State of California

Understanding

Agile

California
Department of Technology
California Project Management Office

Document Control

Version History

Document history and version control is used to record detail of minor and major changes to Understanding Agile.

Version #	Date	Author	Changes
1.0	12.05.2016	CDT	Initial Publication

Table of Contents

1 Introduction 4

2 Agile Defined 5

3 Agile Myths 11

4 Waterfall and Agile 18

5 Selecting the Right Approach 24

6 Transitioning to Agile 27

7 Next Steps 30

1. Introduction

An organization's choice of a project management and development approach for their project needs depend on several factors, including the relative experience of the project team and key characteristics of the project, such as complexity and size. Some approaches place more emphasis on fully planning and predicting the outcomes of project activities as they unfold, to minimize the amount of change that occurs. Other approaches instead accept change as something that is inevitable and place importance on the need to react and adapt. The prevailing predictive and adaptive development approaches are waterfall and agile, respectively. Each presents unique opportunities and challenges for the organization undertaking the project.

Consider project characteristics and team experience when choosing a project management and development approach.

Agile development practices present an alternative approach to the waterfall method more commonly employed in the State of California for information technology (IT) projects. As the landscape evolves and there is a growing awareness and desire to use agile, state organizations need to be equipped with basic information to be able to make informed decisions about the path forward for their project needs. Agile development may present some benefits inherent in its adaptive and reactive methodology, but it is not one-size-fits-all nor suggested to be a more successful approach compared to waterfall. Each organization must understand the practices and principles of agile and determine what is appropriate for their project.

The California Department of Technology (CDT) is developing agile guidance that will be released in three parts. The first part, this "Understanding Agile" document, explores key agile concepts within the context of project management and system development; compares and contrasts waterfall and agile; and discusses next steps for organizations that want to further explore using agile. Part two will help organizations get ready for agile by discussing the core attributes to planning for an agile project and how to assess whether the organization is environmentally and culturally ready take on the effort. Lastly, part three will provide practical guidance and tools – a "how to" guide – for organizations that identify agile as the best approach to implement their project. Each part will build upon one another and, collectively, will enable organizations to focus on the delivery of business value and outcomes to its customers in a continuous and incremental way that is suitable for the government setting.

2. Agile Defined

Learn about agile principles and values.

Agile is an approach in which the solution for a business problem or opportunity can evolve over the life of the project. It promotes:

- Frequent delivery of usable system functionality
- Iterative planning as user needs are more defined
- Delivery of high-value functionality early in the project
- Team is empowered to make decisions and accountable for taking ownership
- Continuous improvement by incorporating feedback and lessons learned on an ongoing basis

These characteristics enable rapid and flexible response to change and frequent Stakeholder exposure. Consequently, agile lends itself most appropriately to projects in which accurate estimates, stable plans, and reliable predictions are not available in the early project stages.

While some may think agile can solve the issues associated with traditional project management and system development that many organizations face, use of agile should be thoughtfully considered for what it might realistically help achieve based on the organization's ability and capacity to be adaptive. Agile is a way of thinking and doing with several core principles that should be followed. These include the following six concepts detailed below:

- Iterative development
- User centered design*
- Iterative planning
- Flexible scope management
- Team ownership
- Continuous improvement of processes

*User centered design is not formally one of the principles of agile. However, it is a positive, well-regarded complement to agile processes and is a key tenet for State of California's implementation of agile.

Iterative Development

As depicted in Figure 2-1, agile includes short development lifecycles, also known as iterations, that are repeated over the course of the project. The duration of each iteration may be anywhere from 1-4 weeks as appropriate for the specific project, and focuses on just a portion of the project's functional scope. Full functionality is built-up cumulatively through a number of iterations, each of which produces a functioning increment of the overall solution. Because working system functionality is produced at the end of every iteration, users have early and frequent exposure to the developing system which provides ongoing opportunities to "course correct" if the system is not meeting expectations.

In Section 4, see Product Development Cycle for an expanded view of how products are delivered for each approach.

