboarding Phase.....	10
Figure 5: Release Planning Phase.....	14
Figure 6: Waterfall vs Agile Release Planning.....	15
Figure 7: Agile Sequential Releases.....	16
Figure 8: Agile Overlapping Releases.....	17
Figure 9: PRRS – PPA Release 1.0.....	22
Figure 10: PRRS – PPA Release 1.1.....	23
Figure 11: Sprint Planning Phase.....	24
Figure 12: Sprints Phase.....	28
Figure 13: Implementation Phase.....	33
Figure 15: Operations and Maintenance Phase.....	35
Figure 16: Disposition Phase.....	37
Figure 17: CMS XLC Agile - Complexity Level 3.....	41
Figure 18: CMS XLC Agile - Complexity Level 2.....	42
Figure 19: CMS XLC Agile - Complexity Level 1.....	43

## List of Tables

Table 1: PRRS Prioritized Product Backlog.....	19
Table 2: PRRS Sprint Backlog.....	26
Table 3: Acronyms List.....	39
Table 4: Glossary.....	44

# 1. Introduction

---

The Centers for Medicare & Medicaid Services (CMS) is committed to the strengthening of its System Development Life Cycle (SDLC) processes. Given the need to respond quickly to business demands, CMS Office of Information Services (OIS) created the CMS Expedited Life Cycle (XLC) as a streamlined model to guide and coordinate Information Technology (IT) Projects.<sup>1</sup>

The XLC provides a flexible approach to IT project execution and governance that is directly associated with each project's complexity. The XLC includes three tailored options to accommodate IT projects of varying complexity. The primary purpose of these XLC options is to balance speed and oversight in a manner commensurate with the complexity and risk associated with a particular IT project. The XLC model promotes agility, effective project review, and establishes appropriate oversight early in the process, increasing predictability and efficiency while reducing risk.

The XLC has been designed as a methodology independent framework that can accommodate multiple approaches to software development. Though the predominant software development approach within CMS has traditionally been Waterfall, the agency acknowledges the benefits to other methodologies and software development approaches such as Agile.

Agile is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Agile promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.

As CMS Integrated Project Teams (IPT) begin to adopt and plan for an Agile software development approach to IT projects, guidance is needed to show how IPTs can adopt Agile development methodology while complying with the benefits of the XLC framework. This guideline document has been created with the intent to promote the alignment and application of both the XLC and Agile using concepts of one of the most widely employed frameworks, Scrum, to successfully implement CMS IT projects.

## 2. Purpose

---

The purpose of this document is to assist CMS IPTs that elect to utilize an Agile software development approach to produce and implement CMS IT solutions. This guide outlines how

---

<sup>1</sup> For a comprehensive overview of the CMS XLC process please refer to the *CMS Expedited Life Cycle Process: Detailed Description Document* (<http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/XLC-DDD.pdf>)

IPTs can comply with XLC requirements while concurrently using iterative development methods. This document provides an example illustrating the practice of a Scrum like software development framework using the XLC.

### 3. Scope

---

The scope of this document is to provide guidelines and examples illustrating the application of CMS XLC while utilizing an Agile development approach. This guide is not intended to become policy, procedure, or directive. It is also not meant to be prescriptive; rather, it will demonstrate that through XLC tailoring, the XLC framework is flexible enough to accommodate various development approaches and will support IPTs with alignment of Agile and the XLC.

### 4. Intended Audience

---

This document is intended for use by all relevant stakeholders of CMS IT projects incorporating Agile software development methods with the XLC. Stakeholders include, but are not limited to; project managers, business sponsors, IT personnel, analysts, business units, contractors and all current and future CMS governing boards.

### 5. Assumptions/Constraints

---

The following is a list of assumptions and constraints for both this guideline document as well as the successful use of Agile within the CMS Enterprise:

#### **Assumptions:**

1. This guide is not an Agile training document nor is it a software development process document. This document provides guidance on following the XLC while adhering to Agile principals. All the phases of software engineering and development (including architecture, design, security, testing, etc.) should be followed per CMS and industry standards. Security must also follow the CMS Acceptable Risk Safeguards (ARS) and the Risk Management Handbook (RMH)<sup>2</sup>.
2. This guideline indicates the use of a Scrum-like methodology, as opposed to pure Scrum. Scrum has set rules that must be followed, whereas the Scrum methodology presented within this guideline document has been tailored for CMS.
3. The XLC provides flexibility via tailoring of Gate Reviews and Artifacts in order to accommodate an Agile development approach.

---

<sup>2</sup> RMH Volume I Chapter 1, Risk Management in the XLC explains the required risk management processes and when to perform them during the phases and reviews of the XLC. (<http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/InformationSecurity/Information-Security-Library.html>)

4. This guide is not intended as a training artifact and assumes a certain level of familiarity with the CMS XLC, Agile, and Scrum framework by the intended audience.
5. The guidelines provided within this document are not intended as policy or procedure. The guidelines are also neither directive nor prescriptive. This fact notwithstanding, appropriate references regarding directive or prescriptive documents and where to find them are included for clarity and understanding.”
6. This guide does not prescribe Agile software development methodology over Waterfall or any other methodology.
7. There are many Agile methodologies available for software development such as Dynamic System Development Method (DSDM), Feature Driven Development (FDD) and Extreme Programming (XP). This guide does not dictate or mandate use of the Scrum framework as the CMS approved Agile methodology.
8. Business Owner participation in the Sprints is essential for success!
9. This document is limited to providing XLC guidelines using concepts from Scrum, one of the most popular Agile frameworks.
10. Any contract related issues and contract modifications are out of scope of this document.
11. This guideline is a dynamic artifact and is therefore subject to improvement based on feedback and recommendation.

### **Constraints:**

1. CMS IT contractors are required to adhere to the XLC regardless of the software development methods used by contractors to produce CMS IT solutions.
2. Some CMS IT contracts may imply contractual use of Waterfall methodology for software development.
3. Distributed ownership amongst various contractors of XLC artifacts and processes may present challenges to an Agile methodology.
4. Scrum emphasizes the integration and co-location of Scrum Team members. This becomes problematic in the CMS environment where CMS and contractors are located in disparate geographical locations.
5. Misperception that the XLC only supports traditional Waterfall development methodology, though the framework is methodology independent.
6. There may not be adequate training within the CMS organization for Agile and Scrum methods which could limit business participation.

## 6. Definition and Overview

---

The following sections provide a definition and high level overview of the XLC, Agile, and Scrum.

### 6.1. XLC Overview

The XLC provides a flexible approach to IT project execution and governance that is directly associated with each project's complexity. The XLC includes three tailored options to accommodate CMS IT projects of varying complexity. The primary purpose of these XLC options is to balance speed and oversight in a manner commensurate with the complexity and risk associated with a particular IT project. Under this model, project risk is assessed and a Complexity Level of 1, 2, or 3 is assigned. The XLC varies the number of gate reviews depending on the project's risk, as shown in Figure 1 on page 5.<sup>3</sup>

Information System Risk, which is different from project risk, is also integrated into the XLC through RMH Volume I Chapter 1 Risk Management in the XLC.

---

<sup>3</sup> For a comprehensive overview of the CMS XLC process please refer to the CMS Expedited Life Cycle Process: Detailed Description Document (<http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/XLC-DDD.pdf>)

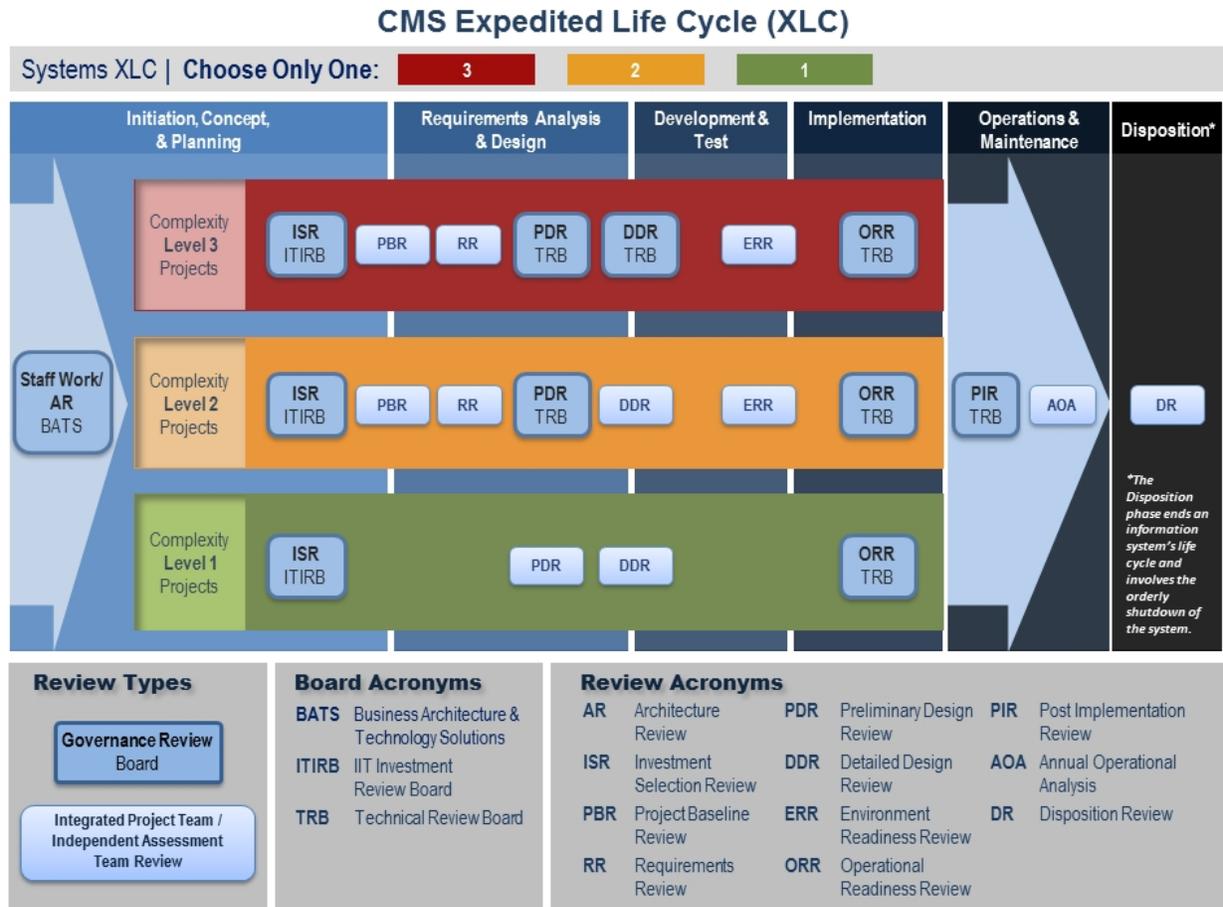


Figure 1: CMS XLC

## 6.2. XLC Artifacts

The Project Process Agreement (PPA) is a key XLC artifact that sets expectation and increases overall project predictability. It is a written agreement between all relevant project stakeholders that establishes which reviews, artifacts, and tests are required based on the project’s complexity level. The PPA provides IPTs a mechanism for process tailoring in order to combine, waive, or substitute artifacts, reviews, and tests. Tailoring requests must be documented within the PPA, accompanied by suitable justification from the IPT, and approved by the appropriate CMS governing board.

In addition to the PPA, there is a complete CMS XLC Artifact Matrix which identifies all documents that may be required throughout the life cycle of a CMS IT Project. For further detail regarding the XLC, PPA, and the Artifact Matrix, please refer to the CMS XLC website (<http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/index.html>).

## 6.3. Agile Overview

Agile is a way of thinking about iterative software development that increases the frequency of feedback loops and delivery cycles. Agile is neither a process nor a system development life cycle, rather, it is a paradigm to implement several iterative software development methodologies and frameworks such as Scrum, Feature Driven Development (FDD), and Extreme Programming (XP).

The base construct for an Agile project is iterative; empowering the self-organizing, cross-functional team to create working, tested, and potentially shippable code. Agile increases productivity and return on investment (ROI) at reduced technical cost and schedule risk by shortening the feedback cycle and managing change effectively.

In February 2001, the Manifesto for Agile Software Development<sup>4</sup> was published to define the approach now known as Agile Software Development.

The Agile Manifesto identifies four main values:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

*That is, while there is value in the items on the right, Agile values the items on the left more.*

## 6.4. Scrum Overview

Scrum, as shown on page 7 in Figure 2, is an iterative and incremental Agile software development framework for delivering software products of the highest possible value. Scrum focuses on a flexible development strategy and uses fixed length iterations, called Sprints, to build products incrementally. Scrum advocates the self-organization of co-located teams to improve verbal communication among all team members.

A key principle of Scrum is its recognition of scope change during a project and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an approach that focuses on maximizing the team's ability to deliver quickly and respond to emerging requirements.

Scrum defines three roles: a **Product Owner** who prioritizes a list of user stories, a cross functional **Scrum Team** whose only objective is to produce working software incrementally in

---

<sup>4</sup> <http://agilemanifesto.org/>

time boxed Sprints (usually 2 to 4 weeks), and a *Scrum Master* who keeps the team focused on its goal.

Three Scrum ceremonies are required: *Sprint Planning* where the team pulls user stories to be completed in a Sprint from the top of the prioritized list, *Daily Standup* to assess Sprint progress, and *Review and Retrospective* at the end of each Sprint to demonstrate the product, assess lessons learned, and incorporate improvement in upcoming sprints.

Three key artifacts are maintained: *Product Backlog* which contains a prioritized list of user stories, *Sprint Backlog* which contains work to be completed in the current Sprint, and a *Burndown Chart* which depicts the graphical representation of remaining task hours in one Sprint and can be used by teams for future capacity planning.

CMS XLC guidelines presented in this document assume that Scrum framework concepts are used as the Agile software development approach.

