

AGL Academy

A community effort by Agile government professionals to help educate and empower those who seek to implement Agile processes into their own agencies.

Powered by Agile Government Leadership

By bringing applied Agile practices to government, we want to redefine the culture of local, state and federal public sector service delivery across all aspects of government. We will work with Agile professionals and organizations to support their work in getting Agile infused into government processes. We will foster a spirit of openness and mentor those new to Agile so that they have the necessary practical advice, resources, tools and community support for successful deployment. Through Agile Government Leadership, we will create a responsive, engaged government that more efficiently and effectively serves its citizens.



Connect with AGL

- [Website](#)
- [Subscribe](#)
- [LinkedIn](#)
- [Twitter](#)

Agile for the Government Executive

[Welcome.](#)

[Contributors](#)

[Introduction for the Executive](#)

[Lesson 1: Introduction To Agile](#)

[What Is Agile?](#)

[Why Agile in Government?](#)

[Reading List](#)

[Video List](#)

[Responsibilities of the Agile Executive](#)

[How Agile Will Feel](#)

[Fundamental Concepts for the Agile Executive](#)

[Agile Assessment](#)

[Lesson 2: Tools of Transparency](#)

[User Stories](#)

[Strategic Objectives](#)

[The Storyboard](#)

[The Burndown Chart](#)

[Lesson 3: The Agile Executive Workshop](#)

[Introduction](#)

[Sprint One](#)

[Sprint Two](#)

[Sprint Three](#)

[Lesson 4: Agile Procurement, Communication, and Modernization](#)

[Reading List](#)

[Video List](#)

[Agile Procurement](#)

[Using Agile in a Legacy Environment](#)

[A Modernity Checklist for 2017](#)

[Exercises](#)

Welcome.

This material is designed for the executive in government who wants to lead their team to start practicing Agile.

Here you will find a step-by-step course to get educated on how Agile will benefit your agency, how it has succeeded in other agencies and what challenges it can help you overcome. You have several options for completing this course:

- Use this document (contains links to outside reading material and resources)
- Use our website (<http://www.agilegovleaders.org/academy>)

Before you begin, please note:

- This course is intended for the Executive in government who has heard Agile can transform development and management but wants to understand how to bring it to their agency.
- Although there are many Agile Methodologies, this document is about Scrum, a common and standard Agile practice that is widely used.
- Agile processes are designed to help you serve your purpose, not to get in your way. However, most people have found that following Agile processes as taught in this curriculum allow you to produce software faster, with less risk, and with infinitely better customer satisfaction than older methods.

We at [Agile Government Leadership](#) hope this empowers you and your team to wow your citizens with efficient delivery of phenomenal digital services. We welcome and appreciate [feedback](#) on what could allow us to iteratively improve this curriculum for the next government executives who take on this challenge.

Contributors

This course was designed by Agile government professionals with combined experience at the federal, state, and local levels, in addition to the private sector. Contributors to this course are members of the [AGL Working Group](#): Doug Birgfeld, Tim Nolan, Elizabeth Raley, Robert L. Read, Son Tran, Joshua Smith, and Mark Vogelgesang.

Introduction for the Executive

Government executives make big decisions. They decide which projects move forward, which get killed, and how many resources each project gets.

Agile methods were created to transform software development, but most executives are not writing code. Therefore, some aspects of Agile software development are critical to programmers but meaningless to Executives, because they have no bearing on the decisions that they must make.

But other aspects of Agile development are absolutely key to the job of making decisions. Agile offers transparency into the progress and status of a team. The transparency means that you, the Executive, can make the best decisions possible. It does not mean that all your dreams will come true. But it means that you will have a great deal of control over what you are building. You will have predictability.

A typical experience for an executive using “waterfall” methods is to learn in the final part of a project that the project is going to take 50% longer than expected. Agile methods do not promise that projects will be done faster, but they do promise that you will be aware of the progress and will know instantly if you start to fall behind schedule. It will be clear earlier in the process that you may need to take some executive action.

Will that decision be to extend the deadline of a project, or to cancel a project altogether? It depends. Agile methods give you the information you need to make that decision.

Even with all the influence you hold as an executive, your power is limited in some ways. You may be stuck with legacy software that burdens your pace of change, through no fault of your own. You may have trouble convincing everyone in your organization to embrace the cultural changes that Agile brings.

This course arms you, as the executive, with the power you need to support cultural change within your organization. If you study these lessons and perform all of the exercises and workshops in this course, you will have positioned your staff and yourself to gain the benefits of Agile mindset and methodology.

Lesson 1: Introduction To Agile

GOAL: Gain a shared terminology and understanding of Agile and Scrum, along with the motivations for using them in government projects.

What Is Agile?

Agile is an iterative and incremental method of managing and developing projects with a team. It focuses on customer collaboration, responding to change, and frequent releases of working software.

The [Agile Manifesto](#) is a single, simple, brilliant sentence that you should refer to often:

“Through this work we have come to value:

- 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, we value the items on the left more.”

Let’s take the manifesto and put it into the context of real work, real customers, and the real world. Agile takes the best of the world’s history of getting things done and rolls it into a single framework. For instance, we know that people work better in teams; we know that trusting relationships have better results than legal documents; we know that predicting the future is super hard; we know that the proof is in the pudding. We also know that 3 things 100% done are more useful than 10 things 75% done. And finally we all know the phrases: “one day at a time”, “one game at a time”, and “keep your eye on the ball”.

Agile takes these intuitive truths and brings them into the world of development, software and beyond. Agile is not a prescriptive dogma. It is a set of activities and methods that can be used individually to make things better, and collectively to make things great.

Agile takes practice and discipline. But just like that play you were in, or the concert you practiced for, you will be happy with the ovation at the end. It's worth it.

Why Agile in Government?

The Agile Manifesto follows [12 Principles](#) that focus on value, collaboration, motivated people, change, rapid delivery, and simplicity. These same principles apply to government regardless of whether your agency is Federal, State or Local. The principles offer a pragmatic approach for working with customers, citizens, employees and each other.

An Agile government provides excellent services and value because managers trust their teams to do the work and to collaborate directly with customers. Instead of micromanaging and frustration, there is daily communication and collaboration. This culture of transparency rubs off on the customers within an Agile government -- and soon, the culture of the agency begins to shift.

With Agile processes paving the way for delighted customers, a government's focus can be on providing genuine happiness instead of settling for "good enough" or shrugging off another failed IT project. Eventually, a government practicing Agile will adopt a permanent Agile mindset. The 12 Principles behind the Agile Manifesto will become self-evident instead of being something to strive for.