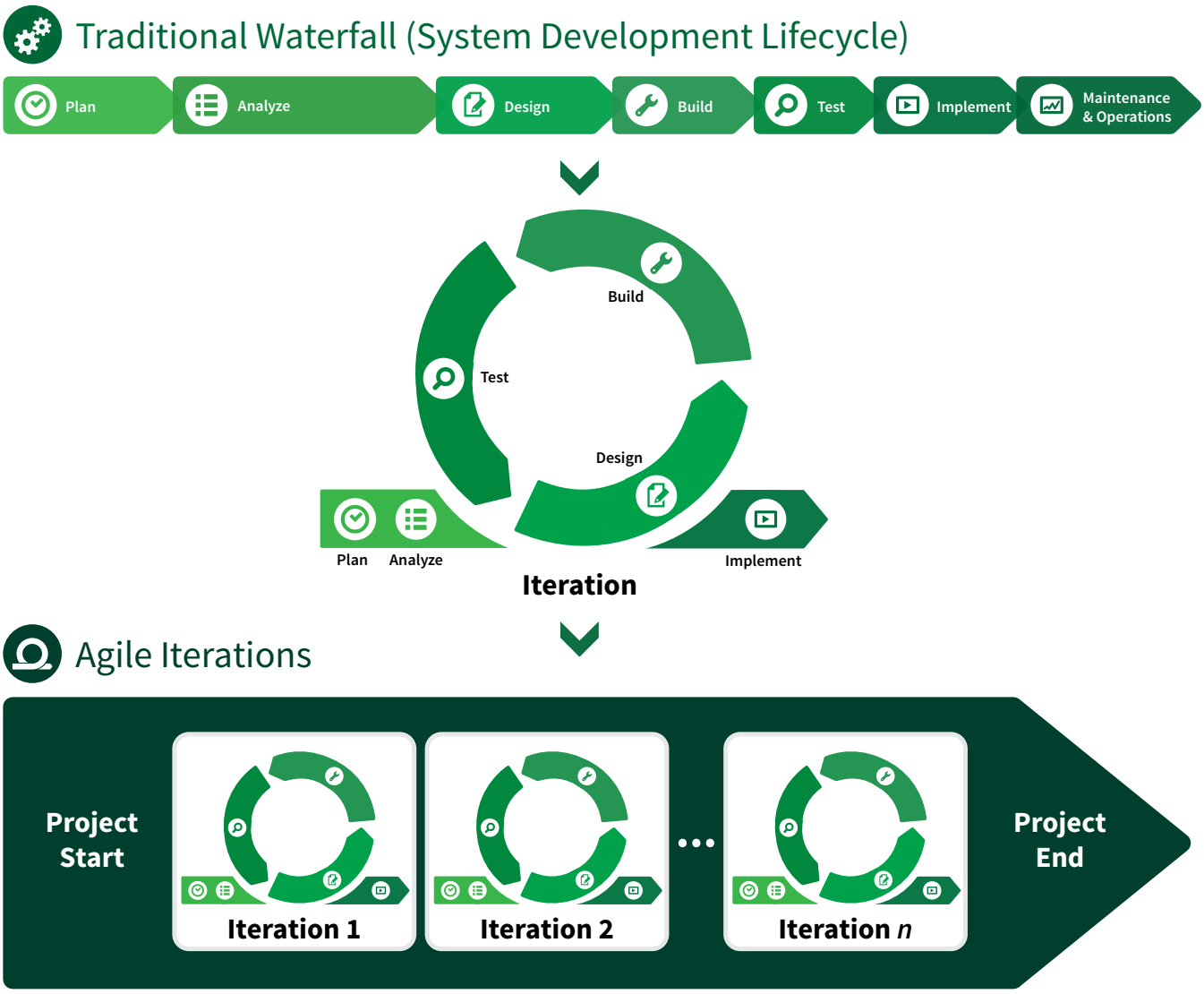

Figure 2-1 contrasts the traditional waterfall lifecycle with an iterative agile lifecycle.

User-Centered Design

User-centered design is a frame of thought where user needs are fundamental to every aspect of the project, including conceptualization, planning, and design. Priority is placed on delivering value to users via a solution that meets or exceeds expectations. Instead of developing a solution based on a pre-determined set of requirements, success is defined by creating a system that enables users to complete their business in a way that brings about efficiencies and makes sense from the users' perspective. This is done through continuous involvement of the business owner to solicit feedback and create solutions that best reflect the users' needs. This way of thinking is important to both project management and system development, whether agile or traditional waterfall is followed.

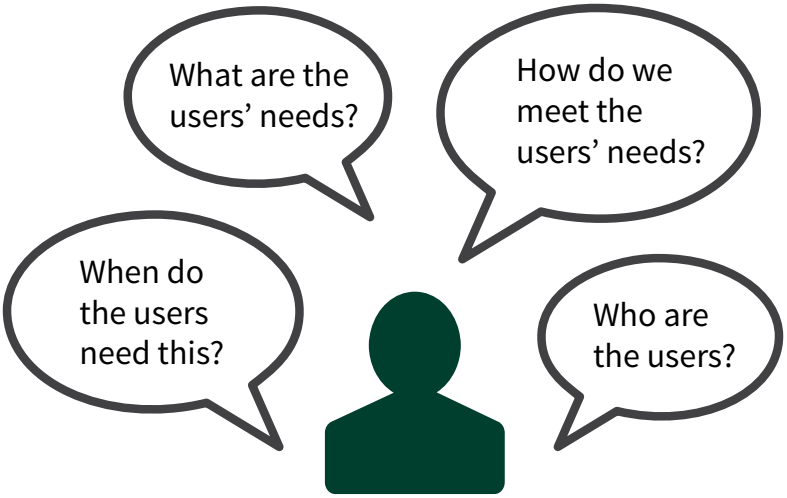

Figure 2-2 illustrates the frame of thought for a user centered design approach.

The following are a few benefits of a user-centered design approach:

Solidifying the Business Case

When identifying business problems or opportunities to conceptualize a project, understanding user needs confirm that you have a solid business case. By understanding who the users are, what they need, how their needs could be met, and when the users need a solution (Figure 2-2), a project is justified in simple terms that can be easily aligned to organizational mission and goals.

Prioritizing the Scope

Having a clear picture of when the solution is needed by users allows a business owner to prioritize scope items and determine the best release strategy for their work. By developing and delivering scope items that meet the users' highest priorities first, the risk of not delivering core functionality goes down significantly, and user satisfaction and probability of project success go up.

Discovering and Validating User Needs

Frequent user feedback through the business owner and the demonstration and usage of functioning product increments help drive development of quality solutions. Feedback is solicited throughout the project lifecycle to discover and ensure user needs are met. Through early and frequent presentation of functionality to the user at the end of each iteration, and actual usage of developed functionality by users, agile teams gain immediate feedback to determine whether they are on track to achieving the product vision. They also use this feedback loop to refine the vision and overall strategy, and adjust future work plans accordingly.

Ensuring user needs are met will increase the probability of project success.

Iterative Planning

While agile training and education typically focuses on the activities associated with system development, knowing how to manage an agile project is equally important. Within agile, planning must occur at multiple levels, as shown in Figure 2-3. Agile planning should address day-to-day activities, iterations, releases, the entire project, the organization's project portfolio, and the broader organizational strategy.

Figure 2-3 shows the many layers of planning for agile.

Like any project, those incorporating an agile approach progress through the project management lifecycle and are initiated with the drafting of a Project Charter. The Project Charter for an agile project should contain a concise and specific vision statement that outlines the purpose of the project and what it hopes to achieve. A roadmap describes the prioritized list of functionality which, when developed, will move the project towards accomplishing the vision described in the charter. The roadmap also outlines the project scope elements and the description of each functional element is elaborated upon as learning occurs within each successive iteration. Progressive elaboration is highly encouraged based on the belief that project teams will know the least about what the user wants and how to achieve it at the beginning of a project. The agile approach is to refrain from elaborating on the vision in detail until the project team is ready to collaborate with the user to develop the functionality.

In Section 4, see Project Management Planning to understand adaptive planning through progressive elaboration.

Flexible Scope Management

Agile teams progressively learn more during each iteration to better understand the scope of work and gain more clarity regarding what the users need. The project team is focused on delivering the greatest business value to users first as opposed to delivering all possible functionality at once. This approach requires a flexible project scope when other factors such as time and budget are constrained. An agile project's end is not based on when the project scope is completed, but rather when the planned time has elapsed and the benefit of releasing resources to begin another development effort outweighs the value of continuing to develop the remaining functionality. Many projects in state organizations may not be afforded autonomy to manage the project scope in this manner; in those cases, it is vital for the team to understand and differentiate the "must haves" from the "nice to haves" early in the planning effort. This facilitates delivery of must-haves early in the development cycle to provide users the greatest benefits as soon as possible.

In Section 4, see Project Management Planning to provide additional information on the waterfall planning.

Team Ownership

Agile teams consist of members who represent the users as well as those who possess the technical skills necessary to develop the solution. The team works together to plan, analyze, design, develop, test and integrate functionality within short development cycles. Agile teams decide how the work will be accomplished in order to best satisfy user needs. In this way, they assume accountability and ownership for delivering a quality solution.

In order for team ownership to take root, project teams must be empowered by management to operate independently. Management trust and support of change is equally important to building an effective team and the overall success of an agile project.

Continuous Improvement of Processes

At the end of each iteration, agile teams receive immediate feedback on their work from Stakeholders and users through a product review meeting. In addition, agile teams typically inspect their processes, tools, communication, environment, and other project management related issues every iteration. The team makes adjustments immediately which are leveraged in the next iteration. This continuous improvement increases the team's ability to produce desired outputs. As soon as one iteration ends, a new one begins.