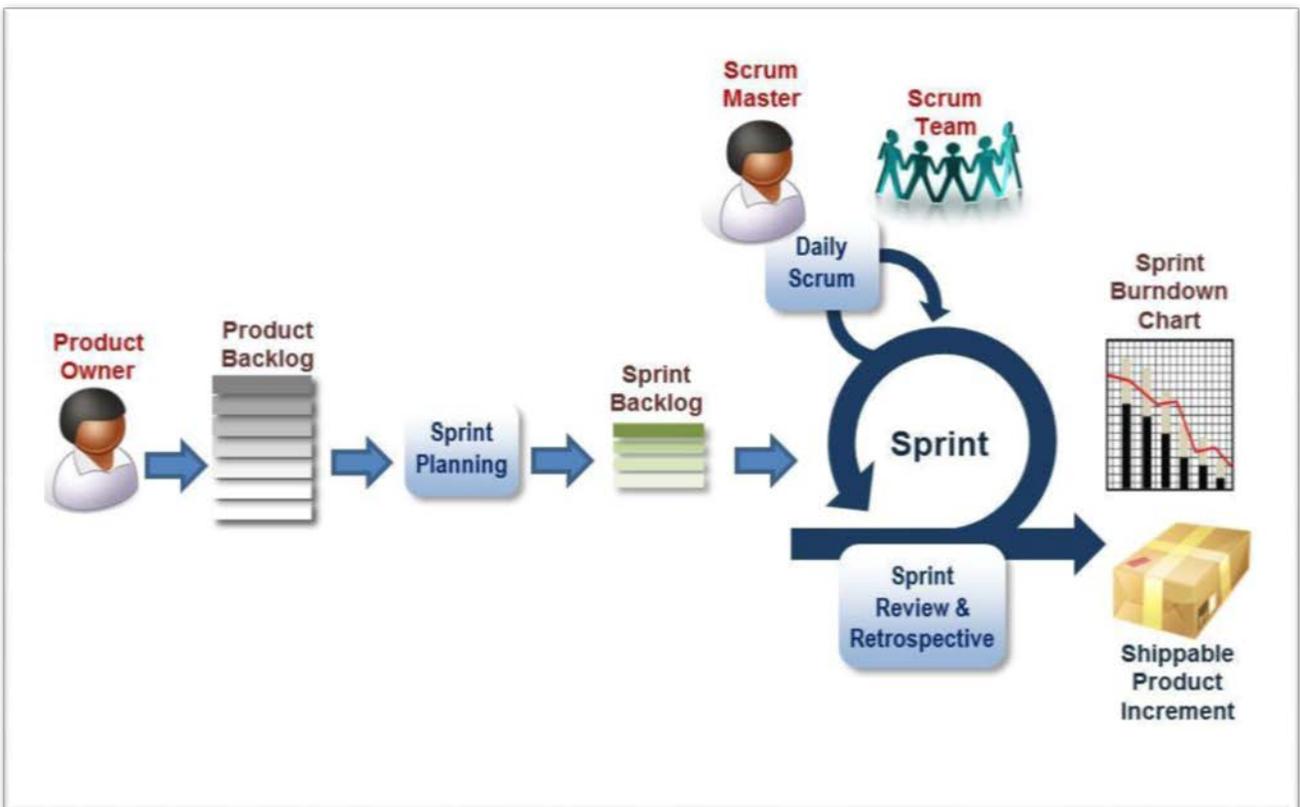


Figure 2: Scrum Framework

## 7. XLC Guidelines for Agile Projects

---

The CMS XLC must be adhered to by all CMS IT Projects developing CMS IT solutions. The XLC defines an IT Project as

*A temporary endeavor undertaken to create a unique information technology product, service, or result (e.g., an automated system). An IT project must have specific starting and ending dates, well-defined objectives and constraints, established responsibilities, and a budget and schedule. An IT project may be self-contained or may be part of a larger project. An IT project further refers to a project that uses, collects, manipulates, transfers, stores, or automates information.*

CMS IT Projects applying an Agile approach to software development must also follow the CMS XLC which includes completion of all gate reviews and artifacts as listed in an approved PPA. If there are acceptable artifact substitutes (e.g., well documented user stories as opposed to formal requirements documents) they must be identified in the PPA. To achieve this requirement, the CMS XLC Framework for Agile IT Projects (Figure 3 on page 9) has been developed to create alignment between the XLC and Agile.

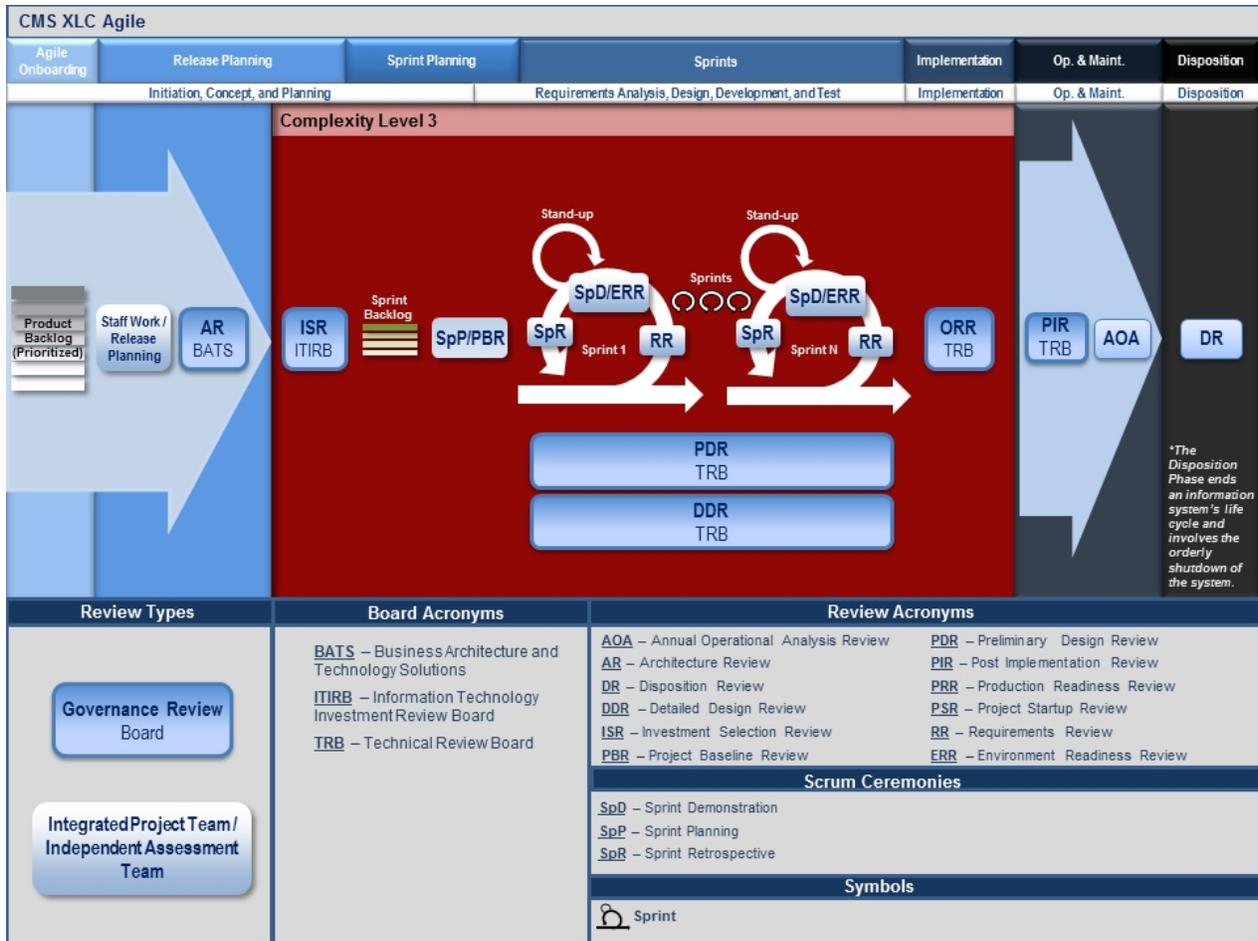


Figure 3: CMS XLC Framework for Agile IT Projects

This document presents general guidelines and best practices to CMS IT Projects for use in incorporating an Agile approach and Scrum like framework within the XLC. Although these guidelines illustrate the specific use of XLC and Scrum, the XLC can be leveraged to support other software development methods.

A walk through of a fictitious CMS IT project, as introduced below, is presented as an example to provide clarity throughout each life cycle phase.

**Example Introduction**

The Office of Information Services (OIS) within CMS has determined a business need for a public facing website which will serve to provide a central rating and review system for outpatient hospitals, inpatient hospitals, physicians, dialysis facilities, and nursing homes. The website will be called KnowYourProvider.gov, with the goal being to provide greater transparency into healthcare facilities for the public.

The project name is PRRS (Provider Rating and Review System), and the Business Owner of the project is the Center for Clinical Standards and Quality (CCSQ). Susan Jones was assigned as

Business Lead to implement this project and John Smith is assigned as Government Task Lead (GTL) or technical lead. OIS has decided to use an Agile Development approach for PRRS. An application development contractor, Better Software Inc. (BSI), with extensive experience in Scrum methodology was awarded this contract for design, development and implementation of PRRS.

## 7.1. Agile Onboarding Phase

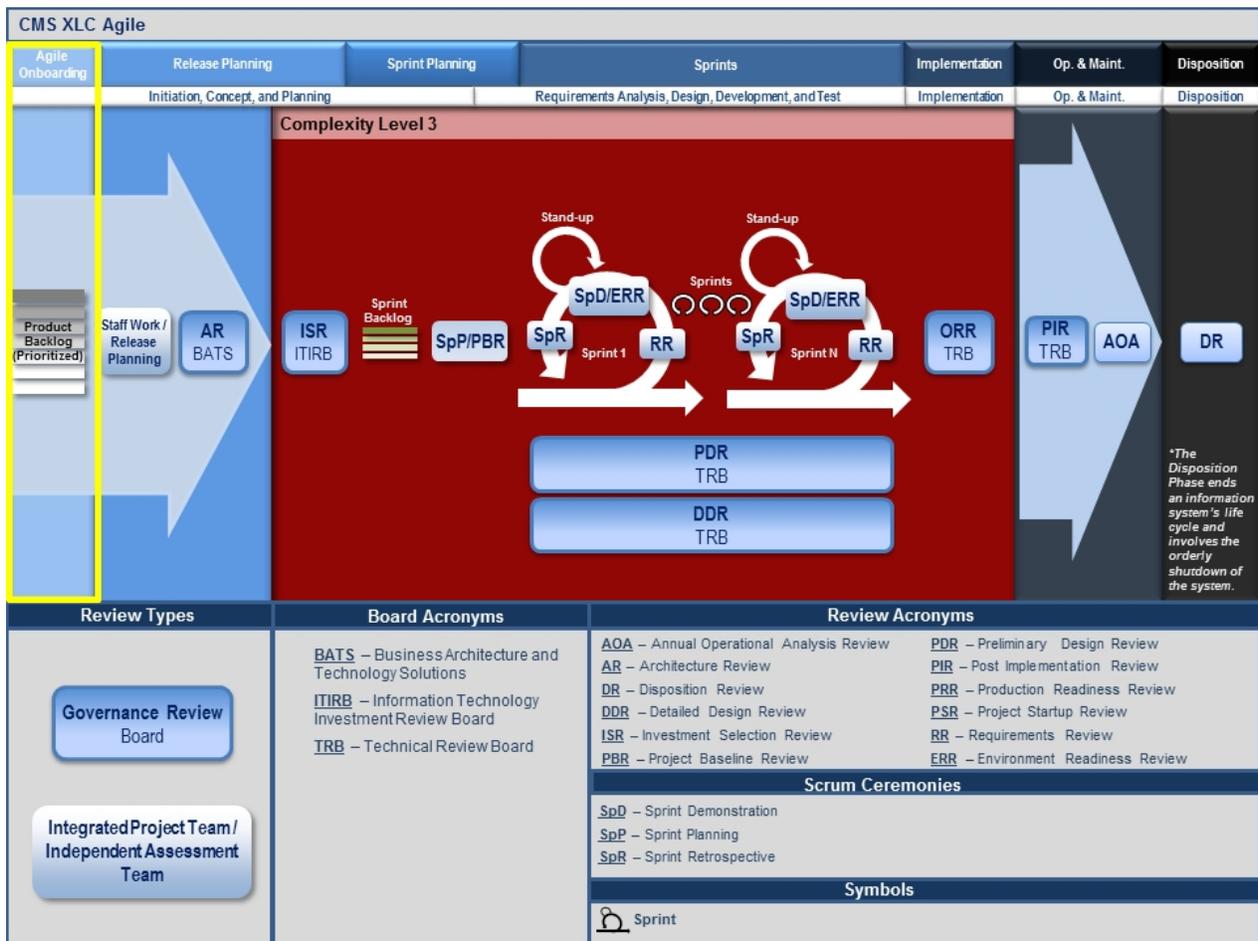


Figure 4: Onboarding Phase

### 7.1.1. Onboarding Phase Synopsis

The first stage of the CMS XLC Agile framework is the Onboarding Phase. The following roles and teams are established with Scrum framework being utilized.

1. **Product Owner:** The Product Owner is the voice of the customer. Within CMS, the Product Owner is analogous to the Business Owner or identified proxy for the Business Owner. If a proxy is used, they must be empowered to make decisions on behalf of the business. The Product Owner (or proxy) is accountable for ensuring that the team delivers value to the business.