The easiest way to identify and explain what Agile offers to your agency is to replace it with "flexible". We want our government to be flexible to meet the changing needs of our customers and citizens. We want our government to be flexible to produce the greatest value within the allotted time frame and budget.

Being Agile (flexible, transparent, efficient) should be the goal of every government agency.

Reading List

Study each of these resources and discuss what you're learning with a colleague.

[Agile Government Handbook](#)

AGL has compiled a handbook that outlines Agile in the government space.

Study Time: 90 minutes.

[Government Challenges to Using Agile](#)

Read about gov-specific challenges that you may face in moving your team to Agile methods.

Study Time: 10 minutes.

[10 Steps in Making Your Agency Agile](#)

Written by a Government executive explaining how to bring Agile to any agency.

Study Time: 10 minutes.

[5 Cultural Shifts Agile Brings to Agencies](#)

Learn the fundamental culture shifts agencies must undergo to become Agile.

Study Time: 10 minutes.

[Agile Terms](#)

New vocabulary will come up while talking about Agile and Scrum. This is a cheat sheet to ensure the team has a shared understanding of what is being discussed.

Study Time: 20 minutes.

Video List

Watch these videos to gain a deeper understanding of the how-to behind Agile and Scrum.

Continue to discuss your findings with team members to ensure that you understand the concepts.

[Agile 101 Case Study Discussion](#)

Study Time: 1 hour

[How to Run an Agile Project](#)

Study Time: 1 hour

[How Agile Works for Government](#)

Study Time: 1 hour

Responsibilities of the Agile Executive

As an Agile Executive, you have many responsibilities:

- You must ensure an appropriate Agile working environment. This means not skimping on equipment. More importantly, you must create an environment that rewards honesty and clarity. You must create an organization where anyone is free to deliver bad news without fear of disapproval. Honesty is the basis of Agile development.
- You must create an environment that values providing utility to the end-user. You must ensure that your team has access to end-users (sometimes called “customers”, although in government work the final stakeholder is rarely a paying customer.) The end-user must be valued more highly than other stakeholders. The software is not being written so that an Executive can track it, but to provide service to the end-user.
- You must make mature decisions. This means making rational decisions instead of emotional ones, based on the good of the end-user rather than appearances.
- As the Executive, you are in the best position to pay attention to the big picture. This means that you must make sure your strategic goals are properly matched with serving your customers. It is your responsibility to ensure that what the Engineering team is doing stays true to your organization’s objectives.

- Make sure that your directors and managers are actively fostering an Agile environment, and are insisting on a user-focus.

How Agile Will Feel

When you are becoming more Agile, you should expect to have some trouble. This table will you give you an idea of how this growth process will feel.

Activity	Why	You know it is working...	How it Will Feel
Shifting to Burndown Charts	To make good management decisions based on accurate progress	When you trust that you really understand the impact of your decisions on your schedule	Like a rusty wheel finally starting to turn smoothly, you will gain control of your organization
Agile Procurement	To de-risk giant projects	When you start seeing software in the first quarter	Like leaping off a cliff and finding that you actually can fly
Using formal Sprints	A consistent rhythm of valuable habits will keep focus on progress and users	When people perform Sprints without fuss	Like starting an exercise program, then realizing that you can actually run a lot further now due to consistency
User Focus instead of Contract Focus	Nobody is smart enough to know what the user wants without focusing on the user	When you are surprised to learn that the user seeks something different than your supposition	Like looking for your keys and realizing they have been in your hand all along

Fundamental Concepts for the Agile Executive

As an Executive, you do not have to understand all of the minutia of the Agile process as a software engineer does. However, you have to deeply understand the following fundamental

concepts which will be extremely valuable to use, and which impinge upon your responsibility as an Executive.

Velocity and Transparency

Perhaps the most valuable concept to the Agile Executive is the concept of *velocity*. It is easier to understand velocity when you have performed one of the recommended workshops for this course, which illustrate the concept well.

In a nutshell, the basic unit of work in an Agile process is the user story. Your engineering team and product owner will have to work to perfect their ability to write, estimate and manage stories. But you must understand that each story gets an estimate, which is preferably NOT done in actual time, but in an abstract measure such as “story points.” The estimate is a representation of how much effort it will take to complete the story.

After each iteration, the estimates of stories which are definitely proven to be completed are summed together to create the “velocity” for that sprint.

The value in velocity is that over time it becomes a reliable predictor of the expected progress of your teams. After about three sprints, you can reliably expect a team to get approximately its average velocity done in each sprint.

This means that with properly estimated stories in an Agile process, you will never be forced to retract a promise made to your own superiors, because you will never be greatly surprised by a lack of progress on the part of your team.

Control of Priorities

User stories provide a convenient means of controlling priorities. A prioritized set of stories is called a Backlog. In general, your product owners, who are the voice of the end-users of your system, will set the specific priorities in the Backlog.

Your responsibility is not to set the priorities of the individual stories, but to make sure that the Backlog is the guiding force for your teams. You should periodically ask to see the Backlog, make

sure it is being kept up-to-date, and ensure it is actively being used as the guide for your engineering team.

These methods provide you, the Executive, with something precious: transparency and control. It prevents the embarrassment of learning at the last minute that a software project is behind schedule. Imagine having no more missed deadlines and no more catastrophic failures.

This is not an inflated exaggeration -- Agile methods really prevent drastic negative surprises. They do not, however, mean that software will be produced faultlessly at any rate you desire. Rather, they give you perfect transparency into the actual state of your project on a continual basis. You may well be disappointed by the rate at which your team produces software, but you will never be surprised, because each iteration you will have an honest measure of your actual progress or lack of progress.

Building an Agile Organization

As the leader, you must drive toward the true goals of your organization. That is, you must ensure that if you successfully build the software you are planning, it will actually meet your strategic objectives. The best way to do this is to ensure that the end-user is “in the room” with the product owner and the engineering team. Sometimes this is physically true: you should facilitate actual meetings, organized around the testing and demonstration of the software. But it must always be metaphorically true: the end-user must always be involved and available to the Engineering team.

You must also encourage and defend investments in learning for your organization. It is all too easy to demand progress at all costs from your team -- but this is penny wise and pound foolish. As the Executive, it is your job to set the tone and principle that all members of your organization must be constantly learning.