3. Agile Myths

Common agile misconceptions explored.

To better understand what agile is, it may be helpful to understand what it is not. Agile is not a new approach, but it may be new to many project team members within state organizations. As with many concepts and approaches, misconceptions and myths are often times communicated and shared amongst new and even seasoned practitioners. Myths can be misleading when trying to understand what work might be involved and what situations are most appropriate for leveraging an agile approach. The following agile myths and explanations can help ensure a common, basic understanding when considering agile, and thus help level expectations for a project team or organization considering applying agile concepts.

Myth 1 – Agile Means "No Planning"

As with any approach, planning is a vital aspect that, if not adequately carried out, greatly diminishes the effectiveness of a successful implementation. However, as opposed to conducting extensive planning upfront, agile spreads this planning activity (e.g., what specific functionality will be delivered when) more evenly throughout the project lifecycle. High-level planning is completed at the beginning of an agile project and is continuously elaborated upon throughout the project as new information becomes available. This continuous planning allows a project to start much quicker and to be more nimble to make ongoing adjustments in strategy as new information becomes available. This can come through changes in business needs or priorities; project issues, risks, or resources; and even changes in available technology. It also provides the project team with the ability to more easily and efficiently adapt to changes and optimize plans as new information emerges.

In Section 4, see Project Management Planning for an example of iteration planning.

Myth 2 – Agile Means "No Governance"

Within an agile approach, the team members working on the project have autonomy over decisions about how to meet the needs of the user. However, most state organizations will find it difficult to allow project teams complete autonomy due to reporting and/or other governance requirements. As a result, organizations transitioning to agile may need to modify their governance practices. This includes incorporating clearly defined parameters within which the project team is free to make decisions and a clearly defined, fast-moving governance process to make decisions that are outside the team's purview.

To create an environment that supports team autonomy, the organization should establish a governance process that meets regularly; can accommodate ad hoc meetings; make decisions quickly; and is comprised of members with appropriate knowledge of the project, business, and users. Defining lightweight,

fast moving, and effective project governance is incredibly important for agile project success. The key is to establish a process that is appropriately specific, but not overly prescriptive.

Myth 3 – There is No Documentation with Agile

The adaptive and iterative nature of agile places less emphasis on the need for documentation compared to waterfall, but that does not mean that no documentation is required. Elements of the project continuously evolve as additional information becomes available and user needs are defined.

As shown in Figure 3-1, a traditional approach results in detailed documentation at the end of each phase. Following an agile approach does mean less documentation compared to a traditional waterfall approach, but documentation is needed nonetheless. In agile, an appropriate level of documentation will be an output of each iteration.

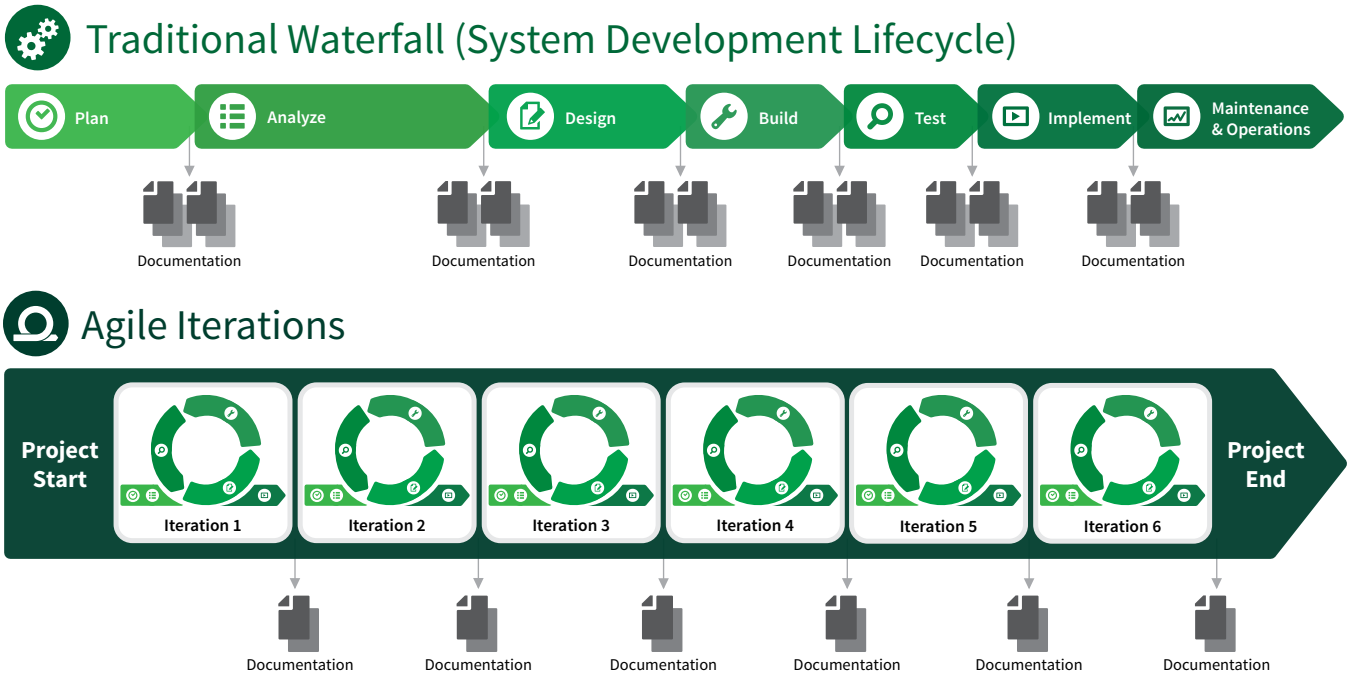

Figure 3-1 contrasts the development of documentation in a traditional waterfall approach versus an adaptive agile approach.

Developing adequately detailed documentation for agile is a necessity to:

- Meet the needs of project Stakeholders.
- Document decisions made.
- Support communication with external groups - including Stakeholders outside the project team or for team members that cannot collocate.
- Support the use, operation, and maintenance of the system.
- Capture lessons learned for continuous improvement and to benefit future projects.
- Report project status and performance metrics.

Whether your approach is agile or waterfall, the documentation that you develop should serve a purpose and not be created only because it was required in efforts undertaken in the past. The effective management of a project should have value-driven documentation that supports the project team's communication with Stakeholders, and enables the business to use the product effectively, and the technical team to support and maintain it. When considering what documentation looks like in your project, think about the value of the document or if it is needed, what information needs to be captured, when it needs to be captured, with whom it needs to be shared, and how that documentation might help the team improve.