2. **Scrum Team:** The Scrum Team is self-organizing and its objective is to deliver potentially shippable increments (PSIs) of product code through time boxed Sprints. Within CMS, the Scrum Team is a subset of the IPT and is typically made up of individuals with cross-functional skills who perform the actual work (analyze, design, develop, test, technical communication, document, etc.).
3. **Scrum Master:** The Scrum Master is accountable for alleviating impediments to the Scrum Team in their effort to successfully achieve product goals and deliverables. The Scrum Master is not a traditional Team Lead or Project Manager, but acts as a buffer between the Scrum Team and any distracting influences. For CMS purposes, the Scrum Master role will vary project-to-project. In some cases, the Scrum Master may be a CMS employee while other times it may be a member of the Application Development Organization (ADO). The Scrum Master is the enforcer of the rules of Scrum, chairs key meetings, and challenges the Scrum Team to improve. Per Scrum, it is preferred that the Scrum Master is co-located with the Scrum Team. The Scrum Master role is sometimes referred to as a servant-leader.
4. **IPT:** The IPT is not one of the three Scrum defined roles. IPT is defined within the CMS XLC as an organized, cross-functional group of individuals collectively responsible for the specific purpose of delivering a product to an internal or external customer. For CMS purposes, the Scrum Team can be thought as a subset of the IPT. The IPT is the extended team that includes the Scrum Team and other relevant stakeholders. The Scrum Team focuses on development activities with the output being potentially shippable increments of product to the business. The IPT focuses on guiding the Product Owner through the XLC gate reviews and artifacts, attends demonstrations of potentially shippable increments, and provides feedback.

The **Product Backlog**, the first of three **Scrum Artifacts**, is created during the Onboarding Phase. The Product Backlog is a prioritized list of desired functionalities, in the form of user stories, which are maintained for an IT solution. User stories are commonly written in the format, “*as a <type of user>, I want <some goal> so that <some reason>.*” User stories consist of new functionalities, enhancements, bug fixes, non-functional requirements, etc. - whatever must be accomplished in order to successfully deliver a working software product.

During onboarding, the Scrum Team authors the user stories. The Scrum Team then determines how the user stories will be scored and assigns a value to each story. The Product Owner then force ranks and prioritizes the user stories on the Product Backlog based on considerations such as risk, business value, dependencies, and date needed. The user story format helps the Product Owner prioritize the Product Backlog.

Though the XLC does not specifically define or identify Scrum artifacts such as the Product Backlog, the flexibility of the PPA allows for IPTs to classify and include additional project specific artifacts as needed.

## 7.1.2. Onboarding Guideline Example

The PRRS Product Owner, Susan, in consultation with various business units, compiles a high level list of requirements for PRRS:

1. Facility Information (name, contact, overview, patient acceptance criteria, facility history information)
2. Facility Score (standardized rating system with score and reviewer comments)
3. Explanation of How Score is Calculated
4. Geographical Search (name, state, zip code)
5. National Rank Filter Based on Score
6. Regional Rank Filter Based on Score
7. National Facility Rank Filter Based on Score and Type of Facility
8. Regional Facility Rank Filter Based on Score and Type of Facility
9. Mobile Device Compatibility
10. User's ability to provide feedback and rating
11. CMS's ability to approve user feedback prior to online posting
12. CMS's ability to capture and view web analytics
13. Compatibility with all major web browsers (IE, Safari, Firefox, Chrome)
14. 508 compliant website
15. User's ability to increase or decrease the text size
16. User's ability to share , bookmark or email particular universal resource locators (URLs)
17. User's ability to sign up for email updates

Product Owner Susan completes and submits an IT Intake Request Form for approval to begin the XLC process. BSI assembles a Scrum Team as a subset of the IPT and identifies Jason Harris, a Certified Scrum Master (CSM), as the Scrum Master for PRRS.

The Scrum Team begins to transform high level requirements into user stories, each with associated acceptance criteria which provides the Scrum Team with an understanding of what it will take to satisfactorily achieve the goal of each user story. The acceptance criteria may also

include non-functional technical requirements such as performance, privacy and security requirements.

The first user story is written: *“As a Product Owner, I want all Facility Information attributes stored within PRRS, so that all users can search the data based upon selected criteria.”* The associated acceptance criteria for this user story is defined as, *“All facility information attributes have been populated and can be searched via the website by the end users.”* The acceptance criteria may also include technical requirements such as *“The response time to display all Facility Information attributes shall be 2 seconds or less”*. The Scrum Team continues to author user stories in this manner for all of the high level requirements until the Product Backlog is complete.

Once the user stories are authored, Scrum Master Jason suggests the Team use Planning Poker to assign point values to each of the user stories. The Scrum Team approves of Jason’s suggestion of Planning Poker and each member is provided with an identical deck of cards that contain story point values (½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ∞). Individual user stories are presented by Product Owner Susan for estimation. After a period of discussion, each participant chooses a card from their own deck that represents their estimate of how much work is involved in the story under discussion. All estimates are kept private until each participant has chosen a card. Estimates are then revealed and discussion ensues. Scrum Team members with high estimates and low estimates are given an opportunity to provide justification for their assessment. Once a consensus has been reached by the Team, the user story is scored and the next story is presented. This process continues until all user stories have been assigned a point value.

With point values assigned, Product Owner Susan reviews the scored user stories for KnowYourProvider.gov\_ which she then prioritizes within the Product Backlog. This force ranked list establishes the overall scope of the project.

This concludes the Agile Onboarding Phase. The next phase is Release Planning which determines how the Product Owner and the Scrum Team will breakdown the overall scope into multiple releases in order to deliver the intended solution.

## 7.2. Release Planning Phase

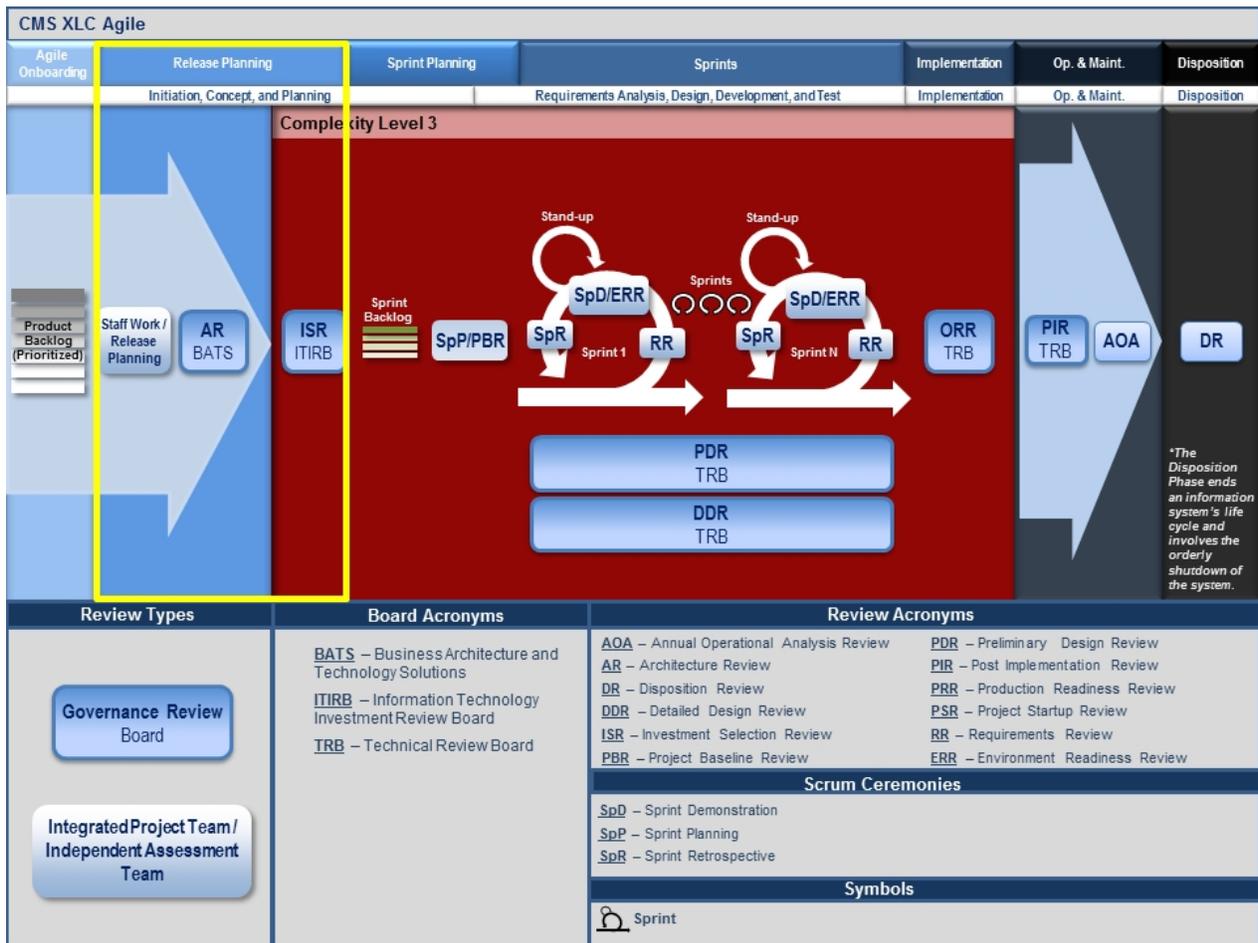


Figure 5: Release Planning Phase

### 7.2.1. Release Planning Phase Synopsis

During the Release Planning Phase, the Product Owner and Scrum Team attend a kickoff/planning session to review the prioritized Product Backlog and determine the number of release(s) required to complete the scope of the IT Project. The number of releases will be dependent upon the extent of the Product Backlog, availability of resources, and business need. For CMS, it is recommended that Agile releases should be no longer than six months in duration, as longer release duration can increase project risk.

The Scrum Team then defines the fixed length Sprint duration for all Sprints within the project. For CMS, it is recommended that a Sprint should be four weeks but no longer than six weeks in duration. The Scrum Team also estimates team velocity which identifies how many story points can be accomplished per fixed length Sprint. These initial numbers will be estimates and become more accurate as historical data is gathered by the project team.

The planning session(s) will be used to determine high level system infrastructure and security related logistics that are required during XLC AR such as: platform, system hosting, connectivity requirements, modes of operation, assurance level, e-authentication level, authorization, and encryption.

Unlike a Waterfall approach which places functionality into production through one large, prolonged release, Agile development places targeted functionality into production through multiple, condensed releases. Shorter release cycles ensure that feedback loops are reduced and that lessons learned from one release are easily incorporated into subsequent releases. Figure 6 compares feedback opportunities between Waterfall and Agile. Waterfall feedback may only occur once at the end of a 12-month release cycle, whereas Agile offers multiple feedback opportunities at the end of each Sprint and release during the same 12 month period.

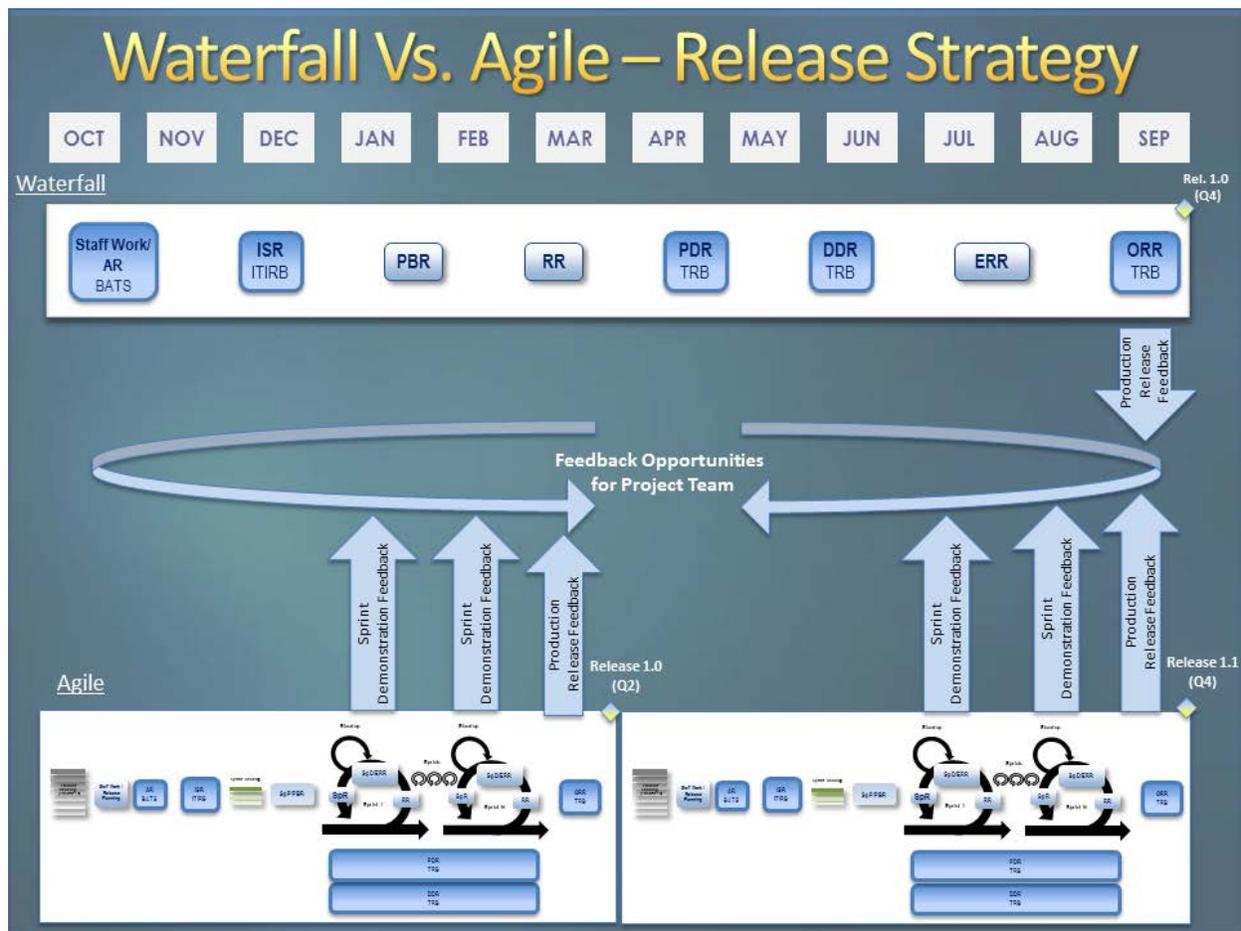


Figure 6: Waterfall vs Agile Release Planning

IT Projects which identify the need for multiple releases have the option to plan release cycles in either a sequential or overlapping order. In a sequential release cycle (Figure 7 on page 16), one release completes the full XLC prior to the initiation of the next release. In an overlapping release cycle (Figure 8 on page 17), multiple releases are in flight simultaneously, but at varying phases of the lifecycle. For both release types, it is imperative that the appropriate availability and use of resources across the lifecycle of a project are accounted for.