Finally, you must build an organization that eschews all deception. Too often in government there is a learned attitude that you must ask for more than is really needed in order to get what you want. Rather, set the precedent of complete transparency and honesty.

Establishing Communication

To be a good Executive, you do not have to be a programmer, but you must spend some small fraction of your time with your programmers. You must establish personal communication with your engineers, and make it clear that any may speak to you freely.

By spending a small amount of time with your engineering team, you will earn their respect and avoid the catastrophe of having your team fail to be perfectly honest with you. You may be afraid of looking stupid in front of programmers, but do not let that deter you. Your programmers will respect you much more if you communicate without arrogance to them than if you do not communicate with them or pretend to know more than you do.

We recommend that you spend 75% of your time with your subordinates. You will naturally spend more time with your direct reports than with their reports. However, be sure to spend a small amount of time with each person who reports up to you, no matter through how many channels. If you manage a very large number of people, you may only spend a few minutes with each person each year, but the principle holds true. You can not lead effectively if you do not understand what the people in your organization, at all levels, are thinking.

Embracing Change

Many people believe that if software is incomplete, it is useless. This is the way engineers think about problems. A bridge which falls down is useless. A bridge which does not span the chasm is useless. There is never a baby bridge that has a useful existence on its own.

But we have learned that software can be developed in a way that allows it to grow slowly through the accretion of individual units of functionality. You don't need all of the units to be known before you begin work.

Sometimes you hear people say, "Change is expensive, so we should cover all our bases and be sure we do the job thoroughly the first time." However, experience has shown the opposite is true -- that if you accept and embrace change, you will actually go faster and with much lower risk.

Embracing Change in Real Life: Collin County's Story

"In Collin County, Texas, the Public Information Officer (PIO) worked with the IT AppDev team to update our Case Look-Up, Active Warrants and In-Jail applications last year. These three apps represent 65% of traffic on our website. Our goal was to responsively design all three apps so that they would look good on smartphones and tablets. During a sprint review session, one of our team suggested that 'it would be neat' if we could link the Active Warrants app to the In-Jail app so that law enforcement would not have to serve a warrant to an empty domicile. The PIO responded, 'Can you do that?'" The Dev team said sure, they could link all three apps together. As a result, we dedicated one additional sprint to build the [Judicial Online Search](#) that consolidated all three previously independent apps into one. We experienced a requirement change in the middle of the development of the three different apps -- and ended up making a better single solution. "

Tim Nolan, Senior Applications Manager, Collin County

Agile Assessment

Determine your team's current Agile capabilities using this [Agile Assessment](#). This will help you see how Agile will be supported or challenged by your agency's leadership, culture, policies, etc.

CONGRATULATIONS, YOU'VE COMPLETED LESSON 1!

Lesson 2: Tools of Transparency

GOAL: Understand how Agile provides transparency and why this is essential to the management of multiple teams.

As an executive you'll find that transparency into the project and the ability to change direction without total derailment is incredibly useful. Below we'll introduce components that your Agile teams will use to facilitate these aspects. These are the tools of transparency.

User Stories

As an Executive, you are unlikely to be directly concerned with writing user stories. Nonetheless, you need to understand how and why user stories are created, since they are basic unit of work in an Agile project.

A user story must provide value to some user. An Agile process is driven by the completion of stories, each of which provides tangible, demonstrable value to the user/customer/stakeholder. A sprint consists of a set of conscientiously prioritized stories. Experience will show that it's best to use a format for each story that identifies **who** the user is, **what** they need, and for what purpose (the **why**). Such stories are written in this format:

"As a ____, I need a ____ in order to ____".

The **who** in a user story could be someone with a particular functional role, who holds a certain title, comes from the perspective of a persona, or embodies the needs and behaviors of a hypothetical user.

The **what** in a user story details in specific terms the need, feature, or functionality desired by the who. This is what your project team will build into the product or service.

The **why** in a user story states the value. It presents the needs of your users and customers up front and center.

Here's an example of a user story that clearly defines the **who**, **what**, and **why**: "As a jazz fan, I need a tuning knob in order to find a jazz station on the radio that I will enjoy listening to."

Keys to a Valuable User Story

- Product Owners must have courage to ask for what they believe their users/customers/stakeholders really want.
- A story must have value to someone. It must make the product better in some way.
 - The story when complete will make a real-world task faster, better, easier to understand, have fewer steps, or collect better info.
 - The high priority stories affect the most users or procure the highest value data.
- "Clean up the bugs we introduced in the last sprint" is NOT a user story because it does not add anything to the product.
- [Remember the INVEST model!](#) Good user stories are:
 - Independent
 - Negotiable
 - Valuable
 - Estimable
 - Small
 - Testable

By making user stories Independent (producing workable features that can stand on their own), Program Managers are given the ability to make intelligent tradeoffs. This power translates into an advantage for you, the Executive, because it allows you to also make budgeting decisions without being beholden to artificial limitations. On an Agile project, you should never hear a subordinate say "We absolutely cannot release on that date," because the software is always releasable -- each iteration should be workable on its own, albeit with room for improvement.

Ensuring that user stories are Independent is the role of engineers and the writers of user stories. You should insist that your teams keep good backlogs of stories and that they follow the [INVEST](#) acronym rules. You may participate in writing user stories occasionally, although that will be rare.

Strategic Objectives

Before beginning any work to create a new product (such as a website) or improve an existing product, an executive should establish and communicate strategic objectives for the product. The strategic objectives for the product provide an outline or framework that guides all of the product's features and requirements.

In an Agile process, the aggregated collection of all user stories for a particular product is called the Backlog. Each individual user story, and collectively the Backlog of user stories, should contribute in some way to the product's strategic objectives. If user stories lose alignment with the product's strategic objectives, either the user stories should be changed or the product's strategic objectives should be changed. Keeping these components aligned improves the likelihood that the effort expended on a product will have the maximum intended business impact. If this alignment is not a focus area, Agile project teams may stray away from the original objectives to new objectives without communicating the change.

Executives will usually not be involved in the daily project mechanics, but Executives should use their interactions with the Agile project team to set the expectation that individual user stories and the collective Backlog of user stories must support the product's strategic objectives. If the Agile process leads the team toward a new set of strategic objectives, executives should be open to hear proposals for changing the product objectives in a formal way. In this way, Agile allows both flexibility and consistency.

The Storyboard

The Storyboard is a Big Visible Chart that is really the heart of the project. It must be where everyone can see it. A whiteboard with sticky notes works fantastically well for this. This is another tool that lends transparency to the project, so everyone can see what is being worked on. The highest priority stories, visually, are on the top of the list in each lane.