Leverage templates from the California Project Management Framework (CA-PMF) as a starting point for agile documentation: capmf.cio.ca.gov/

Myth 4 – Agile Practices are New

The practices of agile have been around for the greater part of the last century. In the 1930's, physicist Walter Shewhart began improving products and processes through iterative cycles. This practice was later modified by W. Edwards Deming to become the Plan-Do-Study-Act (PDSA), also known as Plan-Do-Check-Act (PDCA), cycle for continuous improvement and quality management. Up through the 1980's, the United States military, NASA, IBM, Honda, Toyota, Canon, and others continued to experiment with and evolve concepts and practices we recognize as agile. These ideas led to the publication of the Agile Manifesto in 2001 and identification of the common values and principles for improving the approach to system development projects. Currently, several varieties of agile-based methodologies are used in these efforts, including Scrum, Extreme Programming (XP), and in some cases, Kanban.

For more information on the Manifesto for Agile Software Development and its 12 principles, visit:

agilemanifesto.org/

Myth 5 – Agile Only Works with Small Projects

An agile development team consists of small, cross-functional groups that collaborate throughout the development process. This approach can be equally effective on small projects and larger efforts to develop complex systems since agile teams typically "divide and conquer." For larger projects, this means that multiple teams can be organized and focus on separate components of system functionality and/or technical architecture.

For agile projects of all sizes, but especially for the large and complex, continuous integration of developed components on a daily if not more frequent basis is a critical success factor– more specifically, project teams need to check in and test newly developed code against the larger solution within a production-like environment. In an agile project with typically short development iterations, parallel development efforts, and frequent delivery of functionality, project teams must integrate their work often to detect and resolve errors as quickly as possible, with the ultimate goal of being able to deploy at any time. If project teams delay the integration to just-prior-to-release, they will likely run out of time to adequately perform testing, address defects, and prepare the infrastructure. Agile teams should ensure that they have the right automated build and test tools, and the appropriate processes in place to support continuous integration.

Myth 6 – Agile = Scrum

Scrum is a popular development methodology that is iterative and adaptive; however, Scrum and agile are not the same thing. Scrum is a framework for developing and managing work, while agile is an approach that follows a common set of values and principles that many methodologies fall under. Agile projects do not have to adopt any particular development methodology. Organizations must assess each development methodology to identify which is best suited for the environment. It is important to understand that the different development methodologies all focus on understanding and meeting the users' needs in a flexible and iterative way.

Furthermore, for organizations that are not ready to adopt agile as a development methodology, some agile practices can and should be leveraged to complement a waterfall approach. This includes those that pertain to the culture and environment of an organization (e.g., collocating teams, having access to business owners), or to project planning (e.g., deploying the project over several releases instead of one release at the end). Agile development methodologies have a greater chance of successfully achieving the desired outcomes when adopted in its entirety; this incremental adoption of agile organizational and planning practices can help lay the foundation for a later adoption of an agile development methodology.

In Section 6, see Transitioning to Agile for additional suggestions of agile characteristics for incremental adoption.

As shown in Figure 3-2, Scrum is just one of many methodologies based on agile values and principles. Other methodologies include Scaled Agile, Extreme Programming, and Kanban.

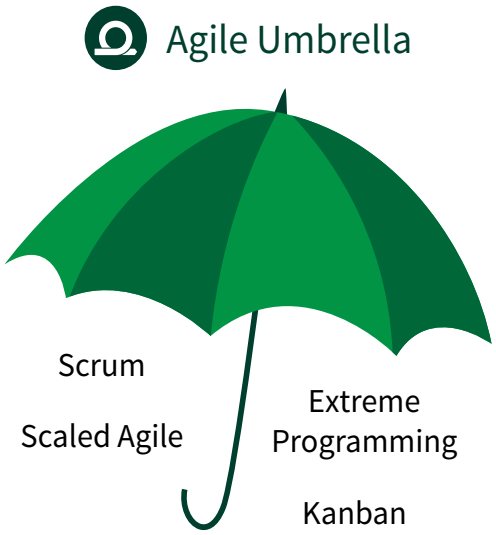

Figure 3-2 depicts the different development methodologies that are collected under the umbrella of agile.

Myth 7 – Implementing Agile is Easy

Change is hard. Transitioning an organization that is more accustomed to a traditional waterfall approach to an agile approach is not an exception to this rule. A significant number of state organizations will not have practices and procedures that are geared towards those of an adaptive approach and will likely need to focus on adapting the project team's project management and system development processes to the unique characteristics of the organization, project, and people.

In Section 6, see Transitioning to Agile for additional guidance on implementing agile.

To achieve the full benefits, project teams must not only learn the best practices of agile; it is also important to understand the specific circumstances of the organization's culture and the project. To start, the project team should assess the organization's readiness and whether the selected project is the right fit for agile (see Section 6 – Transitioning to Agile for more details). Important areas to evaluate include the organization's existing governance structure and project management processes to see if they align or can accommodate the adaptive values and principles of agile, and the level of management buy-in to both support and be an agent for change. It is important to invest the time, resources, and effort to establish the culture, expectations, and infrastructure to support the implementation of an adaptive methodology. Learning how to work in an

agile way requires practice, commitment, clear and timely governance, and learning by doing. For those with little or no experience, consider leveraging an agile approach for a smaller effort to demonstrate success and the team's proficiency before moving on to something bigger.

Myth 8 – Pure Agile is the Answer

Employing agile practices will not be the solution to all project management and IT development issues encountered with a traditional approach, as agile may not meet the varying needs of the organization. Doing anything new within an organization often introduces elements of additional project risk. In this kind of environment, the implementation of agile practices and principles should be done pragmatically and take into consideration the real-world environment in which the project is managed and the system is developed.

In Section 5, see Selecting the Right Approach for additional information on how to incorporate adaptive thinking.

To realize benefits associated with an adaptive methodology without an overhaul of the current environment, organizations can moderate the degree of change. In particular, a project that takes place in a government context will likely be more successful if it integrates adaptive and user centered practices into its traditional waterfall approach. This could be due to rules, regulations, or the organizational structures and cultural expectations that are heavily based on traditional waterfall processes. For small changes, organizations can consider incorporating the following practices to be more adaptive:

- Have lessons learned frequently to continuously improve processes and practices, not just at the end of the project.
- Have short (15 minute) daily stand-up meetings to provide a venue for project team members to communicate roadblocks they are experiencing, and for management to help resolve.
- Manage each project team member's work-in-progress. Set clear and realistic expectations for what work can be accomplished in a given period to not over-allocate resources. This requires the team to prioritize its work and accomplish the most critical tasks first.

For organizations that are ready for bigger change, see "Section 5 – Selecting the Right Approach" for additional details.

Myth 9 – Agile is Undisciplined

Agile project management and system development practices are not only demanding of the project team, but they also require the support and a shared commitment for success by the leaders of the organization. The continuous integration and test-driven development of agile requires skill, coordination, collaboration, and discipline from the entire project team. Successful agile teams consistently deliver quality product increments that demonstrate working functionality in short time frames to provide value and benefit to the organization. To achieve this level of delivery, the leaders of the organization must delegate authority to the team to enable them to make decisions rapidly; this requires a high degree of developer and team discipline.