Regardless of the release strategy chosen, each production release is considered a distinct IT project which will require a separate PPA to identify tailoring of gate reviews, artifacts, and tests specific to the needs of that particular release.

Figure 7 displays an example of an Agile project employing three sequential production releases, each of four months in duration.

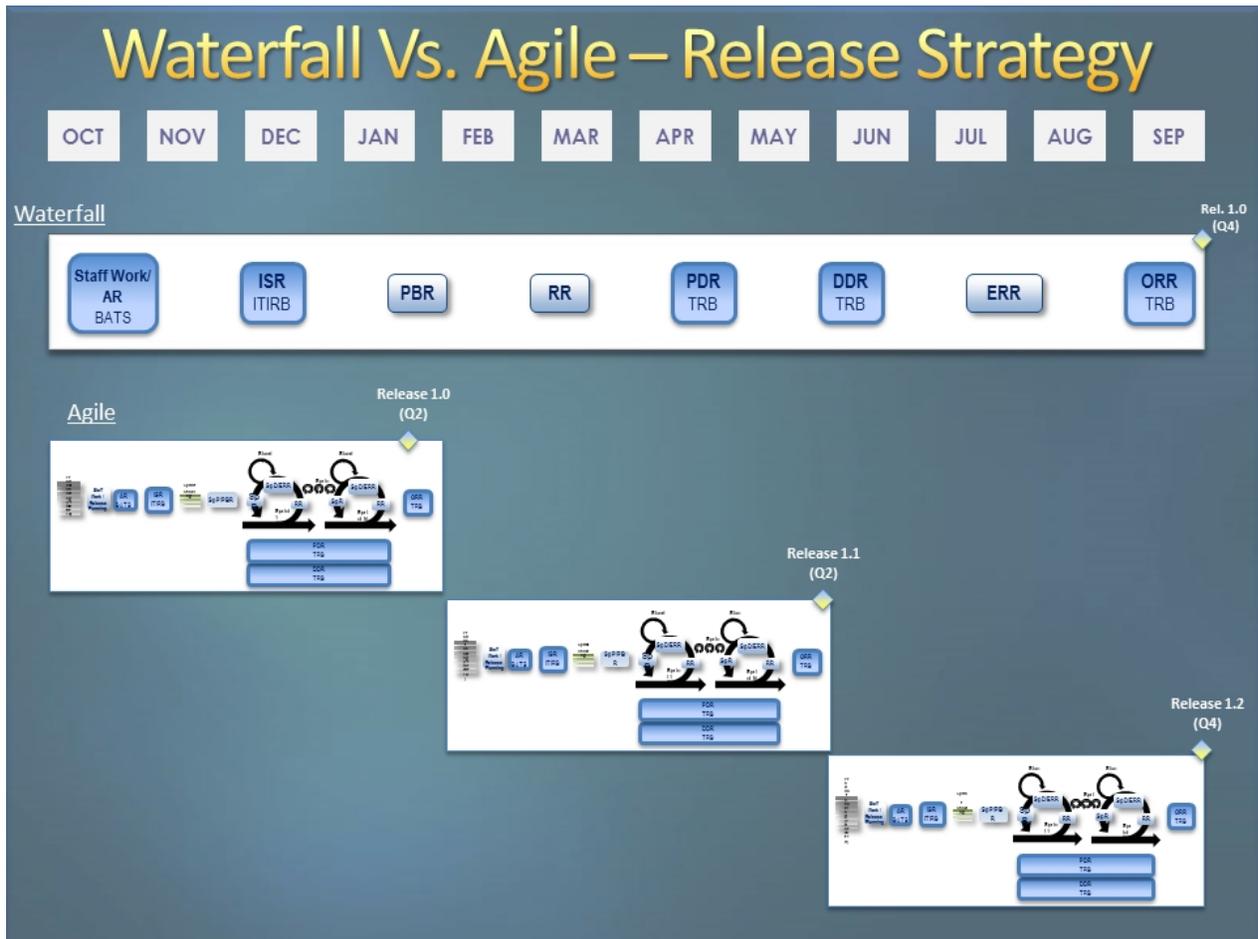


Figure 7: Agile Sequential Releases

Figure 8 depicts an example of an Agile project using three overlapping production releases, each of 6 months in duration.



Figure 8: Agile Overlapping Releases

Release strategies will always vary by individual project need. Smaller, less complex projects may only require one release. For longer, more complex projects, multiple releases are recommended to manage risk effectively. Scrum Teams should choose the most logical and appropriate release strategy that meets the needs of the Team and maximizes feedback opportunities. A new ATO may be required for each release, some releases, or only the first release depending on the sequencing of the development and implementation of security requirements.

The Scrum Team confirms the scope of a particular release by selecting prioritized user stories from the Product Backlog that will be implemented during that release. Upon conclusion of the first release, the Scrum Team will come back to the Release Planning Phase to select the next set of user stories from the Backlog to be implemented in the next release.

### 7.2.1.1 Architecture Review and Investment Selection Review

The IPT must attend all XLC gate reviews as listed and approved within the PPA per release. The Release Planning Phase includes the AR with the Business Architecture and Technology Solutions (BATS) Board and the Investment Selection Review (ISR) with the Information Technology Investment Review Board (ITIRB).

The purpose of the AR is to determine whether the proposed project potentially duplicates, interferes, contradicts or can leverage another investment that already exists, is proposed, under development, or planned for near-term disposition. The business need is assessed to determine if it is sound and conforms to the CMS Technical Reference Architecture (TRA).

The purpose of ISR is to determine if it is a sound, viable, and worthy of funding, support and inclusion in the organization's IT Investment Portfolio. The business need and objectives are reviewed to ensure the effort supports CMS' overall mission and objectives and will not compromise initiatives on the horizon. This is an outward focused review designed to ensure funding and approval to proceed from CMS senior leadership.

If a project is employing a multi-release approach and there is no modification to the system architecture or business processes in subsequent releases, the AR may only need to be attended by the IPT during the first release. Similarly, the ISR may only occur once during the first release of a project if funding for the entire project has already been secured. Approval to perform, waive or combine reviews is always subject to PPA approval by the Technical Review Board (TRB).

Once the AR and ISR have successfully concluded, the Release Planning Phase is complete. The IPT will then move into the next Agile Phase, Sprint Planning.

### 7.2.2. Release Planning Guideline Example

Product Owner Susan holds a meeting with the Scrum Team to determine if the scope of the IT Project is too large and needs to be broken into multiple releases. After much review and discussion, it is agreed upon that the project will be broken down into two distinct releases; Release 1.0 and 1.1. This determination is based on the extent of the Product Backlog and an estimate of the Scrum Team's capacity. Each release will be a fixed 6 month duration. Due to resource constraints, the two releases will be deployed to production in sequential order with no overlap.

Table 1 depicts an abbreviated Product Backlog containing user stories that will be incorporated into Release 1.0, associated story points and priority. Upon conclusion of Release 1.0, the Scrum Team will come back to the Release Planning Phase to select the next set of user stories from the Backlog for implementation into Release 1.1.

Table 1: PRRS Prioritized Product Backlog

	Priority	User Story #	User Story	Story Points	Acceptance Criteria
Release 1.0	1	1	As a Product Owner, I want all facility information attributes stored within PRRS, so that all users can search the data based upon selected criteria.	20	All facility information attributes have been populated and can be searched via the website by the end users.
	2	13	As a Product Owner, I want the website to be compatible with all major internet browsers so that users can view and use website.	13	PRRS successfully completes browser compatibility testing with no outstanding findings.
	3	3	As a User, I want an explanation of how a score is calculated so that I can understand what each Provider is measured against.	2	An explanation of how the facility score is calculated is provided on the website.
	4	4	As a User, I want to be able to search a Provider by geographical region (name, state, zip code) so that I can locate Providers by location.	5	All Providers can be searched by the end user based on name, state or zip code.
	5	2	As a User, I want the facility score (standardized rating system with score and reviewer comments) to be provided so that I can accurately assess the Provider population.	20	Provider facility score is available to the end user via the website.
	6	14	As a Product Owner, I want the website to be 508 compliant so that disabled users are able to use the website.	13	PRRS successfully completes 508 testing with no outstanding findings.
	7	6	As a User, I want to be able to search Providers by regional rank and score so that I can find the best Provider on a regional level.	5	All Providers can be searched by the end user based on regional rank and score.
	8	15	As a User, I want to have the ability to increase or decrease the text size so that I can accurately read information displayed on the webpage.	2	PRRS successfully completes user acceptance testing (UAT) with no findings.

	Priority	User Story #	User Story	Story Points	Acceptance Criteria
	9	5	As a User, I want to be able to filter by national rank and score so that I can locate the best national Providers.	5	All Providers can be filtered by the end user based on national rank and score.
	10	7	As a User, I want to be able to filter by national facility rank based on score and type of facility to find the best Provider on a national level based on their facility type and location.	5	All Providers can be filtered by the end user based on national facility rank, score and type of facility.
	11	8	As a user, I want to be able to search by regional facility rank based on score and type of facility, so that I can locate the best regional facility based on their score and type.	5	All Providers can be searched by the end user based on regional facility rank and score.
	12	16	As a User, I want to be able to bookmark, share or email a webpage to someone so that they can access the same information I am seeing.	5	A user has the ability to bookmark, share or email a PRRS specific universal resource locator (URL).
Release 1.1 (Tentative)	13	17	As a User, I want to have the ability to sign-up for email updates so that I know the latest information pertaining to the website.	5	A user can sign-up for email updates.
	14	9	As a User, I want the system to be mobile device compatible so that I can access the system remotely on a phone, PDA or tablet.	20	The website is compatible with mobile devices, PDAs and tablets.
	15	10	As a User, I want to be able to provide feedback and ratings on Providers to increase transparency for other users.	13	End user feedback and ratings about Providers can be captured on the website.
	16	11	As a Product Owner, I want to have the ability to approve user feedback prior to online posting to ensure appropriate nature of the content.	8	Feedback can be reviewed and vetted prior to being posted online.

	Priority	User Story #	User Story	Story Points	Acceptance Criteria
	17	12	As a Product Owner, I want the capability for system analytics and reporting so that I can analyze site traffic and activity.	20	Web analytics are being captured and can be viewed and analyzed.
<b>Continued...</b>					

After estimating Scrum Team velocity, Scrum Master Jason and the team determine that five Sprints will be required for Release 1.0 in order to adequately implement the user stories. Each Sprint will be a time boxed four week duration. To achieve end to end product integration, a 6th integration testing Sprint of four week duration is added. This gives the Scrum Team six months to complete Release 1.0.

Since each production release is defined by the XLC as an independent IT project, both releases require a separate PPA. A preliminary PPA is drafted for Release 1.0 which identifies that PRRS will be a Complexity Level 3 project because it is a new system, introduces new business process models, and contains significant data and interface complexity. Preliminary artifacts, as identified within the CMS XLC Artifact Matrix, are also begun by the IPT ahead of the first XLC gate, Architecture Review (AR).

PRRS plans to attend AR only once during Release 1.0, as the business architecture will not change for Release 1.1. Therefore, the IPT plans to waive the AR for Release 1.1. This waiver request will be justified within the Release 1.1 PPA that will be prepared during the next release planning meeting. Similarly, since funding for PRRS will already be secured, the ISR will be waived in Release 1.1. Figures 9 (on page 22) and 10 (on page 23) demonstrate how, through the use of the PPA, reviews such as the AR and ISR will only occur during Release 1.0 of the project, subject to approval by the TRB.

PROJECT PROCESS AGREEMENT -- STAGE GATES		Directions: Enter information into green cells Review stage gate list & details in column A-D. Add any project defined reviews in row 14&15. Record agreement for the project's initial release in column E and F.			
Project Name		PRRS			
Project Description		CMS has determined a business need for a public facing website which will serve to provide a rating and review system for hospital outpatient, hospital inpatient, physician, dialysis facility, and nursing homes. The goal of the website will be to provide greater transparency into healthcare facilities for the public.			
Release Complexity Level		1 3			
STAGE GATE REVIEW		CMS Governance / Delegated to Projects			
#	STAGE GATE REVIEW (AR)	D	E	F	G
		STAGE GATE DEFINITION	PROJECT AGREEMENT	AUTHOR (specific for Project)	JUSTIFICATION and/or NOTES
1	Architecture Review (AR)	Maybe Delegated	Perform		
2	Investment Selection Review (ISR)	Governance	Perform		
3	Project Baseline Review (PBR)	Maybe Delegated	Perform		

Figure 9: PRRS – PPA Release 1.0

PROJECT PROCESS AGREEMENT -- STAGE GATES		Directions: Enter information into green cells Review stage gate list & details in column A-D. Add any project defined Record agreement for the project's initial release in column E and F.			
Project Name		PRRS			
Project Description		CMS has determined a business need for a public facing website which will serve to provide a central rating and review system for hospital outpatient, hospital inpatient, physician, dialysis facility, and nursing homes. The goal of the website will be to provide greater transparency into healthcare facilities for the public.			
Release		1.1			
Complexity Level		3			
A	B	C	D	E	G
#	STAGE GATE REVIEW	CMS Governance / Delegated to Projects	STAGE GATE DEFINITION	PROJECT AGREEMENT	CMS IT PROJECT JUSTIFICATION and/or NOTES
1	Architecture Review (AR) May be Delegated		Determine whether the proposed project potentially duplicates, interferes, contradicts or can leverage another investment that already exists, is proposed, under development, or planned for near-term disposition. The business need is assessed to determine if it is sound and conforms to the CMS Enterprise Architecture.	waive	This review is waived as it already occurred during the AR that took place during PRRS Release 1.0 on 10/15/13.
2	Investment Selection Review (ISR) Governance		Determine if it is a sound, viable, and worthy of funding, support and inclusion in the organization's IT Investment Portfolio. The business need and objectives are reviewed to ensure the effort supports CMS' overall mission and objectives and will not comprise initiatives on the horizon.	waive	This review is waived as funding has already been secured for this project through the ISR that took place during PRRS Release 1.0 12/15/13.
3	Project Baseline Review (PBR) May be Delegated		Obtain management approval that the scope, cost and schedule that have been established for the project are adequately documented and that the project management strategy is appropriate for moving the project forward in the life cycle. The PBR includes review of the budget, risk, and user requirements for the investment; emphasis should be on the total cost of ownership and not just development or acquisition costs.	Perform	
4	Requirements Review (RR) May be Delegated		Verify that the requirements are complete, accurate, consistent and problem-free; evaluate the responsiveness to the business requirements; ensure that the requirements are a suitable basis for subsequent design activities; ensure traceability between the business	Perform	

Figure 10: PRRS – PPA Release 1.1



### 7.3.1. Sprint Planning Phase Synopsis

Sprint Planning (SpP) is held at the beginning of each Sprint Phase to select what work will be accomplished during the Sprints. Prioritized user stories that were selected from the Product Backlog during Release Planning are further broken down into detailed tasks in order to more clearly define quantities of work. Sprint Planning involves the Scrum Team in order to accurately identify how long it will take to perform each task and how much work can be accomplished per Sprint. Sprint Planning can also be readdressed at the end of a Sprint to adjust and plan for the upcoming Sprint.