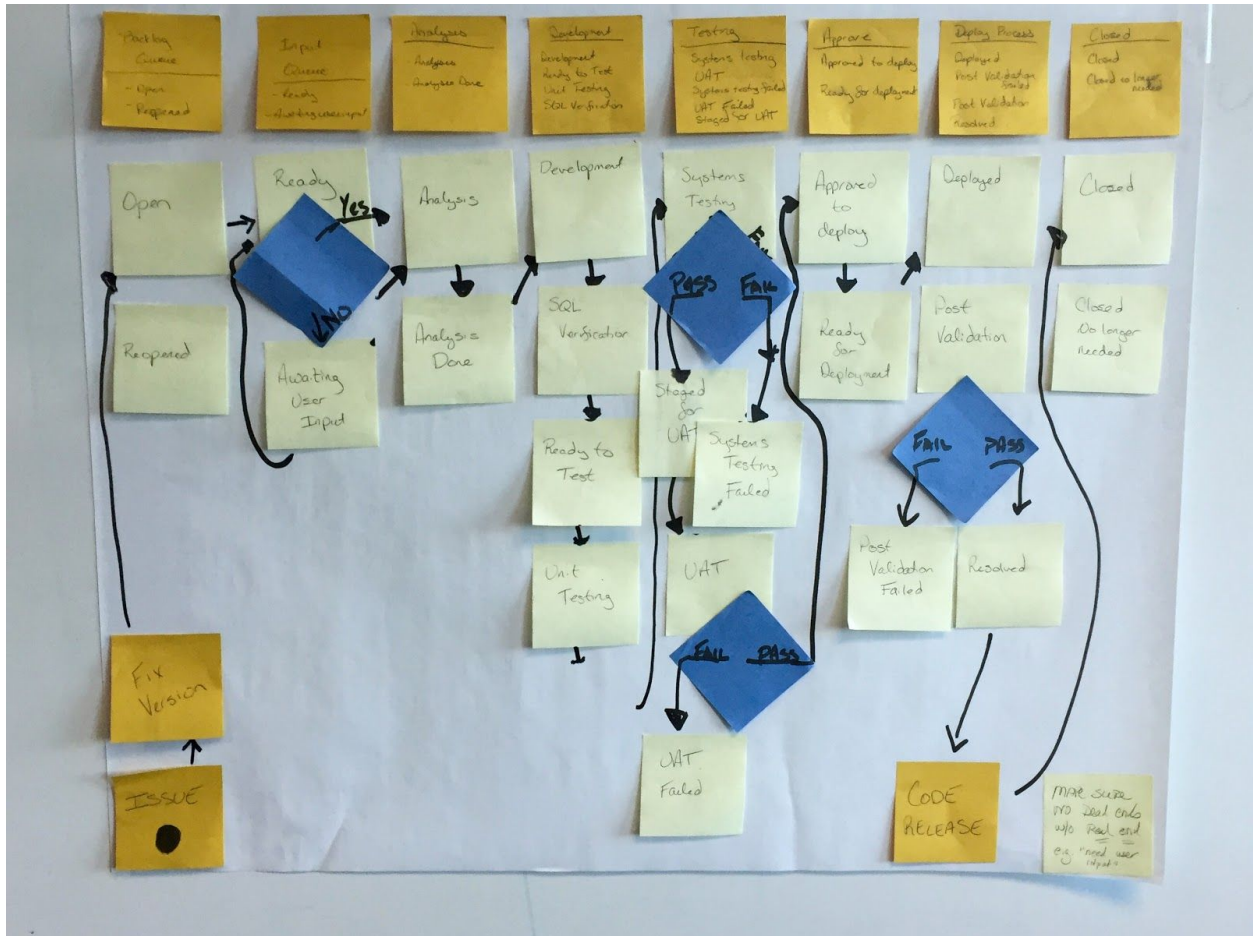
As an executive, you will not be working with the Storyboard as much as the Product Owner and Product Manager do. However, it is your job to make sure that each of your teams *IS* keeping an

up-to-date Storyboard, using it as a tool for managing their work within the project and for producing accurate, usable Burndown Charts (described next).

The Product Owner can put a story into the backlog at any time. The PO and others on the customer-facing team should be constantly “refining” the Backlog for stories ready to be placed on the board, based on changing business needs or because they have been able to complete a story with enough detail to be worked on.

Here are some actual working boards:

Sprint : Sprint 14 (SPR-102)						
Jan 26, 2016 - Feb 16, 2016 Status: Open-InProgress Velocity:1/31.1 Project Team: BPC (Pesticides)(PROJ-15)						
31.1 pts 100.2hrs rem.	New	Blocked	Open	Pending Verification	Completed	
Sprint Bugs						
Sprint Issues						
US-2756-1 Manage Pesticide Product Registrations 3 pts 12 hrs rem.	TASK-5612-1 Unit testing ProjectM... 3	TASK-5611-1 PO review ProjectM... 1	TASK-5479-1 QA Testing ProjectM... 5	TASK-5640 Test case PREP ProjectM... 1	TASK-5610-1 Design Melissa H... 2	
US-2566-1 Validate Pesticide Registrant Access 3 pts 3 hrs rem.	TASK-5600-1 Create Manage Product ProjectM... 0	TASK-5596-1 Update Nav Rule ProjectM... 0	TASK-5597-1 Create Declare Lookup Page ProjectM... 0	TASK-5474-1 Test case Prep ProjectM... 0	TASK-5601-1 Create Activity to Iterate Through Melissa H... 0	TASK-5598-1 Create When Conditional Rule Melissa H... 0
	TASK-5475-1 QA Testing ProjectM... 3					
US-2728-1 Pesticide License Templates 3 pts 4 hrs rem.	TASK-5454-1 QA Testing ProjectM... 4	TASK-5453-1 Test case Prep ProjectM... 0				
US-2819 Associate Individual to a Company/Agency - 1 pts 0 hrs rem.					TASK-5633 Development ProjectM... 0	TASK-5634 Unit testing ProjectM... 0
					TASK-5642 Test case PREP ProjectM... 0	TASK-5635 PO review ProjectM... 0
						TASK-5654 QA Testing ProjectM... 0