4. Waterfall and Agile

Compares key agile principles and values discussed earlier with waterfall principles and values.

There is a spectrum of project management and system development approaches available for projects to choose from, anchored on either ends by the predictive waterfall approach and the adaptive agile approach. To better describe the ends of this spectrum, the following sections will compare and contrast key characteristics of agile with the more familiar aspects of waterfall. The table below identifies the characteristics that will be explored and summarizes the difference between waterfall and agile approaches.

Characteristic	Waterfall	Agile
Scope and Resource Management	Fixed scope with estimated resources and time	Fixed resources and time with estimated scope
Product Development Cycle	Product delivered at the end with a linear, phased approach	Product delivered incrementally with short term, iterative development cycles
Project Management Planning	Detailed, long-term project planning completed prior to execution	Continuous planning based on iterations
Team Composition	Definitive team roles with individual assignments of accountability	Flexible, cross-functional team roles with the team sharing accountability equally
Stakeholder Involvement	Stakeholders are typically involved at the beginning and end of project development	Stakeholders are deeply involved throughout project development

Scope and Resource Management

With waterfall projects, requirements are developed and thoroughly documented in project planning in order to come up with a fixed scope. Resource and time estimates are generated based on the fixed scope and are tracked and managed throughout the project. As the project goes through the development phases and new information identified necessitates a change to the scope, the project goes through a structured change control process and resources and time are adjusted accordingly.

As shown in Figure 4-1, agile places emphasis on resources and time over scope. Agile projects commit to delivering project outcomes within a fixed time frame with fixed resources. With time and resources being constant, scope that is not core or critical to achieving the project outcomes become the constraint that is most negotiable. Design features or functional elements that are not deemed core or critical to the project are flexible and are dependent upon the amount of time and resources available for the project. Another aspect of scope flexibility allows the project team to make the decision to shift delivery of functionality from iteration to iteration, as long as core or critical functionality is delivered before the project ends.

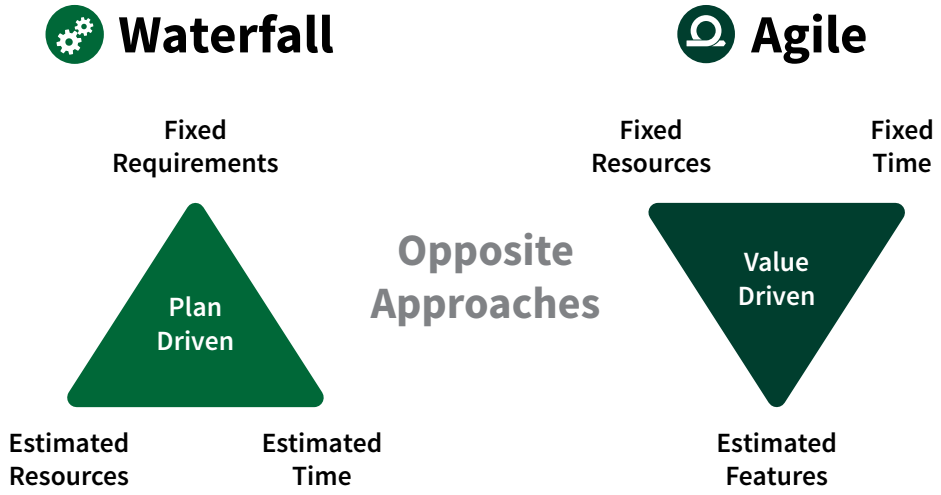

Figure 4-1 compares how scope, resources and time are perceived in traditional waterfall versus agile.

Agile contends that non-essential functionality within scope can and should be sacrificed in order to deliver higher priority items. This requires the prioritization of scope in a continuous, iterative, and collaborative process. It allows new information discovered through ongoing collaboration between the team and users to result in an adjustment in the priorities. In this way, functionality with the highest business value is prioritized for delivery first and elements that are outside of the core functionality are prioritized last.

Product Development Cycle

The development lifecycle for traditional waterfall follows a linear approach where each phase is completed in sequence, culminating in the delivery of a product or service at the end of the project. Although titles may vary among organizations and projects, the series of phases often follow the pattern of plan, analyze, design, build, test and implement. The deployment of all functionality is delivered once at the end.

By contrast, agile includes iterative development cycles that focus on building functional product increments of the overall solution, repeated over the course of the project. Figure 4-2 shows how functioning product is developed incrementally with several product releases over the life of the project, in contrast to traditional waterfall. Through continuous integration, product increments are frequently released to users to provide business value up-front and on an ongoing basis.