A ***Sprint Backlog*** is created by the Scrum Team. The Sprint Backlog contains user stories broken down into the detailed tasks the Scrum Team must address during the Sprints. The Scrum Master tracks the velocity of previous Sprints as data becomes available. In order to change a committed Sprint Backlog, the Product Owner and the Scrum Team must agree to the modifications in order to maintain team velocity.

Sprint Planning is performed in conjunction with XLC Project Baseline Review (PBR). PBR seeks to baseline the project and obtain management approval that scope, cost, and schedule established for the project are adequately documented. Sprint Planning defines the goals of the Sprint and required tasks needed to achieve those goals.

It is important to note that the Initiation, Concept, and Planning XLC Phases align with the Agile Onboarding, Release Planning, and Sprint Planning Phases as shown above in Figure 11 on page 24.

### 7.3.2. Sprint Planning Guideline Example

Upon conclusion of the Release Planning Phase, the PRRS Scrum Team begins to break down user stories selected for Release 1.0 into detailed work tasks. Under the guidance of Scrum Master Jason, a Sprint Backlog is created to capture all tasks required for completion of Release 1.0 user stories. Table 2 provides an example of the Sprint Backlog and tasks required to complete the first user story. In addition to task detail, the Scrum Team will also estimate hours required to complete each task and identify resources assigned to accomplish each task.

Table 2: PRRS Sprint Backlog

	User Story #	User Story	Sprint #	Tasks
Release 1.0	1	As a Product Owner, I want all facility information attributes stored within PRRS, so that all users can search the data based upon selected criteria.	1	<p>Create database entries for all applicable facility information values</p> <p>Create user input fields (text box, check box, drop down list) within UI</p> <p>Configure and point UI fields to database</p> <p>Link UI fields to Get_Facility_Info_Service_1.0</p>
	13	As a Product Owner, I want the website to be compatible with all major internet browsers so that users can view and use the website.	2	TBD
	3	As a User, I want an explanation of how a score is calculated so that I can understand what each Provider is measured against.	2	TBD
	4	As a User, I want to be able to search a Provider by geographical region (name, state, zip code) so that I can locate Providers by location.	2	TBD
	2	As a User, I want the facility score (standardized rating system with score and reviewer comments) to be provided so that I can accurately assess the Provider population.	3	TBD
	14	As a Product Owner, I want the website to be 508 compliant so that disabled users are able to use the website.	4	TBD
	6	As a User, I want to be able to search Providers by regional rank and score so that I can find the best Provider on a regional level.	4	TBD

User Story #	User Story	Sprint #	Tasks
15	As a User, I want to have the ability to increase or decrease the text size so that I can accurately read information displayed on the webpage.	4	TBD
5	As a User, I want to be able to filter by national rank and score so that I can locate the best national Providers.	5	TBD
7	As a User, I want to be able to filter by national facility rank based on score and type of facility to find the best Provider on a national level based on their facility type and location.	5	TBD
8	As a user, I want to be able to search by regional facility rank based on score and type of facility, so that I can locate the best regional facility based on their score and type.	5	TBD
16	As a User, I want to be able to bookmark, share or email a webpage to someone so that they can access the same information I am seeing.	5	TBD

In conjunction with this activity, The PRRS IPT holds a Sprint Planning Review/PBR. The IPT ensures compliance and accuracy of project scope as work proceeds through the life cycle. Upon confirmation of concurrence from the IPT, the Sprints Phase of the life cycle can begin.

In preparation for the Sprints Phase XLC gate reviews, the IPT begins to draft the artifacts as identified within the CMS Artifact Matrix and approved within the PRRS Release 1.0 PPA for the Preliminary and Detailed Design Reviews.

## 7.4. Sprints Phase

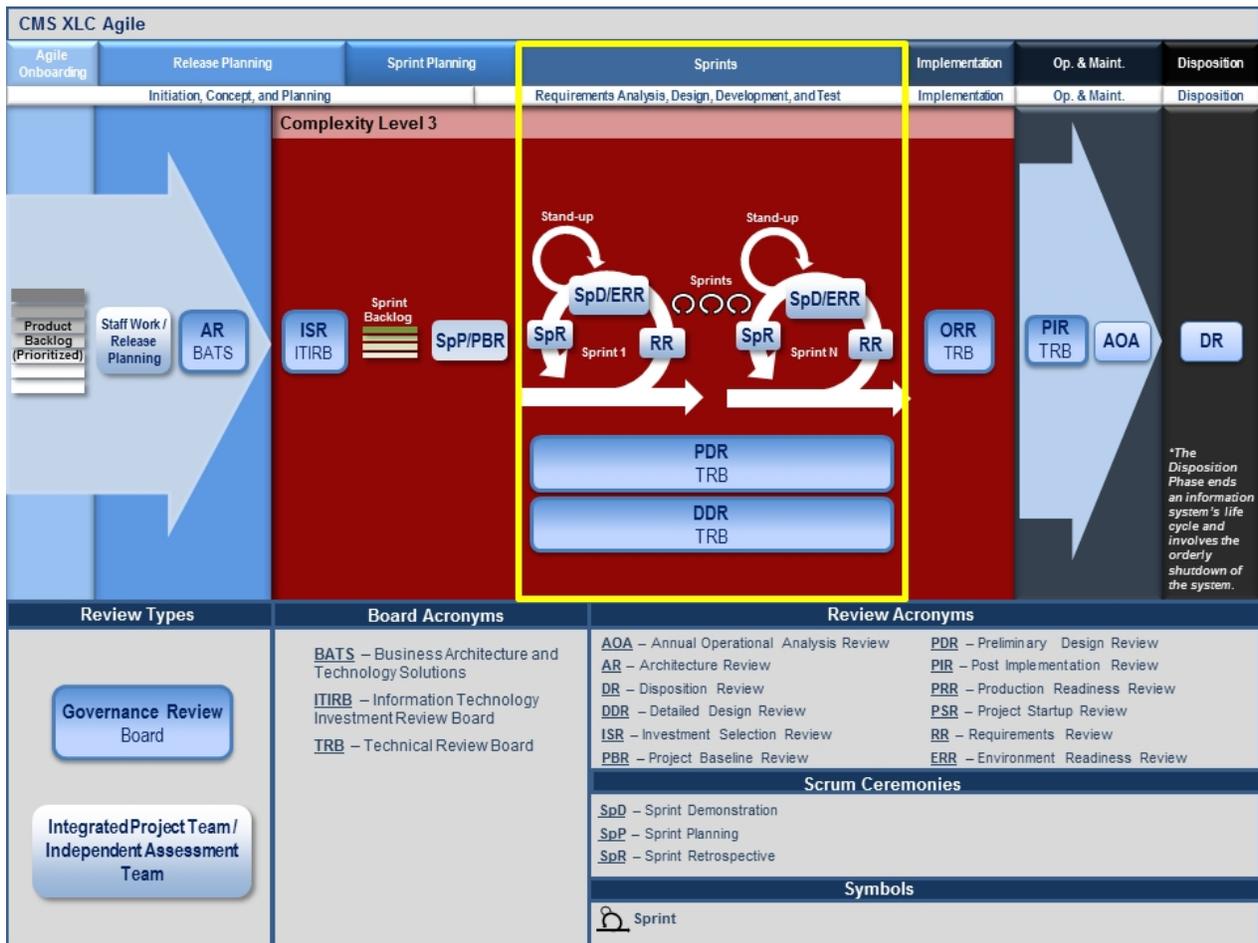


Figure 12: Sprints Phase

### 7.4.1. Sprint Phase Synopsis

The Sprint Phase encompasses repetitive cycles of the following traditional Waterfall Phases; Requirements Analysis, Design, Development, and Testing as depicted above in Figure 12. During each of these cycles, Requirements Review (RR), Sprint Demonstrations (SpD), Environment Readiness Review (ERR), and Sprint Retrospective (SpR) are performed. Every production release consists of a number of fixed length Sprints as determined during Release Planning and each Sprint ends with tested, production ready incremental code. No code is deployed to production until the Project Team has concluded their ORR and received an Authority to Operate (ATO) from CMS.

As a general guideline, an optional initial Sprint (sometime referred as Sprint 0 – not depicted in Figure 12) can be used to handle environment placement or build out decided upon during the Release Planning Phase. Sprint 0 may be leveraged to build architecture component or inclusion of any reusable software component. Scrum Teams may perform this Sprint immediately after

the ISR so that the appropriate infrastructure contractor(s) have adequate time to stand up all necessary infrastructure and hardware.

As the Scrum Team works through the Sprints, the Scrum Master is responsible for removing impediments that may prohibit the Scrum Team from achieving the functional and non-functional product goals and deliverables. The Scrum Master is not a traditional team lead or project manager, but acts as a buffer between the Scrum Team and any distracting influences. The Scrum Master is the enforcer of the rules of Scrum, chairs key meetings, and challenges the Scrum Team to improve. The role has also been referred to as a servant-leader to reinforce these dual perspectives.

An important element of each Sprint is the **Daily Standup** which serves as the primary Scrum Team communication meeting. During the meeting, each Scrum Team member answers three questions: “*What work did you accomplish yesterday?*”, “*What work do you have planned for today?*”, and “*Are there any impediments preventing you from forward progress?*”. Any impediments identified during the Daily Scrum are documented by the Scrum Master and resolved outside of this meeting. The Daily Scrum starts precisely at the same time and in the same location every day, even if some Scrum Team members are absent. The meeting length is set (time boxed) to 15 minutes.

As design and development progress, a **Sprint Burndown Chart** can be used to depict remaining work within the Sprint and Backlog. The Sprint Burndown Chart is an information radiator that graphically depicts the progress of tasks and overall health of the Sprint. The Burndown Chart is updated daily and provides a simple view of Sprint progress.

#### **7.4.1.1 RR, SpD/ERR, SpR**

As defined by the XLC during this Phase, the IPT performs RR, SpD/ERR, and SpR during each Sprint.

##### **Requirements Review (RR)**

RR is performed to acknowledge an understanding of requirements, use cases, and acceptance criteria associated with selected tasks to be performed within that Sprint. Unlike Waterfall methodology, where release requirements are reviewed all at once in their entirety (sometimes taking days), here, requirements are reviewed incrementally focusing only on the work planned for that Sprint.

##### **Sprint Demonstrations and Environment Readiness Review (SpD/ERR)**

The SpD affords the customer an incremental visual review of the intended IT solution. SpD allows the customer to see what was accomplished during the Sprint. The SpD identifies work that was completed during each Sprint, as well as planned work that was not completed and must be carried forward into subsequent Sprints. The Product Owners is provided the opportunity to ensure the product is meeting expectations and can request the addition of user stories to the Sprint Backlog.

In conjunction with SpD, ERR allows the Product Owner to make a decision as to whether or not the product or solution is ready to move to the next environment. This incremental review of

verification and validation testing that may be occurring during each Sprint provides the customer with valuable feedback. Because of this similar nature of SpD and ERR, it is recommended that IPT combine the two reviews. This SpD/ERR combination allows participants or stakeholders to make a determination based on demonstration of the working software rather than solely based on paper results of testing.

### **Sprint Retrospective (SpR)**

SpR allows the Scrum Team to reflect on the past Sprint to determine where process improvements could be implemented. The SpR is analogous to a lessons learned session. Unlike Waterfall, where lessons learned happen only once at the end of a release, Scrum lessons learned are assessed after each Sprint. Feedback gets incorporated before the end of the release to improve the process and product by reducing the feedback cycle. The Scrum Master is typically tasked with the responsibility of instituting positive incremental changes. Generally, two improvement steps are selected, by the team, to be implemented within next Sprint.

The IPT schedules and performs RR, SpD/ERR, and SpR during each of the Sprints as they progress towards completion. The Scrum Team begins the next Sprint immediately following completion of the SpR.

In order to accomplish end to end integration testing, an additional Sprint (sometimes referred to as an Integration Sprint) can be added at the end of the Sprint cycle prior to implementation. This Sprint integrates and tests the output of each Sprint into a deployable production release.

#### **7.4.1.2 PDR and DDR**

During the Sprints Phase, and as approved within the PPA, the XLC requires that the IPT attend a Preliminary Design Review (PDR) and a Detailed Design Review (DDR). It is important to note that unlike RR, SpD/ERR and SpR, PDR and DDR are not performed for each Sprint. The PDR and DDR are performed only once per release. It is up to the IPT to determine the best time to schedule PDR and DDR with the TRB. The TRB has the discretion to require Project Teams to come back for additional review(s) if needed based on complexity and completeness of their project.