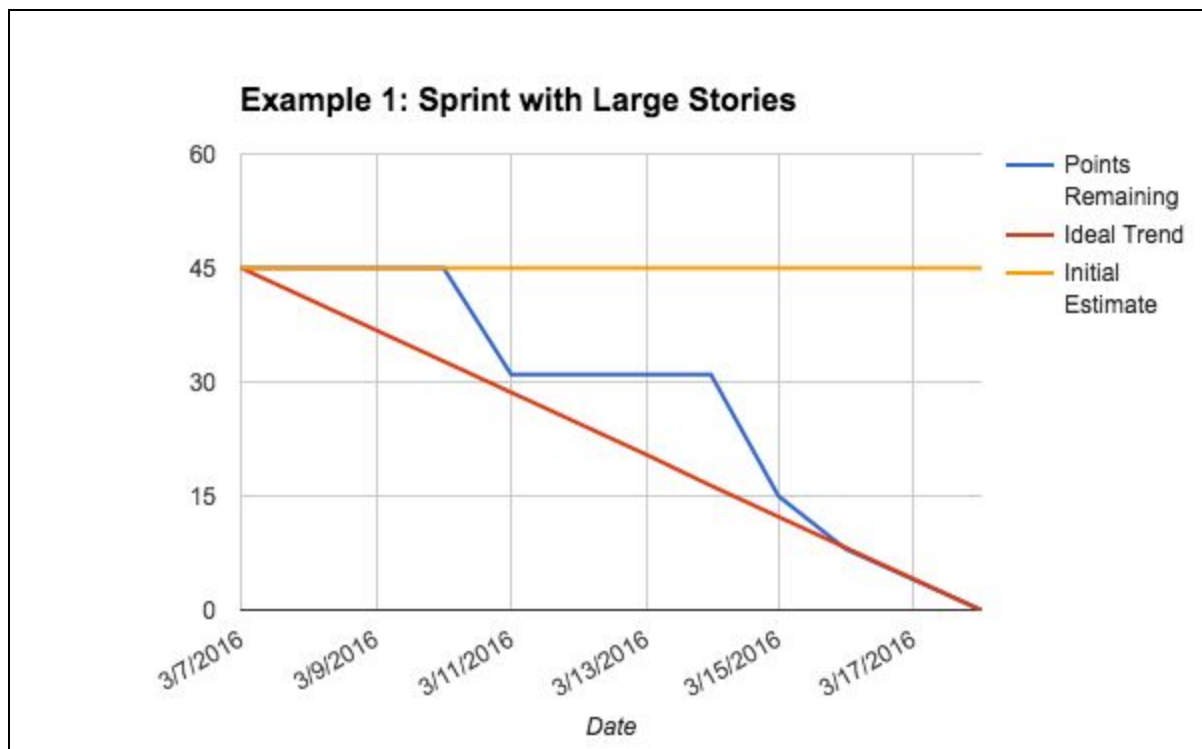


The Burndown Chart

The progression of stories and their assigned points, as they move incrementally from Backlog to Done, can be expressed in a Burndown Chart, shown below. This is one of the tools that gives transparency to the progress in the project. Every day it is updated, and it lives in a high traffic area so anyone on the team can view it.

The Burndown Chart is based on estimates of how much effort each user story is expected to take. Estimates exist to serve you, the Executive. The purpose of keeping estimates is to allow good management decisions to be made. It is difficult to estimate how long it will take for work to be done on a project, particularly within a Sprint or time box. An estimate is a guess.

The example below shows a team working in a two week sprint cycle. Based on historical experience known as velocity, the team knows that it can complete roughly 45 Story Points worth of work each sprint. Working with the product owner, they identified a mix of high priority user stories that added up to 45 points as represented by the yellow line. From the 45 point mark, a red line is drawn to last day of the sprint which produces an ideal burndown trend. Most teams will fluctuate above and below this line throughout the sprint. As the sprint progresses, user stories are completed and are represented by the blue line.



Hands-on Practice: Agile SIM

In order to allow you to simulate the resource management decisions you make as an executive in charge of multiple projects, we have developed an online game which illustrates these concepts:

- The Burndown chart and velocity serve to give you visibility into the progress of your projects.
- Teams vary in their effectiveness, which you must take into consideration.

- As an executive you get to make tradeoffs to try to provide maximum value, expressed as “Victory Points” in the game.
- Dealing with negative events beyond your control is part of your job.
- There are some actions you can take to improve the effectiveness of your teams.

In this lesson your assignment is to play the Agile SIM game for about an hour, which should amount to five or six game plays. We welcome your feedback on how this game has been helpful (or not) and how we can improve it.

>>> [Play now](#)

CONGRATULATIONS, YOU’VE COMPLETED LESSON 2!

Lesson 3: The Agile Executive Workshop

GOAL: Experience the Agile process from an Executive viewpoint and lead your team through three sprints that demonstrate Agile methods.

Introduction

This workshop will focus on the unique role of an Executive to a Scrum process in government. Unlike Project Managers and Product Owners (who are directly responsible for working with developers in the making of the product), Executives act as either force-multipliers or impediments to the team's ability to produce great product.

The Executive must create or provide the best environment and circumstances for the project to develop successfully. This three-sprint workshop allows you to practice doing this, highlighting some key skills you will need to develop as an Agile Executive.

- Sprint One - Defining the Mission, Goal, and Roles
- Sprint Two - Asking Agile Questions
- Sprint Three - Being a Servant/Leader

Sprint One

This sprint will help you learn to clarify the targets for the development team. As an Agile leader, you won't be micro-managing everything they do each day, so you need to make sure the development team has enough guidance to self-direct in the right direction.

FOCUS: Mission, Goal, and Roles (MGR)

TIME: 75 min

YOU WILL NEED: Whiteboard, Marker, Management Team and Devs

Preparation

For this workshop you can use either a real life goal that you and your team need to meet, or a fantasy mission of your choosing. Examples:

1. We need some kind of website for something
2. We need to improve customer service
3. We need to protect the castle from the barbarians
4. We need to overthrow the leader of the Illuminati

Once you select the mission, you will work with the group to devise a one sentence statement that is simple and clear, allowing everyone on the team to understand the project. Explain to everyone that during the hour-long workshop the team will:

- Devise a single sentence that describes the project or mission perfectly
- Establish 3-5 measurable, observable goals that must be met to complete the project
- List the types of general resources required to do the job
 - Could be a job title : “Lead Dev”
 - Could be task related: “Someone to make reports”

It is super important that you set out the goal for the workshop clearly and speak with confidence. Allow a few minutes for folks to shuffle in disbelief, but then crack on.