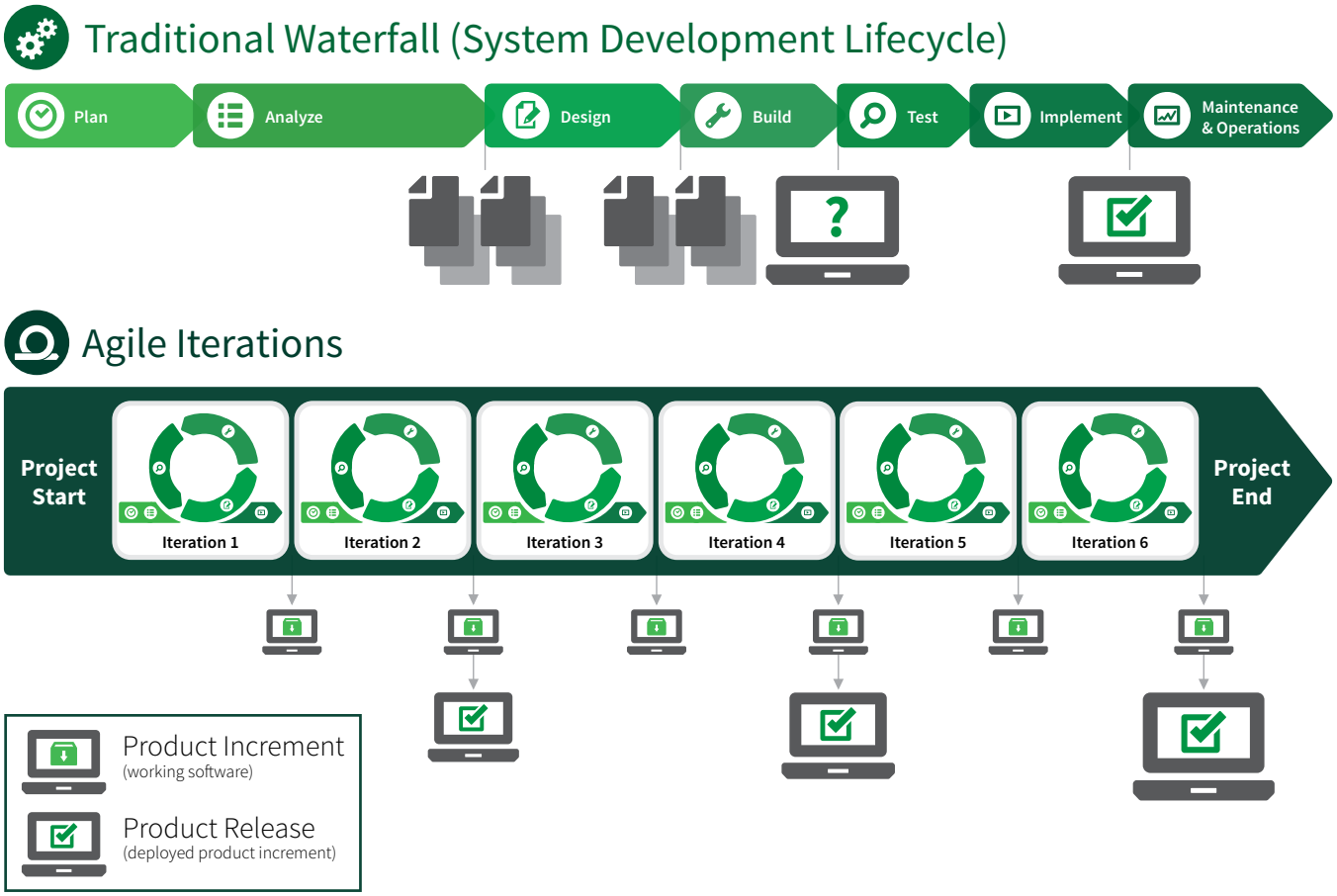

Figure 4-2 contrasts the output of product development in traditional waterfall versus agile.

Project Management Planning

Project management planning practices also vary between waterfall and agile, with the biggest difference being the depth and timing of project management activities. Project management for waterfall requires detailed planning that begins after project initiation with the development of the Project Charter. For a period of time, prior to any system development activities, the project team documents the processes, procedures, and controls that will be carried out during the project that are intended to ensure that the entire project scope is delivered on time and within budget. Over the course of project execution, project variances identified must go through governance channels for analysis and approval prior to implementation; significant changes may require project re-planning efforts.

This section combines and elaborates upon the concepts of iterative planning and flexibility of delivery from Section 2.

Agile places more emphasis on the ability of the project team to react to change. Planning occurs continuously, though at a high level in the beginning of the project, to form a prioritized framework for what the project team sets out to achieve. Like waterfall, this includes developing repeatable processes for disciplines such as communications, governance, change control, risk and issue management, etc. However, with agile, the project team has the flexibility to manage work as necessary, with the focus on incrementally delivering system functionality. The decomposition of these priorities into specific activities occurs continuously throughout the development cycle as more information is learned with each iteration. Typically bound by the fixed constraints of time and number of resources, project teams have the liberty to actively manage scope. The team makes decisions based on the goal of delivering functionality that presents the highest business value to the users first. However, in the context of government, scope can only be managed to the extent that all core functionality (e.g., legislatively mandated functionality) is delivered by the end of the project.

Figure 4-3 depicts the continuous elaboration and re-prioritization of system functionality. Scope items are initially ordered from high to low priority. Starting with the highest priority, the project team decomposes the scope item into smaller work efforts that can be completed within an iteration (A). With the business owner providing direction, the decomposed items are considered individually and all scope items are re-prioritized. This may result in placing components of scope at a lower priority than it was previously based on the business value offered. During the iteration, the highest priority scope items are developed into a functioning product increment. To plan for the next iteration, the project team will repeat the process of taking the highest priority scope items and decomposing the work, if necessary (B). Individual scope items are re-prioritized, the highest priority items are developed into a working product increment, and the process repeats until all scope items are developed. This example describes the type of planning that can occur at the iteration level, but, as depicted in Figure 2-3, similar planning activities may occur at all levels of the project and organization.

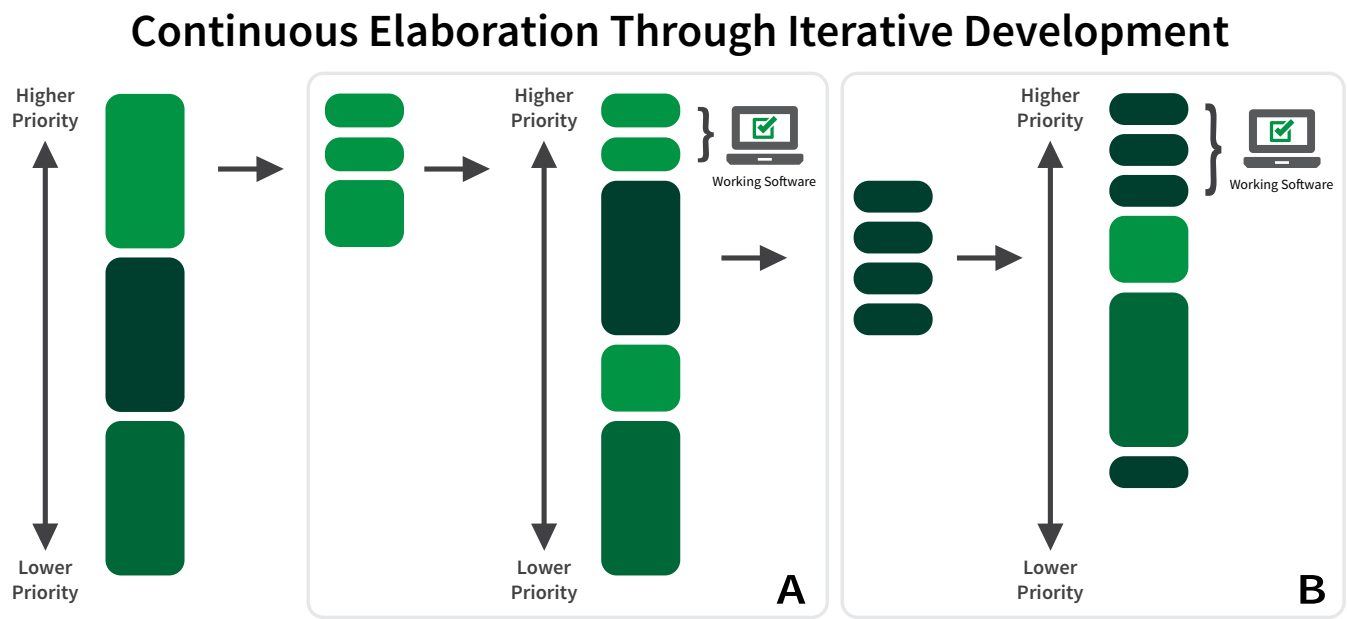

Figure 4-3 illustrates how continuous elaboration occurs through iterations.