Generally, the PDR is scheduled early in the Sprint cycle. The PDR must be done early enough so that any design changes can be made with the least impact to the project. The DDR is scheduled as soon as the IT solution architecture and design are complete. The IPT drafts the applicable PDR and DDR XLC artifacts and presentation templates, and schedules each review once per release. Once all of the Sprints and XLC gate reviews have been successfully completed, a project can move forward to the Implementation Phase.

Lastly, it should be reiterated that this document does not serve as a training guide on software and architecture implementations. Project Teams should always exercise established industry best practices keeping in mind the importance of establishing architecture objectives, application overview, design coupling, and creating candidate architecture solutions with the Project Team.

## 7.4.2. Sprint Guideline Example

The Product Owner identified early on in the project life cycle that PRRS will likely serve 200 million Americans. Therefore, procurement and infrastructure build out activities occurred during the Release and Sprint Planning Phases prior to the start of development. Because this work was completed on time and accurately, the Scrum Team can begin to build on this foundation through Sprint iterations and code development.

After completing the Sprint Backlog within the Sprint Planning Phase, the PRRS Scrum Team gets to work on the first of six Sprints within Release 1.0. As the first Sprint begins, the PRRS IPT performs and completes RR to discuss and acknowledge the details and acceptance criteria for each of the tasks assigned to Sprint 1 (see Table 2 beginning on page 26).

The PRRS Scrum Team begins development of the intended solutions. Scrum Master Jason facilitates the first of many Daily Scrum meetings to review progress based on the Sprint Burndown Chart. While development is proceeding, members of the IPT are drafting the relevant artifacts required to attend the PDR with the TRB which has been scheduled at the conclusion of Sprint 1.

Scrum Master Jason and the Scrum Team continue to progress through Sprint 1, holding a Daily Scrum and maintaining the Burndown Chart. Jason schedules the SpD/ERR and invites the IPT. Via WebEx, the Scrum Team demonstrates the progress they have made towards completion of the Sprint 1 tasks. With the exception of a few minor comments, Product Owner Susan and the rest of the Scrum Team concur that the acceptance criteria has been satisfied for each of the Sprint 1 user stories. Thus, the Scrum Team will prepare to move forward with Sprint 2 development tasks. V&V testing results are then reviewed and reveal only minor issues that will easily be addressed prior the end of Sprint 1. Product Owner Susan agrees that the project appears on track, no scope changes are necessary, and concurs with the continuation of development.

As the Sprint moves towards completion, Scrum Master Jason schedules a SpR to provide the opportunity for the Scrum Team to offer suggestions for improvement and lessons learned. During the SpR, Product Owner Susan offers a process improvement suggestion that Sprint Demonstrations be held in person, in addition to WebEx. Susan believes that face to face communication is a more effective way to resolve any future issues should they arise.

At this point, preliminary design is achieved. The IPT submits the PPA approved XLC artifacts and PDR presentation slide deck to the TRB and attends the PDR gate review. Through the review, the TRB confirms that PRRS is in compliance with the CMS Technical Reference Architecture (TRA). In addition, the TRB concludes that there are no significant technical, security, or business risks affecting the continued detailed design and development of PRRS. The TRB recommends PRRS move forward to a DDR.

The PRRS Scrum Team continues to repeat Sprint cycles as identified above. The Daily Scrum, Burndown Chart updates, RR, SpD/ERR, SpR all continue throughout each cycle. As issues arise as a result of testing, scope changes, or other impediments, the Scrum Team works collectively,

and in an Agile manner, to ensure that Release 1.0 development and testing remain on track through timely issue mitigation.

As the detailed design and architecture are finalized, the IPT submits the PPA approved XLC artifacts and DDR presentation slide deck to the TRB and attends the DDR gate review. As in the PDR, the TRB confirms that PRRS is in compliance with the CMS TRA. In addition, the TRB concludes that there are no significant technical, security, or business risks affecting the progression towards testing, implementation, and operations and maintenance activities. The TRB recommends PRRS move forward to an ORR.

Finally, the PRRS Scrum Team arrives at Sprint 6 which will incorporate all of the output from Sprints 1 through 5 in order to perform integration testing. Integration testing reveals minor issues and the Scrum Team feels confident in moving forward into the last Phase, Implementation.

Once all of the Sprints and XLC gate reviews within this phase have been successfully completed, the IPT can begin to coordinate activities with the Infrastructure Contractor during the Implementation Phase.

## 7.5. Implementation Phase

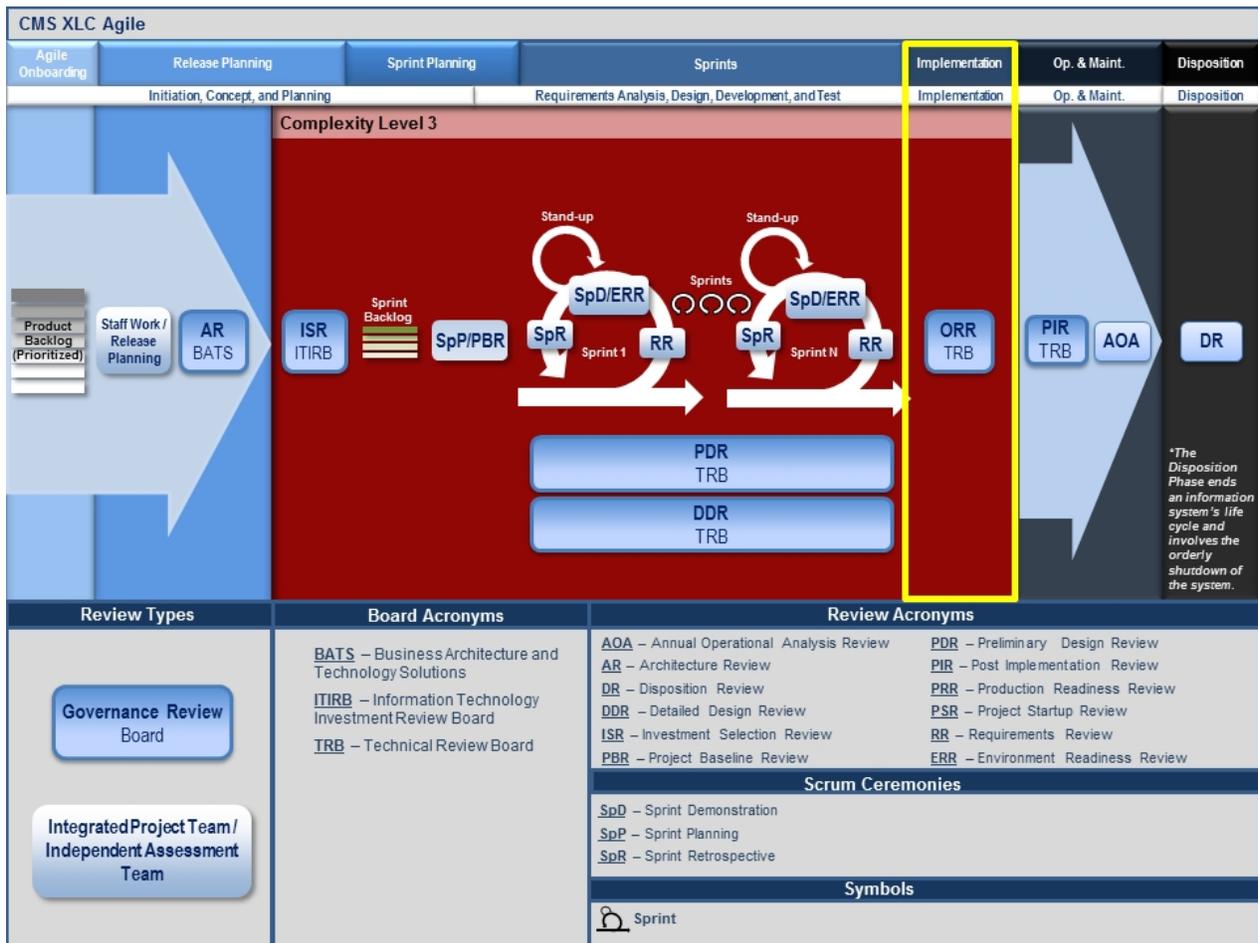


Figure 13: Implementation Phase

### 7.5.1. Implementation Phase Synopsis

The Implementation Phase contains the Operational Readiness Review (ORR) with the TRB. The purpose of this review is to ensure that the system/application completed its implementation processes according to plan and that it is ready for turnover to the Operations & Maintenance team and operational release into the Production environment.

At this point, the Scrum Team has successfully completed the Sprints Phase and concluded the final Integration Sprint. All development and test activity for this release should be completed and the IPT should be working with the Infrastructure contractor to ensure formal handoff for deployment. This phase does not differ in any significant way from the waterfall process.

Note: If a multi-release approach is chosen for an IT Project, the IPT would repeat the full lifecycle (Onboarding through Implementation) for each subsequent release. All IPTs are required to adhere to the contents of an approved PPA when creating and planning activities

surrounding artifacts, gate reviews and testing moving forward. In addition, subsequent releases may require a new ATO before implementation can occur into the production environment.

### 7.5.2. Implementation Guideline Example

Upon completion of the Sprints Phase, the PRRS Scrum Team has now finished all Sprints including full end-to-end testing, including security controls assessment (SCA), through the Integration Sprint. The IPT prepares final XLC artifacts as outlined in the approved PPA for the ORR with the TRB.

In order to attend ORR, the PRRS IPT must have a signed Authority to Operate (ATO) from the CMS Chief Information Security Officer (CISO) stating that all security requirements are met and risk is at an acceptable level. Without an ATO, PRRS will not be allowed to deploy into production.

During the ORR, the PRRS IPT presents the final artifacts to the TRB. The TRB reviews PRRS validation test results, and verifies approval of both the Operations and Maintenance manual and the signed ATO for PRRS Release 1.0.

Once the TRB is satisfied that the PRRS IPT is ready for deployment of Release 1.0, they issue formal approval to proceed to production. The PRRS IPT has now successfully completed the ORR gate review. At this point, the PRRS IPT works in conjunction with the Infrastructure Contractor(s) to ensure a smooth transition and promotion of PRRS Release 1.0 into the production environment for public consumption.

As Release 1.0 is deployed to production, the PRRS IPT begins the cycle anew for Release 1.1. Feedback received from Release 1.0 is collectively aggregated and assessed. A new PPA is drafted with the AR and ISR being waived. Product Owner Susan re-prioritizes the Product Backlog, Release Planning and Sprint Planning meetings are scheduled, and Release 1.1 gets underway.

## 7.6. Operations and Maintenance Phase

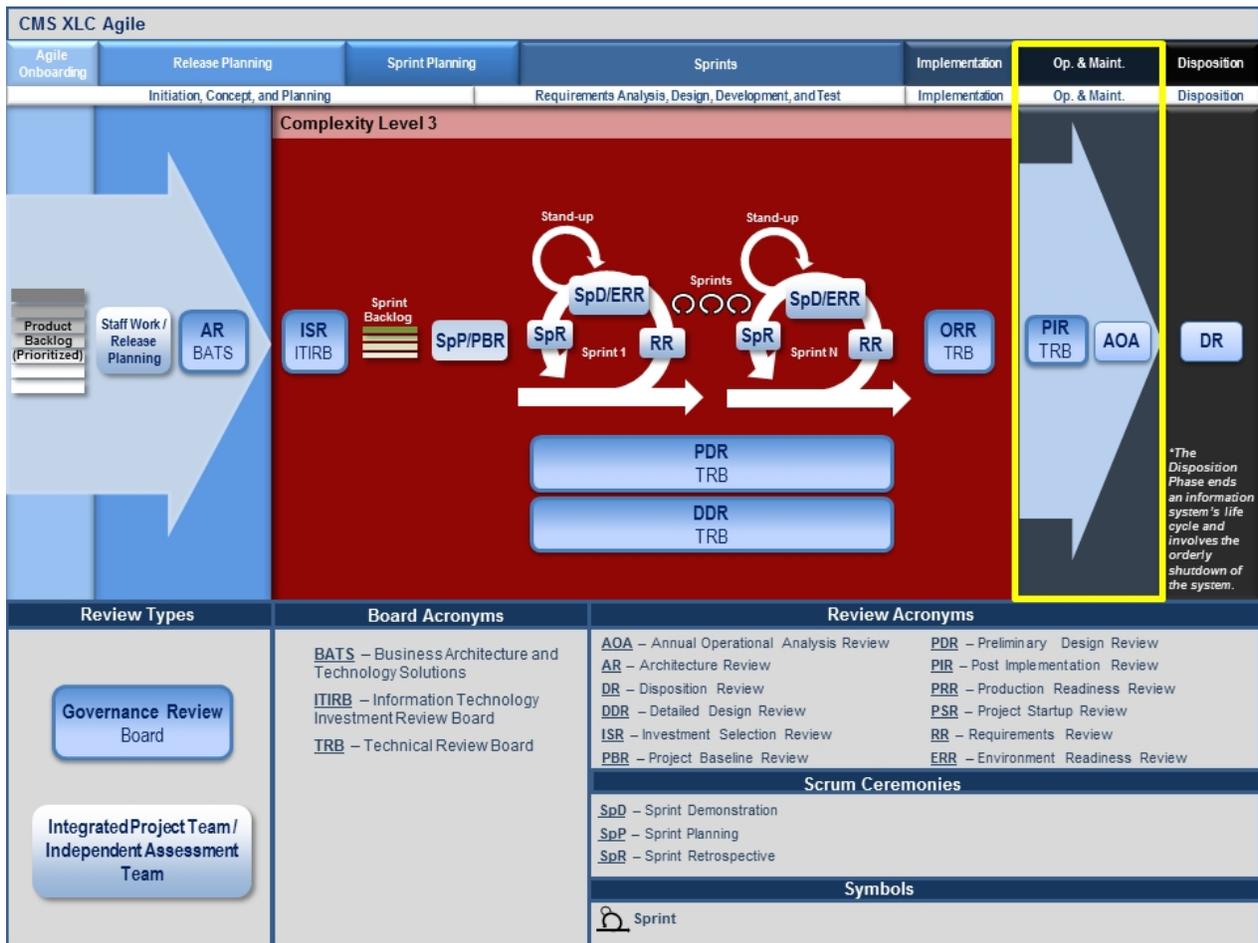


Figure 14: Operations and Maintenance Phase

### 7.6.1. Operations and Maintenance Phase Synopsis

Once a CMS IT solution is deployed to production, it enters the Operations and Maintenance (O&M) Phase. During this phase, the system is monitored and maintained for business need, performance, and security by the TRB, Infrastructure, and ADO organizations.