Defining the Mission

If the mission you are using is a real mission, simply explain the outcome you are seeking. If you are using a fantasy mission, tell a little tale to set it up. For example:

“Once there was a lovely kingdom near a wood, with fertile fields and sparkling streams. Sadly, this lovely place has lately had its economy damaged by barbarians. Our goal is to keep the barbarians from continuing to harm our lovely hamlet . . . “

Present it to the group and ask, “Can anyone here put this in a single sentence right now?” Someone will try. If not, then you try. The sentence has a noun, “ We” or “The Department”, so all that’s left is the verb and the object. Do not allow the group to put reasons or justification language in the sentence:

“ Because we are plagued by barbarians and our citizens are sad . . .”

“The system we have now is terrible and will die anytime, so we need . . .”

This kind of extra information is not necessary and just makes it hard to restrict your description to a single sentence. It may take as long as 25 minutes to finally arrive at a sentence that the team can agree on. Examples of good goal sentences:

1. We will rewrite all the copy on our website
2. We will make a new IVR call tree
3. We will build a moat
4. We will hack the central computer of a suspected Illuminati leader

Note the lack of detail -- these sentences are not long or full of disclaimers and positioning words. Nor do they say anything about the attributes of the goal we are seeking. We capture the attributes in the next section of this sprint.

Goals and Criteria

In this section of the workshop sprint, we define the criteria that must be met in order to claim the mission is a success. These are the terms of victory. Together with the one sentence mission statement, this becomes your team’s high-level “definition of done”.

This can take as long as 25 minutes to complete because you must ensure that each goal is something that can be observed or measured. You must also take care that these are not vague or incredibly obvious. Goals like “it must work well” or “it must be user friendly” are both obvious and unmeasurable, and therefore would not pass muster. You are aiming to create nice, meaty goals that give clear conditions that must be met. For example:

- Click-through traffic on website increases by X
- Customer services ending up with live operator are reduced
- Barbarian incursions completely end
- We are able to destroy Illuminati robotic soldiers without detection

To expand on the barbarian invasion example, some other goals might be:

- Must not block egress to courtyard
- Must not involve magic (remember last time?!?)
- Must be made of locally sourced fibers or materials

When you have 3-5 conditions you are getting close to done. You know you are done when the group begins to struggle to come up with new ones or when some suggestions can be grouped together because they are re-statements or sub-goals.

Roles

In order to accomplish any of the goals, it is important for Executives to understand how to recruit the best type of talent to join the team. Hopefully, the skills needed are already represented by the group, but it is possible that a goal on the list may require extra resources.

If we have used a fantasy mission, you may not have the exact resources required (i.e. medieval engineers) at the meeting, but you can still assign resources at a high level. We will work on refining this in the third sprint of the workshop.

The group will now focus on each goal and suggest the type of resources that may be needed. For example, in the barbarian protection scenario, a goal might be that the solution needs to pass the hamlet's building committee ordinance. The team could then identify that we need a legal resource to handle the permitting. Other examples from above:

- Click-through traffic on website increases by X
 - We need an Search Engine Optimization resource
 - We need a UX designer
 - We need a copywriter
- Customer services ending up with live operator are reduced
 - We need an Interactive Voice Response programmer
 - We need a customer service expert
- Barbarian incursions completely end
 - We need a medieval siege consultant

- We need serfs to help build walls
- Able to destroy Illuminati robotic soldiers without detection
 - We need a mole in the organization
 - We need a robotics whiz
 - We need an AI bio-machine hacker

Once this is done, the Executive and the team have a list of all the ingredients needed to accomplish the goals. We will use this work in the next sprint.

Sprint Two

FOCUS: Asking Agile Questions

TIME: 75 min

YOU WILL NEED: Whiteboard, Paper Easel, Marker, and Management Team and Devs

Preparation

You should have decided on Mission, Goals, and Roles (MGR) from the previous workshop. Now that we have a mission and some goals to work with, as an Executive you will want to check in on the team and get various indications of Status. This is accomplished by asking “Agile” questions. You may also be asked to respond to challenges the team might face and help solve problems that the team cannot solve on their own.

The questions you ask self-directed teams are different from the questions you would ask a waterfall team, and the way you respond to challenges is likewise not the same as it might be for a waterfall project. The questions you ask and the positions you take should leave the team feeling empowered and trusted to get the job done. They should also sense that the right information is being used to make possibly difficult decisions which are meant to keep the project on track.

The basics of what you need to know as an Executive do not really change by using Agile, but the manner in which you glean this information must be adjusted if you want to achieve the highest levels of success and adoption. In this workshop you will learn the best ways to get and respond to the essential units of information you need as an Agile Executive.

Essential Information

The essential measures for Agile projects you need to understand and respond to are:

- Quality
- Risk
- Trust
- Precision

- Health of the team
- Focus
- Prioritization
- Maturity
- Team Improvement

This is different than the quite short list of the iron triangle of waterfall inquiry:

- Cost
- Scope
- Time

Using the project selected in Sprint One, you and your team will draw from what you learned in the previous lessons to develop a list of questions and positions that align with the Agile approach to project inquiry.

Break the plenary into groups of 4-10. The groups can (but don't have to) be the same groups used in the MGR Sprint, but each group must select a mission (and its goals) developed in the first sprint.

One member of the group will play the role of a member of the executive team, who has a stake in the project. The group will work together to make a list of 10 questions that will give them the best idea of the status of the project.

What You Are Looking For

These are "Agile" questions. These types of questions lead to actionable responses, which have an overall positive effect on project success. Agile questions can be about:

1. Defect backlog (fixing defects in the system)
2. Team failures (learning from recent failures)
3. Collaborations (working together to be more effective)
4. Customer interaction (listening and talking with customers)
5. Points Velocity (observing the effort expended on user stories)

6. Burndown (observing the total progress through a sprint)
7. Scope adjustments (examining whether project specs need to change)
8. Prioritization and trade-offs (discovering where the ultimate focus should be)
9. Team balance and confidence (listening to your team to see if they need help)
10. Retrospective-inspired improvements (discovering new directions based on feedback)

What You May Get

As your groups attempt to come up with questions that give insight into project status, you may hear “waterfall questions” which yield little or no actionable responses. This type of question leaves no room for the executive or team to make a direct change to improve the chance of success. For example:

1. Are we done yet?
2. Is everybody working hard enough?
3. What can we do to make things go faster?
4. Will we have to move the date?
5. What is the percentage of completion thus far against estimates?
6. Is there scope creep?
7. How much money is left in the budget?