Wherever the project team ends up on the project management spectrum between agile and waterfall, users will benefit most when project teams prioritize their needs when making decisions. User-centered approaches to project management and system development lead to meeting users' most important needs earlier and with greater consistency across an organization's project portfolio.

For more information on traditional waterfall planning, see the California Project Management Framework (CA-PMF): capmf.cio.ca.gov/

Team Composition

Many waterfall project teams consist of separate roles specializing in specific skills or expertise. For example, an analyst typically hands off requirements to a developer, and then a developer hands off the product to testers, and so on. Conversely, agile teams are cross-functional; each member of an agile team can perform one or more of the skills required to take a requirement through all phases within an iteration. Agile teams typically consist of at least three people to ensure there are enough skills, but no more than nine to minimize complexity of communication and collaboration.

Figure 4-4 illustrates the contrast between teams with undefined roles and defined roles. Each letter represents a different team member and each color represents a different skill. With more adaptive cross-functional teams, each team member has one or more skill, broken down by the differing sizes and colors. Collectively, the team has more than adequate depth in all the skills that

are needed. With more traditional specialist teams, each team member has a specialty skill and serves in a defined role. While cross-functional roles are more prevalent in the private sector, it is desirable for public sector organizations to develop cross-functional teams, as well. In doing this, the organization lends itself to supporting adaptive methodologies.

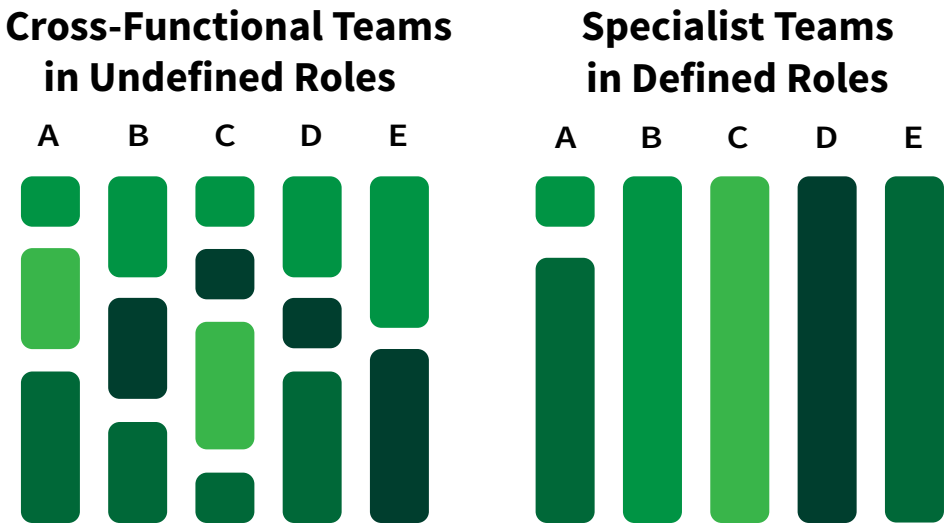

Figure 4-4 illustrates what team composition looks like in an environment with cross-functional teams in undefined roles versus specialist teams in defined roles. Each letter represents a team member and each color represents a different skill.

Stakeholder Involvement

During project planning, there is similar Stakeholder involvement for both waterfall and agile. Stakeholder involvement differs in the development lifecycle. On waterfall projects, there are two pivotal moments in the development lifecycle where Stakeholder involvement is critical to ensure the delivered solution meets the users' needs– during requirements development and acceptance testing. During design and build, the project team will work to achieve the defined requirements with minimal engagement of the Stakeholders until they have the opportunity to validate the requirements by using the finished product during testing. Traditionally, Stakeholders receive project communications, but are not necessarily actively engaged in two-way exchanges.

With the incorporation of iterative development cycles, agile teams are able to solicit Stakeholder feedback on a continuous basis. Stakeholders are engaged before and during each iteration to provide input into the functional priorities. At the end of each iteration, Stakeholders are provided the opportunity to see working components of the product that they can react to and provide feedback.

5. Selecting the Right Approach

Provides guidance for selecting the right approach for your project and organization.

A waterfall approach is best aligned to projects that have a clearly defined and complete set of requirements. It is expected that Stakeholders have a clear vision of the project scope; significant changes to scope are not intended after planning is complete. Because the complete product is not delivered until the end, emphasis will be placed on quality while speed of delivery is not a priority. The strict, linear methodology fits a project team and sponsoring organization that requires structure in the team's day-to-day activities and a detailed project plan for the organization to march towards.

Agile may also be feasible for projects that have a clear scope, where an adaptive approach would be effective in promoting early and frequent delivery; however, agile is also well suited for projects that do not have a clear and detailed vision of what the system should look like, how it should work, or what features need to be included. This lack of clarity upfront leads to changes in Stakeholder priorities and possibly scope as more information becomes known during design and development. An agile approach encourages this type of scope flexibility through progressive elaboration of the needs and wants of the users, making it an ideal approach for these situations.

Another need that an agile approach can help address is rapid and incremental delivery of functionality to users. However, with functioning product increments delivered frequently, speed of decision-making and team action becomes an important factor. Creating and maintaining an environment that supports rapid delivery requires a supportive governance process, as well as project team members that are skilled and can work independently without detailed guidance on what needs to happen next.

Many state organizations may not be ready to fully adopt agile as a project management and/or development approach, but certain agile practices related to the organization and project planning can be leveraged in a traditional waterfall environment. By incorporating adaptive thinking and a user centered design approach, organizations can make incremental changes that can bring about increases to user satisfaction that is a cornerstone of agile values and principles.

One such option is a modular approach where the linear project management and development cycle of waterfall is utilized, but the project scope is broken into sizable pieces or components of the full project scope, as illustrated in Figure 5-1.