After a period of sustained operation, generally six to 12 months, a Post Implementation Review (PIR) is conducted by the TRB on new systems to determine the level of success of the IT project. The PIR focuses on lessons learned during the development and implementation of the solution, in addition to the level of satisfaction reported by stakeholders, customers, and users of the system.

An Annual Operational Assessment (AOA) is also performed in order to evaluate system performance, user satisfaction, adaptability to changing business needs, and new technologies that might improve the system. This review is diagnostic in nature and can lead to development or maintenance activities. Ultimately, AOA determines whether the IT Investment should

continue, be modified, or terminated. Often, the first PIR and AOA on a new system are combined.

In an Agile, multi-release approach, the IPT may propose to perform O&M phase reviews (PIR and AOA) with proper PPA justification only after all related releases (e.g. Release 1.0, Release 1.1) are in operation rather than for each release.

### 7.6.2. Operations and Maintenance Guideline Example

PRRS Release 1.0 was successfully deployed into production and the IPT began work on Release 1.1. After six months in production, the IPT schedules a combined PIR and AOA for review of Release 1.0 with the TRB. Artifacts as approved within the PPA are prepared for review along with the PIR presentation.

During the PIR/AOA, the IPT presents the results of PRRS Release 1.0 surveys and usage trends to the TRB. The IPT demonstrates how the project met the business need from the stakeholder's perspective in regard to performance expectations and actual outcomes. The TRB concludes that functionality and utilization of PRRS are consistent with the defined project goals and affirms continued production operation.

Consistent with the original release strategy, and post PIR/AOA review, PRRS Release 1.1 is deployed to production. After six months of production operation, the PRRS IPT will attend a PIR to review PRRS Release 1.1. AOA review for PRRS will be conducted on an as needed basis to determine whether the IT investment should continue, be modified or terminated.

## 7.7. Disposition Phase

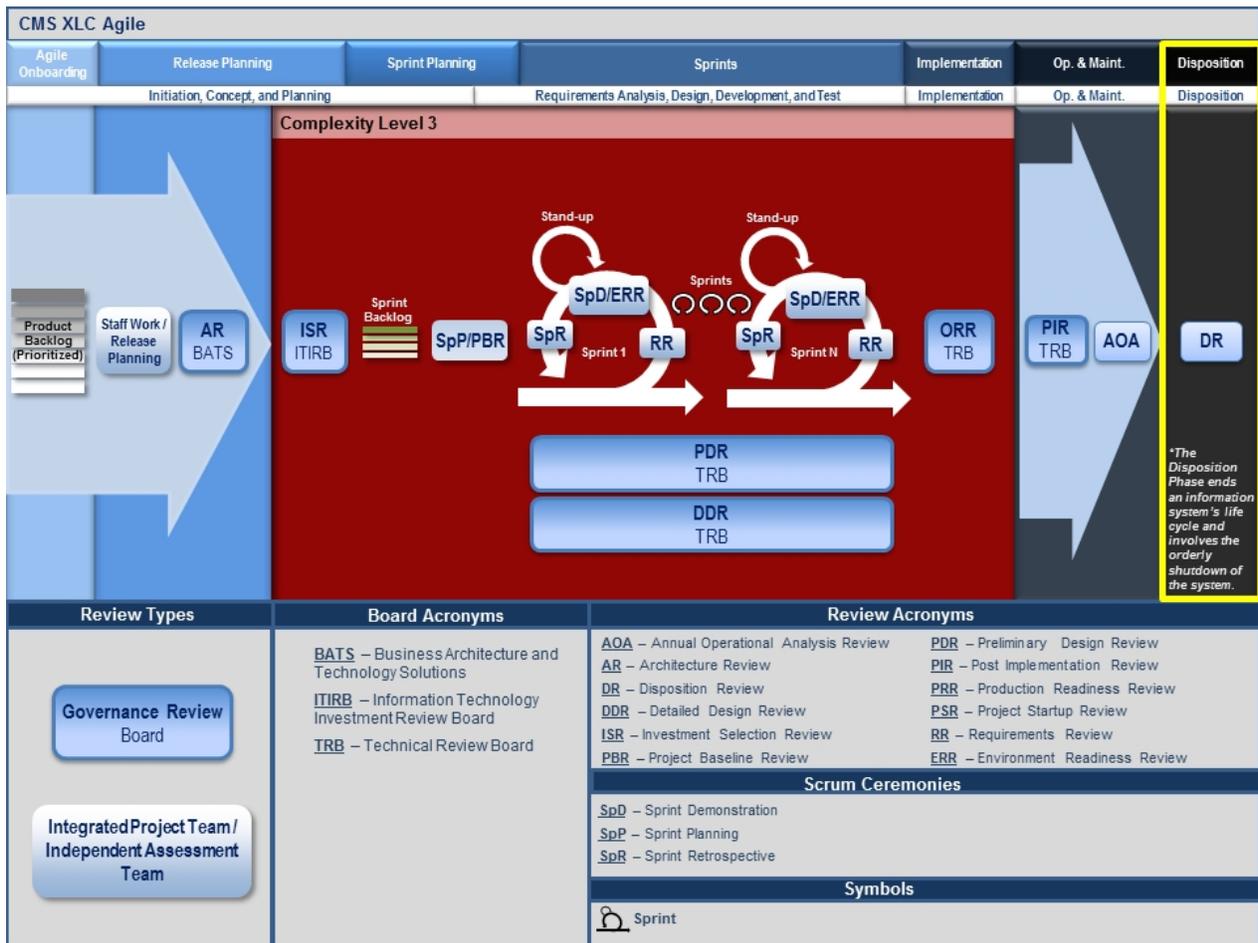


Figure 15: Disposition Phase

### 7.7.1. Disposition Phase Synopsis

The Disposition Phase of the XLC includes the decommissioning of IT solutions that no longer meet the business needs of CMS. If, during PIR/AOA review, it is determined that a system no longer satisfies the CMS business need, a Disposition Plan is presented at a Disposition Review (DR) and the system is subsequently retired in accordance with the approved plan. Disposition ensures that the IT investment has been completely and appropriately transitioned and disposed, thereby ending the life cycle of the IT project.

### 7.7.2. Disposition Guideline Example

After five years of continuous production operation, it is determined that PRRS is no longer aligned with CMS business needs. The IPT prepares the DR template for presentation of the disposition plan. The system is appropriately transitioned and decommissioned per standard operating procedures.

---

## 8. Information and Assistance

---

Contact the XLC Program Manager within the Division of IT Governance for further information regarding these guidelines ([XLC@cms.hhs.gov](mailto:XLC@cms.hhs.gov)).

## Appendix A: Acronyms

Table 3: Acronyms List

Acronym	Literal Translation
AR	Architecture Review
ADO	Application Development Organization
AOA	Annual Operational Assessment
ATO	Authority to Operate
BATS	Business Architecture and Technology Solutions
BDC	Baltimore Data Center
BO	Business Owner
CCSQ	Center for Clinical Standards and Quality
CISO	Chief Information Security Officer
CMS	Centers for Medicare & Medicaid Services
CSM	Certified Scrum Master
DDD	Detailed Description Document
DDR	Detailed Design Review
DR	Disposition Review
DSDM	Dynamic System Development Method
ERR	Environment Readiness Review
FDD	Feature Driven Development
GTL	Government Task Leader
IPT	Integrated Project Team
ISR	Investment Selection Review
IT	Information Technology
ITIRB	Information Technology Investment Review Board
O&M	Operations and Maintenance
OIS	Office of Information Services
ORR	Operational Readiness Review
PBR	Project Baseline Review
PDA	Personal Digital Assistant
PDR	Preliminary Design Review
PIR	Post Implementation Review
PPA	Project Process Agreement
PRRS	Provider Review and Rating System
PSI	Potentially Shippable Increment
RMH	Risk Management Handbook
ROI	Return on Investment
RR	Requirements Review
SCA	Security Controls Assessment
SDLC	Systems Development Life Cycle
SpD	Sprint Demonstration
SpP	Sprint Planning

<b>Acronym</b>	<b>Literal Translation</b>
SpR	Sprint Retrospective
TBD	To Be Determined
TRA	Technical Reference Architecture
TRB	Technical Review Board
UAT	User Acceptance Testing
URL	Universal Resource Locator
V&V	Verification and Validation
XLC	eXpedited Life Cycle
XP	Extreme Programming