Once each team has a pretty good list, ask them to align each of their questions to one or more essential measures in the Agile list above. It should be noted that “iron triangle” questions do not fit the Agile measures very well. For instance, “Are we done yet?” does not yield an answer that provides an actionable response.

Positions: What are we going to do about it?

If everything is awesome and cooking really well, then the correct position is: “Carry on,” and “Thanks.” However, problems and challenges are in the destiny of most projects. When something goes wrong, you as the Executive are expected to decide what should be done about it. You will take a position -- even if it is to do nothing -- and your position will be felt and noted by the team.

We will now work on recognizing the difference between Agile and Non-Agile positions. A position is the response from an Executive or manager after they have been given some information about the project.

For this segment the full group can assemble together, or small groups can stay in their work teams. The facilitator will indicate to the group that there is a problem with the project, which must be solved with input from the executive team. For example, the problem could be:

- We are running out of time to achieve the scope
- The scope has recently expanded
- We lost a resource that was needed to complete the project

The facilitator should form two questions to ask the group about the difficulties with the project. One of the questions should be in the style of an Agile project, and one of the questions should be in the style of a waterfall project. For example:

- Has the team has failed recently? (Agile)
- What is the percentage of completion thus far against estimates? (Waterfall)

Instruct the group to pretend that they must report unfavorable circumstances in answering the questions. Then they must explain how they plan to remedy the issue.

On the whiteboard, the facilitator will write both questions side by side, with the unfavorable answers underneath. For example:

- What is the percentage of completion thus far against estimates?
 - “Well, we are about two weeks behind schedule, but we think we can cut out some testing time at the end so we can utilize some of that for development.”
- Has the team failed recently?
 - “Yes, our last two deployments took a long time and we found a number of bugs in production that we did not find in lower environments. We plan to increase review unit testing scripts.”

The facilitator then initiates a conversation with the group about each question style, asking the group which questions lead to actionable answers with supportive outcomes.

Participants are then asked to voice their conclusions based on the exercise, and the facilitator reinforces the idea that Agile questions lead to the support of a team in finding actionable answers that lead to visible improvements in the project status.

Sprint Three

FOCUS: Servant Leadership

TIME: 75 min

YOU WILL NEED: Whiteboard, Paper Easel, Marker, Management Team and Devs

Preparation

In Agile, the Executive mindset must be adjusted from “command and control” to “inspire and support”. The following workshop gives participants a chance to understand these skills, which can only be fully developed over time. While completing this workshop will not instantly transform the team, it will provide a head start for solving problems and setting the groundwork for success.

As the Executive, it is important that you model the behaviors you want the managers and others on the project to display. This workshop will explore the qualities of Servant Leadership and invite participants to identify behaviors that display each quality.

As with any term, there are many opinions on the core qualities of a servant leader. The “Greenleaf” definition is the most common and lists 10 qualities.

1. Listening (What do you think?)
2. Empathy (How do you feel?)
3. Healing (How can I help?)
4. Awareness (I notice that you look happy/sad.)
5. Persuasion (Sell, don't tell.)
6. Conceptualization (In a perfect world, how could it be done?)
7. Foresight (Have a Plan B and Plan C.)
8. Stewardship (How can I make things better?)
9. Commitment to the growth of people (Let others lead.)
10. Building community (We are all in this together.)

Identifying Servant Leadership

This workshop has a simple structure. The facilitator begins by describing servant leadership:

“Servant leadership is a management style where the leader’s primary role is to give team members the tools they need and to remove impediments, then trust the group to use those tools to succeed.”

The group is separated into groups of 5-10, each gathered around a flip chart. The facilitator shares the Greenleaf list, which will give participants some general ideas of servant leadership methods. The facilitator then instructs the groups to list specific examples of servant leadership on their boards, using the Greenleaf list as a guide.

For example:

“People are allowed to talk and give feedback during meetings” might be a real-world example of the concept of “Listening”.

“Management asks how subordinates are feeling about their workload” might be a real-world example of the concept of “Healing”.

“Management is willing to take blame for poor decisions” might be a real-world example of the concept of “Stewardship”.

Give the group 15-20 minutes to make a nice list. Then ask each group to report on their list and explain their examples. Discuss any similarities or differences between the lists and the Greenleaf standards of Servant Leadership.

Next, have the groups cross off items on their list that are lacking in their current work environment. (10 - 15 min). Then each group reports why they crossed off some items and left others.

The facilitator now addresses the entire group:

You can see the recipe for servant leadership and you have some of the ingredients and not others. What can you do as an Executive/Manager/Lead Programmer/etc. to make your recipe better?

Give the group some time to speak up and tell you. This exercise will start the wheels of change turning in your organization.

CONGRATULATIONS, YOU'VE COMPLETED LESSON 3!

Lesson 4: Agile Procurement, Communication, and Modernization

GOAL: Understand recent trends for procuring agile services in government, measure and improve the modernity of your organization, and facilitate productive communication.

This lesson contains resources that will help you understand the lay of the land, so you will be armed to intelligently discuss the latest trends computer technology and software methodology with other executives, program managers, technology officers and contract officers. You will also understand how you, as an Executive, can facilitate the success of your teams and get more done by asking Agile questions and focusing on communication.

Reading List

Study each of these resources and discuss what you're learning with a colleague.

[The encasement strategy: on legacy systems and the importance of APIs](#)

A discussion of rewriting legacy software systems -- and why you should.

Study Time: 20 minutes.

[Digital Services Playbook](#)

Explore the advice from USDS about the key "plays" to providing excellent digital services.

Study Time: 45 minutes

[Announcing the Agile BPA awards: A conversation about the process](#)

18F leaders show how they worked with the Federal Acquisition Service and agile vendors to pioneer the process of an Agile Blanket Purchase Agreement.

Study Time: 15 minutes

[How to use more open source in your next federal IT acquisition](#)

Open source software provides huge opportunities for anyone building an IT project. Discover how to use it securely and effectively.

Study Time: 45 minutes

[The People's Code](#)

The White House released the Federal Source Code Policy to make it easier for agencies to access and use custom software code built for government.

Study Time: 15 minutes

Video List

[AGL Live: Agile Government with USCIS CIO Mark Schwartz](#)

Study Time: 1 hour

Agile Procurement

The Executive has a unique power and responsibility with respect to initiating and supporting Agile technologies within government at the beginning and end of the software lifecycle. The Executive can demand that the procurement process -- the genesis of much government software -- support Agile. This will normally have to be accomplished within a regulatory framework and with procurement professionals. In the Federal context, these are the Federal Acquisition Regulations and contract officers.