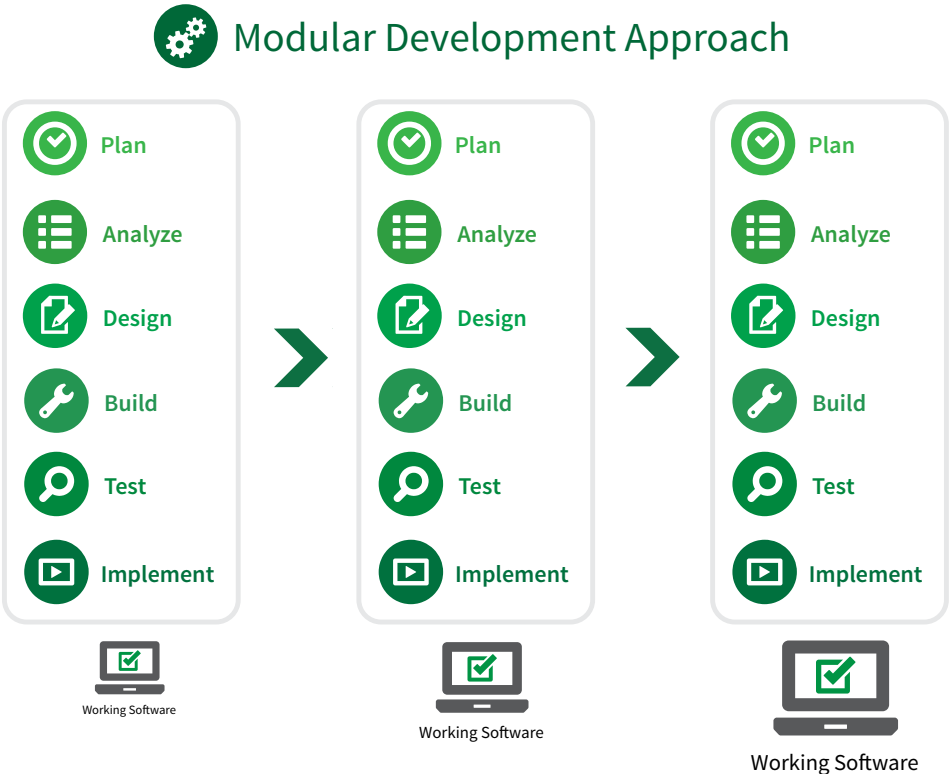

Figure 5-1 depicts the progressive deployment of functionality in a modular approach using traditional waterfall.

This approach provides the project team the familiarity of the waterfall project management lifecycle and development cycle across the entire project, but accelerates the delivery of functionality over several releases. Though use of this modular approach is still a longer cycle than an agile iteration, development time frames will be shorter than traditional waterfall and release of functionality will not be dependent on completion of the entire product or project. Stakeholders will not have to wait until the end to realize some benefits, and the project team will have multiple opportunities to obtain Stakeholder feedback and incorporate lessons learned into each release. Organizations using this approach can be more aggressive over time and have more, shorter releases, to the point where iterations of 4 weeks or less become attainable and there is a natural progression to move towards agile development.

When deciding the degree of which to adopt agile, consider the following questions:

- Is the organization ready and able to change to adaptive ways of thinking and doing?
- Would the culture of the organization support empowerment and decision-making at the team level (instead of the management level)?
- Are there team members with cross-functional expertise that can be dedicated to the effort and be collocated?
- What business goals and user needs will be addressed?
- Are the Stakeholders/users willing to engage frequently and commit the time?
- Are there well-defined requirements? How likely is it that these requirements will change over the course of the project?

6. Transitioning to Agile

Tips to ease organizational transition to agile.

While agile methods are not overly complex in and of themselves, it can often be very difficult to transition to agile wholly or in part, especially when many state organizations are entrenched in traditional processes and cultural norms that are difficult to move away from. A variety of factors contributes to successfully adopting agile processes and practices, both for the teams and the organization. State organizations considering a transition to agile can start with a recognition that the transition may begin with a series of small incremental shifts in the culture of the organization or the way a project is planned. Incrementally incorporating adaptive thinking is key to successfully navigating the transition to being agile.

Incorporate these practices on any project to be more adaptive and responsive to user and project needs.

To institutionalize a culture that supports adaptive strategies, state organizations should create a plan to identify and address the potential challenges it may face. This may include identifying and developing multi-skilled, collocated team members, and developing a governance strategy that provides the necessary support to make adjustments if teams struggle with the transition. A few of the success criteria for being more adaptive in pursuit of being agile are provided below.

Access to Business Owners

Agile projects are most successful when a business representative can be dedicated to representing the users and business owners of the organization on a daily basis. As depicted in Figure 6-1, identifying someone who can be dedicated to the project, accessible to the development team, and empowered to make difficult business decisions on a daily basis increase the chance for success.

Figure 6-1 shows the business owner as the conduit of communication between the development team and the Stakeholders/users.

Multi-skilled Teams

Teams must be equipped to effectively self-organize and determine how to complete the work. For developers, this means possessing the skills needed to take a requirement from inception to completion. For all team members, this means exceptionally good judgment, risk management, collaboration, and decision-making skills. Ideally, each member of the team possesses multiple skills to remove any single points of failure.

Ability to Collocate Teams

Face-to-face communication is often the most efficient and effective method of sharing information for understanding and agreement. Removing communication barriers drastically increases the chances of an agile team being able to effectively self-manage, self-organize, and understand the needs of the user.

Executive Support

Team-level adoption of agile practices will fail without leadership support at all levels. Although teams may adopt agile techniques tactically, support at the strategic level enhances their success at organizational and inter-team interactions.

Stakeholder and User Involvement

Stakeholders and users need to be willing to participate regularly and consistently in review meetings and offer feedback. This is very important when implementing a user-centered delivery approach.

Pilot Approach

Development of a pilot project using agile can be a less risky approach to gauge the project team's capability and proficiency with agile practices. Having a project team employ agile on a small, localized project will allow the team to learn and, if successful, bring forward the knowledge and experience gained to continue agile development for the rest of the project.

Access to Agile Expertise and Transition Support

Organizations that attempt to transition to a full-on agile approach without embedded agile expertise and transition support tend to have trouble. Agile expertise provides in-the-moment course correction when adopting new practices and processes. Agile transitions require continuous learning and consistent course correction.

Flexibility with Documents and Meetings

Constraints for an agile transition likely exist in the state government environment, where existing reporting requirements and project documents are often based on traditional methods, as well as legislative requirements. When selecting an initial or pilot project to apply an agile approach, state organizations should choose a smaller, less complex project that doesn't have stringent reporting requirements and/or work with the appropriate oversight agencies to set expectations. Project teams require more flexibility with documents and meetings when adopting agile practices.

7. Next Steps

As State organizations evaluate adopting agile, consideration should be given to making small, incremental changes. If an organization is interested in becoming more agile in its project management and system development practices, follow these recommended next steps:

Steps to bolster the organization's readiness for agile.

1. Engage an internal or external certified agile expert to assess the current state of the organization and create an agility adoption strategy that identifies specific challenges that need to be addressed.
2. Create a roadmap of small, incremental changes based on the findings of the assessment.
3. Form an executive team to support the organization's transition.
4. Thoughtfully identify a pilot project for which it is appropriate to use a subset of agile practices. Successes early create momentum, but early failures in adoption create skepticism; choose the pilot effort with care.
5. Train all pilot program participants and Stakeholders on the selected agile implementation, including organizational leadership and decision-makers.
6. Kick off the pilot project with active, embedded agile coaching.
7. Gather feedback frequently and continually improve.
8. Share successes within your organization and with others outside your organization that may benefit from understanding your experience using an adaptive approach.

In addition to developing parts two and three of the agile guidance, CDT is exploring changes to current state policies to support the decision of departments to be more adaptive. Announcements will be made on CDT's website as they become available.

Visit the California Department of Technology website:

cio.ca.gov