## Appendix B: CMS XLC Agile

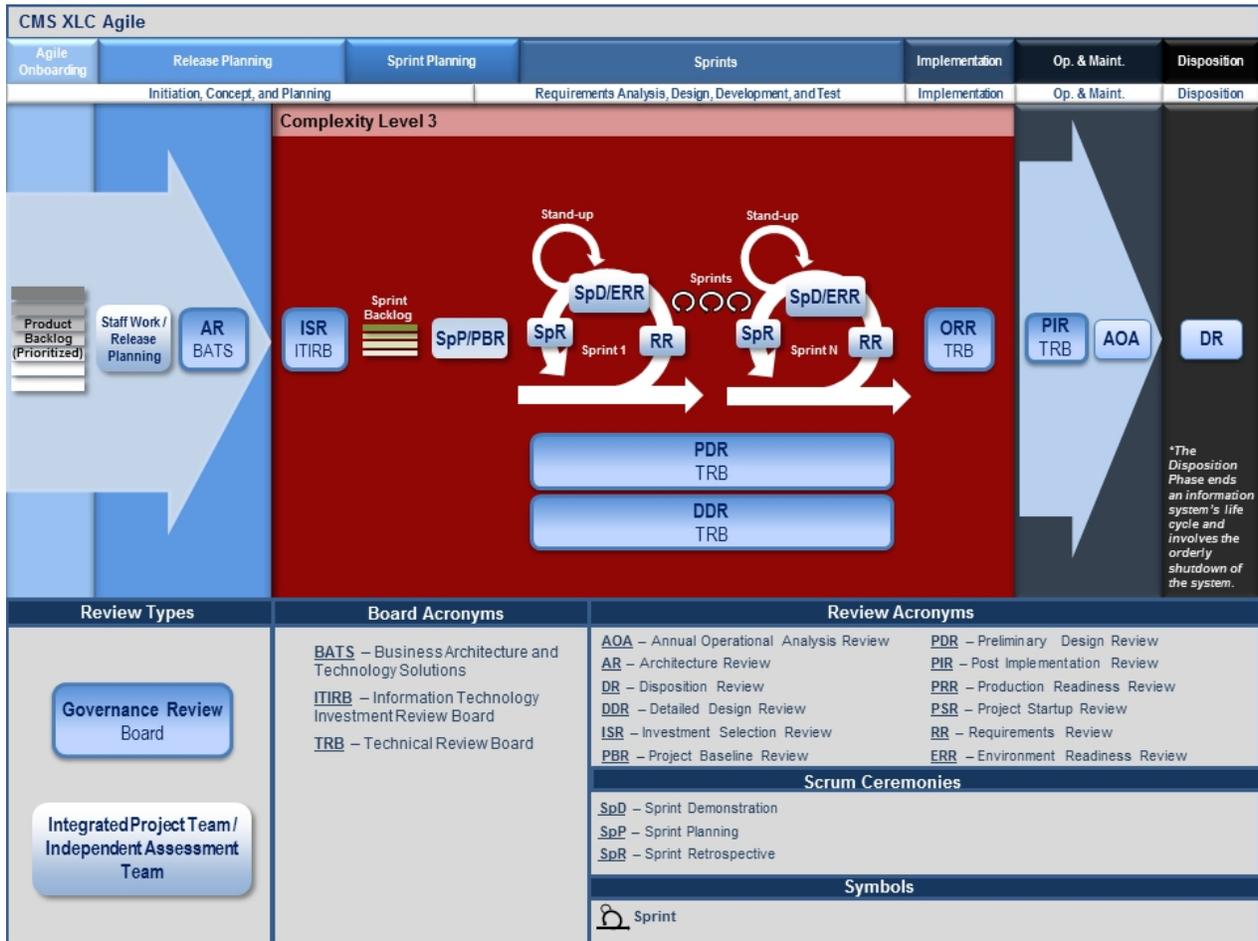


Figure 16: CMS XLC Agile - Complexity Level 3

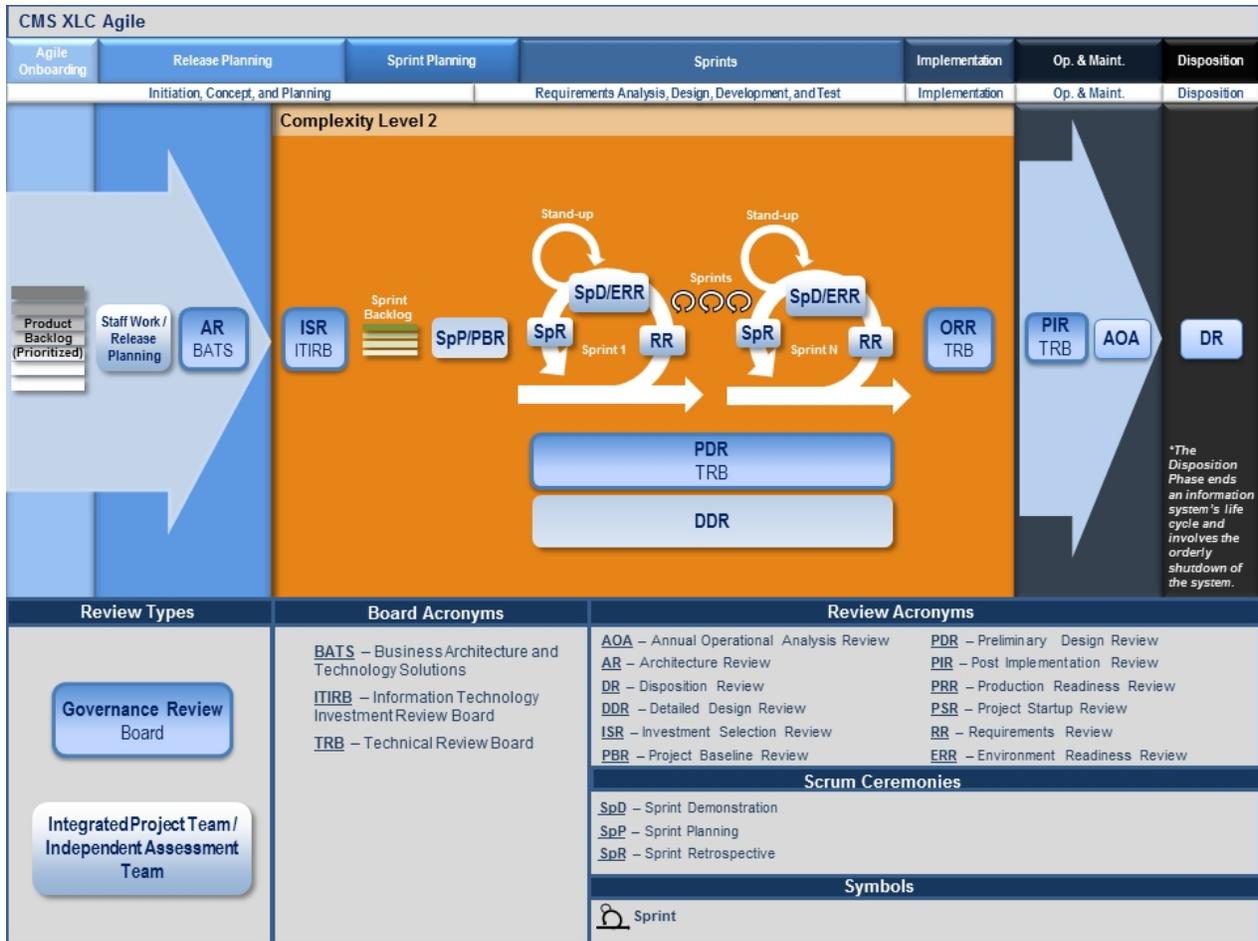


Figure 17: CMS XLC Agile - Complexity Level 2

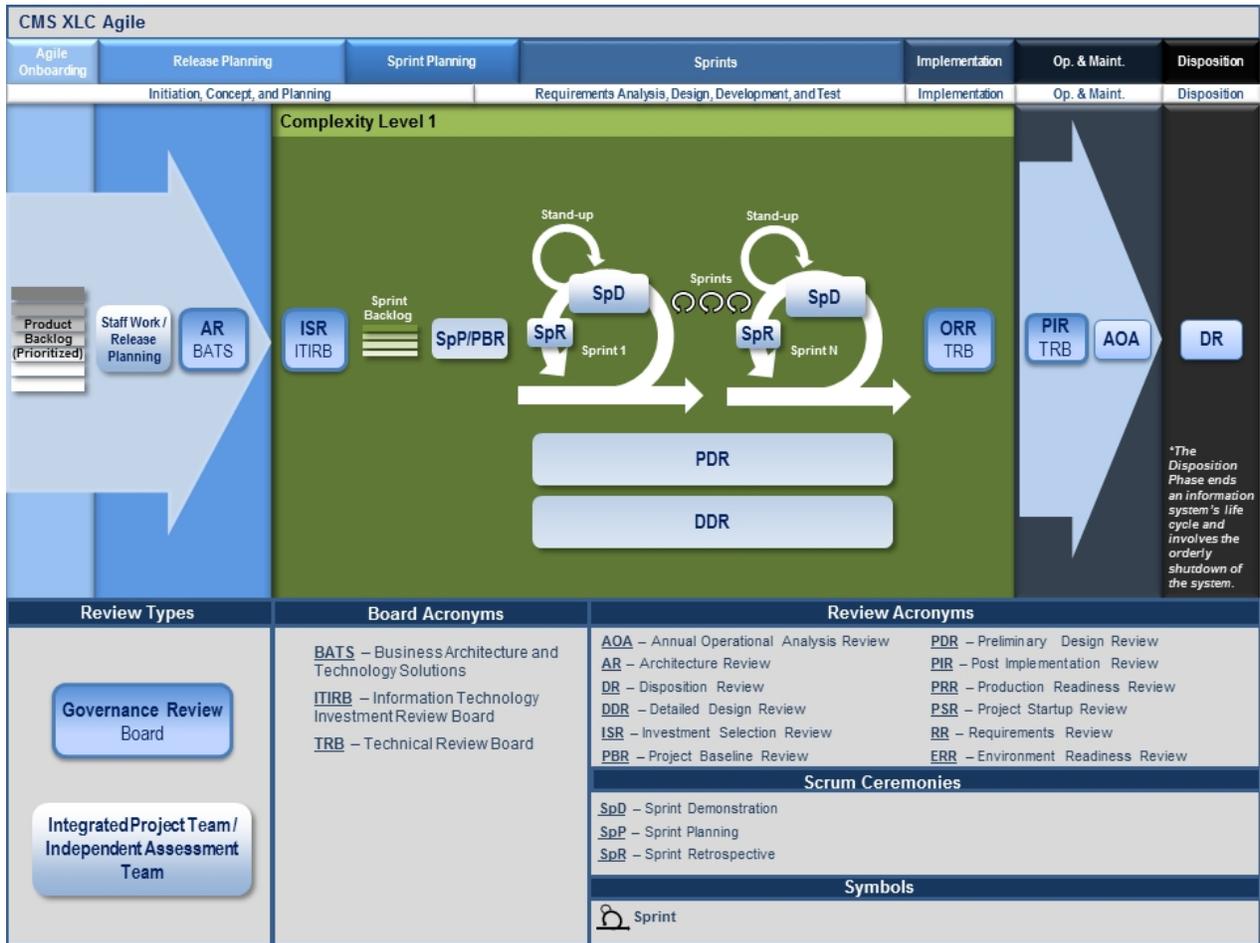


Figure 18: CMS XLC Agile - Complexity Level 1

## Appendix C: Glossary

Table 4: Glossary

Term	Definition
<b>Acceptance Criteria</b>	Shared understanding of what it means for work to be complete, to ensure transparency.
<b>Agile</b>	Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
<b>Burndown Chart</b>	Burndown charts show work remaining over time. Work remaining is the Y axis and time is the X axis. The work remaining may fluctuate but should eventually trend downward.
<b>Daily Scrum</b>	<p>A fifteen-minute daily meeting for each team member to answer three questions:</p> <ul style="list-style-type: none"> <li>- "What have I done since the last Scrum meeting? (i.e. yesterday)"</li> <li>- "What will I do before the next Scrum meeting? (i.e. today)"</li> <li>- "What prevents me from performing my work as efficiently as possible?"</li> </ul>
<b>Estimation</b>	<p>Relative estimation is one of the several distinct flavors of estimation used in Agile teams, and consists of estimating tasks or user stories, not separately and in absolute units of time, but by comparison or by grouping of items of equivalent difficulty.</p> <p>Estimates are a team's best guess at a point in time and should improve as historical data becomes available.</p>

Term	Definition
<b>Impediments</b>	Anything that prevents a team member from performing work as efficiently as possible is an impediment. Each team member has an opportunity to announce impediments during the daily Scrum meeting. The Scrum Master is charged with ensuring impediments get resolved. Scrum Masters often arrange sidebar meetings when impediments cannot be resolved on the spot in the daily Scrum meeting.
<b>Planning Poker</b>	A consensus based Agile estimation and planning technique used to score User Stories.  Other scoring methods include: Fibonacci, Fists of Five, etc.
<b>Product Backlog</b>	The product backlog (or "backlog") is the requirements for a system, expressed as a prioritized list of product backlog Items. These included both functional and non-functional customer requirements, as well as technical team-generated requirements. While there are multiple inputs to the product backlog, it is the sole responsibility of the product owner to prioritize the product backlog.  During a Sprint planning meeting, backlog items are moved from the product backlog into a sprint, based on the product owner's priorities.
<b>Product Owner</b>	The Product Owner is the person responsible for maximizing the value of the product, the work of the Development Team, and management of the Product Backlog.
<b>Scrum</b>	Scrum is a framework structured to support complex product development. Scrum consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.

Term	Definition
<b>Scrum Master</b>	<p>The Scrum Master is a facilitator for the team and Product Owner. Rather than manage the team, the Scrum Master works to assist both the team and Product Owner in the following ways:</p> <ul style="list-style-type: none"> <li>• Remove the barriers between the development and the Product Owner so that the Product Owner directly drives development.</li> <li>• Teach the Product Owner how to maximize ROI and meet his/her objectives through Scrum.</li> <li>• Improve the lives of the development team by facilitating creativity and empowerment.</li> <li>• Improve the productivity of the development team in any way possible.</li> <li>• Improve the engineering practices and tools so that each increment of functionality is potentially shippable.</li> <li>• Keep information about the team's progress up to date and visible to all parties.</li> </ul>
<b>Scrum Team</b>	<p>The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team.</p>
<b>Servant-leader</b>	<p>Servant leadership is both a leadership philosophy and set of leadership practices. Traditional leadership generally involves the accumulation and exercise of power by one at the “top of the pyramid.” By comparison, the servant-leader shares power, puts the needs of others first and helps people develop and perform as highly as possible.</p>

Term	Definition
<b>Sprint</b>	<p>An iteration of work during which an increment of product functionality is implemented.</p> <p>The sprint starts with a one-day sprint planning meeting. Many daily Scrum meetings occur during the sprint (one per day). A sprint review meeting is held at the end of each sprint, followed by a sprint retrospective meeting.</p> <p>During the sprint, the team must not be interrupted with additional requests. Guaranteeing the team won't be interrupted allows it to make real commitments it can be expected to keep.</p> <p>Out of practical necessity, some teams choose to bend this rule by declaring some team members 80 percent available at the outset so they still have some cycles left for "Priority One" bugs and emergencies.</p>
<b>Sprint Backlog</b>	<p>Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.</p>
<b>Sprint Planning Meeting</b>	<p>The sprint planning meeting is a negotiation between the team and the Product Owner about what the team will do during the next sprint.</p> <p>The Product Owner and all team members agree on a set of sprint goals, which is used to determine which product backlog items to commit from the uncommitted backlog to the sprint. Often new backlog items are defined during the meeting. This portion of the sprint planning meeting is time-boxed.</p>
<b>Story Point</b>	<p>A tool for estimating software development projects. Story points are not equal to task hours.</p>

Term	Definition
<b>User Story</b>	<p>In software development and product management, a user story is one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function.</p> <p>User stories are used with Agile software development methodologies as the basis for defining the functions a business system must provide, and to facilitate requirements management. It captures the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail by what can be hand-written on a small paper notecard.</p>
<b>Velocity</b>	<p>In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.</p> <p>Once established, velocity can be used to plan projects and forecast release and product completion dates.</p>

## Appendix D: Quick Reference

This appendix serves as a quick reference to detailed information regarding the CMS XLC.

Document or Website Name	Description	URL
XLC Website	Public website for CMS XLC related artifacts and templates.	<a href="http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/index.html">http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/index.html</a>
XLC Detailed Description Document	Provides detailed information regarding all aspects of the XLC process.	<a href="http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/XLC-DDD.pdf">http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/XLC-DDD.pdf</a>
CMS Expedited Life Cycle e-Learning Course	Provides participants with foundational information on CMS' IT project management framework called the eXpedited Life Cycle (XLC).	<a href="https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Courses/XLC.html">https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Courses/XLC.html</a>
CMS Project Process Agreement (PPA) e-Learning Course	Provides participants with foundational information about tailoring the XLC framework to accommodate the unique characteristics of each IT project.	<a href="http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Courses/PPA.html">http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Courses/PPA.html</a>

## Appendix E: References

1. Schwaber, Ken. (2004). *Agile Project Management with Scrum*. Redmond, Washington: Microsoft Press.
2. Szalvay, Victor. (2007). *Glossary of Scrum Terms*. Retrieved from <http://www.scrumalliance.org/community/articles/2007/march/glossary-of-scrum-terms>
3. *CMS Expedited Life Cycle Process: Detailed Description*, Version 2.11, Centers for Medicare & Medicaid Services (CMS), November 13, 2012. Retrieved from: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/XLC-DDD.pdf>
4. *XLC FAQ*. Centers for Medicare & Medicaid Services (CMS) December 24, 2013. Retrieved from: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/FAQ.html>
5. Agile Software Development. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development)
6. Extreme Programming. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/Extreme\\_programming](http://en.wikipedia.org/wiki/Extreme_programming)
7. Feature Drive Development. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/Feature-driven\\_development](http://en.wikipedia.org/wiki/Feature-driven_development)
8. Return on Investment. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/Return\\_on\\_investment](http://en.wikipedia.org/wiki/Return_on_investment)
9. Servant Leadership. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/Servant\\_leadership](http://en.wikipedia.org/wiki/Servant_leadership)
10. User Stories. Retrieved March 14, 2014 from Wikipedia: [http://en.wikipedia.org/wiki/User\\_story](http://en.wikipedia.org/wiki/User_story)
11. Glossary of Scrum Terms. Retrieved March 14, 2014 from Scrum.org: <http://www.scrum.org/Resources/Scrum-Glossary>