The Agile Manifesto insists on **customer collaboration over contract negotiation**. This is directly opposed to the traditional "Request for Proposal" process, which one attempts to precisely define what is needed and then perform competitive bidding to get the best value for the taxpayer. This basic approach grew out of civil engineering, and it works well for bridges and roads. It is anathema to good software development.

The Agile Executive must therefore change the way they think about procurement, and lead the organization into doing so as well. The fundamental concept has been articulated by Mark Schwartz: "But functional teams, not software."

You have to change the mental script. Here are some examples:

Instead of....	Think....
Contracting for working software	Contracting for software creation capability
Contracting to hit a deadline	Contracting for progress
Writing a detailed RFP	Insisting upon ongoing contractor/user interaction
We want a functional system years from now	We want something testable and improvable right away
We want a list of requirements	We want a prototype that we can check with users
We want to evaluate how well firms can write an answer to an RFP	We want to evaluate how well a firm can produce a prototype in a limited time

In practice this means that “Firm Fixed Price” as a contract mechanism must simply be abandoned, despite the persistent and erroneous belief that it reduces risk. Consider using a meritocratic competence process as pioneered by 18F’s [Agile BPA](#).

If one is not attaching a financial incentive to the completion of a large piece of software, one must find alternative approaches to ensuring good work from contractors. We recommend providing financial incentives on a periodic basis based on performance. However, this may be politically uncomfortable and regulatorily tricky. Mark Schwartz uses a simple technique: use more than one firm, and provide more work to the firm(s) that seem to be performing better, and less work to the firm(s) that are performing less well.

You may have to ask your procurement office to step way outside their comfort zone and traditional way of doing things to accomplish this. It is your job to make them do that, and it is their job to do it, so be prepared to break some eggs to make the omelette. If you allow hide-bound tradition in procurement to force you into detailed RFPs and financial incentive to firms for adherence to contracts rather than listening to the user and really delivering value, you have lost the game on the first move.

Using Agile in a Legacy Environment

Unfortunately, most government executives are placed in a legacy environment that is not of their own making. This is true both of the procurement practices and the actual existing software.

Rewriting a legacy system is one of the toughest and most common jobs facing government executives. The absolute key to [rewriting a legacy](#) system is to find a way to do it one piece at a time. This has been articulated by Martin Fowler as the “[Strangler Pattern](#)”. The basic idea is to build Application Programming Interfaces (APIs) which encase particular components. What is encased by an API can be safely rewritten with minimal risk.

The Strangler Pattern and APIs are intimately connected with automated testing. Your engineers should understand this and be able to explain it to you. Your job is to insist on a culture of automated testing as a risk-mitigation approach. This will decrease the chance that you have an unpleasant experience testifying to Congress.

You may find engineers who tell you that a system cannot be rewritten using the Strangler Pattern. If they tell you this, tell them to think harder about it. If they continue to insist that it can't be done, demand an outside opinion and a cogent explanation for why it can't be done. If your engineers insist on a position that they can't articulate clearly enough for you and others to understand, then they are not doing their job and you have to fire them. Only one in hundred legacy systems cannot be rewritten with the Strangler Pattern, and all of them involve some sort of weird hardware integrated with the software system.

A Modernity Checklist for 2017

The landscape of computer technology and even software methodology evolves constantly. Like any subculture, software has its buzzwords, hype, and fads. An Executive need not invest heavily in every new concept, but should at least be conversant with the terminology in order to have intelligent conversations.

The following checklist for 2017 represents practices and techniques which we believe have stood the test of time. These should be considered for adoption by every modern government software organization. As an exercise, evaluate your own organization against this checklist. To make it more fun, ask a peer in a different organization to have lunch with you, and challenge them to complete the checklist for their own organization as you complete yours and compare notes.

2017 Government Modernity	
<input checked="" type="checkbox"/>	Evaluate yourself against this checklist
<input type="checkbox"/>	My organization uses The USDS Checklist: https://playbook.cio.gov/
<input type="checkbox"/>	My Engineering Team uses Spikes and has demoed one to me recently
<input type="checkbox"/>	My Engineering Team uses Rapid Prototyping regularly
<input type="checkbox"/>	My Engineering Team is familiar with GitHub
<input type="checkbox"/>	My Engineering team looks for Open Source solutions as a matter of course
<input type="checkbox"/>	We use virtual containers easily and appropriately
<input type="checkbox"/>	We have dedicated DevOps functions and personnel
<input type="checkbox"/>	My Engineering team builds automatic test suites
<input type="checkbox"/>	We use automated security scanning consistently and periodically
<input type="checkbox"/>	[EXTRA CREDIT] We have read about 18F's Meritocratic Competency Bid Evaluation and are considering using something similar

<input type="checkbox"/>	[EXTRA CREDIT] We have used the Wardley-Duncan Map to analyze our legacy systems
--------------------------	--

Exercises

1. Plan a one hour meeting with one or two of your best engineers to discuss the Strangler Pattern and how it applies to your specific legacy systems. If you have not legacy systems, use this meeting to discuss your automated testing strategy.
2. Plan a two hour meeting with your best engineer and a procurement professional. Take an RFP which has not yet been executed and discuss how it can be changed to make it Agile by contracting for software production capability rather than using detailed requirements. If you have no RFP in process, use dig up the last one your organization used and use that.
3. Evaluate your organization against our modernity checklist.
4. Plan a one hour meeting to evaluate one of your projects against the USDS playbook checklist.
5. Deputize someone to understand the [Wardley-Duncan Map](#) . Have a one hour meeting with them where you begin to build a Map of one of your legacy systems. Have them go away and try to “make a move” as explained [here](#) and present it to you later. Decide if it is worth sharing with your entire organization.
6. Invite a peer in a different organization to lunch and challenge them to fill our our “2017 Modernity Checklist”.

CONGRATULATIONS, YOU’VE COMPLETED LESSON 4!

Conclusion

If you completed all the exercises and materials in this course, you are familiar with all the concepts you need as an Agile Executive.

Once your agency begins to reap the benefits of Agile, you'll likely want to continue studying and pursuing this methodology that allows governments to bring delightful, effective services to citizens and customers.

This course is still evolving. Please [contact us](#) to provide feedback!