Day 2: Questions from Day 1?



Loose Workshop Outline – Day 2 – Agile/Scrum

- 8:30 AM Kick off
 - Review/questions from Day 1
- 9:00 AM Agile/Fundamentals
 - Quecreek Rescue
 - Agile Benefits
 - **Exercise:** How Agile is ETS Today?
- 9:30 AM Agile Long Term Planning
 - Agile Roles
 - Overarching Process Flow Review
 - Review of Office Pro Plus Business Case
 - Exercise: Elevator Statements
 - Exercise: The Focus Matrix
- 10:00 AM Break
- 10:15 AM Finding Project Scope
 - Finding scope with story mapping
 - Simple user stories
 - Prioritizing Scope
 - Estimation and Release Planning
- Noon: Lunch

PICKOUP

- 12:30 PM Delivering with Scrum
 - Detailed user stories
 - Sprint Planning
 - Refinement
 - Daily Standup
- 1:45 PM Break
 - Demos
 - Retrospective
 - Plan Adjustments
- 2:50 PM Final Questions
- 3:00 PM Workshop Ends

What ls Agile?



The Story of Quecreek





The Shift Starts



The Tragedy Begins - Wednesday



Looking For a Sign - Thursday



The Big Drill Arrives



The Last Man Out





Parallels between Quecreek and Projects:

Clear Priorities

- Quecreek had a major time constraint *Get the miners out before they died*
- Most software projects have to be delivered by a deadline or their value is marginalized
- The analogy: "What is the value of a Sunday newspaper on Monday?"





Parallels between Quecreek and Software Projects:

Expect change and be ready to adapt

The Quecreek rescuers adapted to:

- Broken drill bits
- Gas lines blocking their drill path
- The need to reduce the time required to manufacture a fishing device

In projects we encounter:

- A missing requirement
- A technical constraint that invalidates your design
- A department or vendor delivering their part of the project later than expected



Parallels between Quecreek and Projects:

Collaboration

Ideas came from all Quecreek team members

- Idea Drill a pilot hole first
- Idea Use positive air pressure to keep the water at bay

On projects we want:

- Ideas from all team members
- To avoid a contract/adversary relationship with vendors or the customer







Parallels between Quecreek and Projects:

Early Feedback

At Quecreek the rescue team thought they knew where the miners were

• A quick pilot hole verified the assumption

On projects we:

• Demonstrate frequently to verify we are on target with the customer needs





Agile is What We Just Discussed

- 1. Synchronizing everyone on the main goal
- 2. Harnessing the knowledge of the team
- 3. Expecting and adapting to changing needs
- 4. Establishing a collaborative environment
- 5. Minimizing risk through frequent feedback
- 6. Removing all forms of waste from the process
- 7. Delivering while the need still exists
- 8. Planning at a level that correlates to the information available
- 9. Delivering extensible projects

10.Emphasizing *value delivery* over following a plan

Would any of these not make sense within ETS?

Agile Benefits - VersionOne 2020 Survey



Roles





Agile Teams

- Cross-functional all the skills needed to deliver what the customer needs
- Work together to deliver value to the customer, may include helping outside of their functional area
- In a perfect world self organizing
- In a perfect world all have good soft skills and can resolve conflict
- Make sure their work and project status are 100% transparent
- Expose and work risks as soon as possible
- Seek feedback from the customer throughout the project
- Plan at the beginning but realize re-planning after discoveries is even more important
- Coordinate their work with support teams such as load testing, dev/ops, security, help desk, regulatory, and localization
- Often use collaborative techniques to reach consensus
- Provide high level estimation to create an early project plan and detailed estimation to deliver a sprint
- Take ownership for the project



Roles Related to Requirements and Needs

The Business Owner

- Involved in long term strategic planning
- Often has a product roadmap for the next 12 to 24 months
- Creates an initial business case for a *program* or high level initiative
- Programs and initiatives usually tie to one to many projects
- Common titles: Director or VP of Marketing, Business Area Director

Product Owner

- Defines the features and stories of the product, provides input on release plan, owns and maintains the product backlog
- Aggregates input from users, stakeholders and other interested parties Prioritizes features according to business value
- Defines the acceptance criteria and conditions of satisfaction for the Sprint and the project backlog
- Takes feedback from users and stakeholders in the demo and makes a business decision on which are added to the feature list and which are set for the next Sprint
- Common titles: Product Manager, Program Manager, Business Analyst

Overview of The Process for an Agile Project

1) Establish the project team

- ${\mbox{>}}$ Get the best possible person for the Product Owner Role
- > Do a RACI Chart to make roles clear

Busines	IST OTO	a Des		50		
Task	Address and Address	C. Owne	loome	2	SENA ST	er
Organize pre-work planning meeting	С	С	С			R/A
Review stories that will be pre-worked with P.O.	R	R				А
Discuss the sequence the stories will be worked in	R	R				А
Determine the artifacts needed for each story	A/R	R		с	С	
Layout a schedule of pre-work milestones/deliverables	R	R				А
Complete mockups	С	A/R				
R = Responsible (Doer)						
A = Accountable (responsible for making sure it gets done)						
C = Consulted						
I = Informed						

2) Synchronize the P.O. and Team on the Vision

- > Create an Elevator Statement Together
- For: Video Game lovers
- Who: desire a simple way to play and enjoy video games
- The: Xbox 360 Kinect
- Is a: video game platform
- That: provides a wide variety of games for gamers and families
- Unlike: the Nintendo Wiii

And

- Unlike: the Sony PlayStation
- Our: product does not require a controller and allows a person to interact with a game through natural gestures and spoken commands

> Do a Focus Matrix to make priorities clear

	Fixed (1)	Emphasized (1)	Flexible (2 to Infinity)
Cost			х
Schedule		х	
Scope	x		
Quality			х

3) Find or review the requirements

> Story Mapping is a great way to find potential scope



4) Prioritize the requirements (user stories)

> Team helps, PO makes final call> Start with Minimum Viable Product (MVP)



> Do a second sort by technical dependencies and risk

5) Establish the project foundation

> Get environments up, complete contracts, set-up tracking/reporting> Determine sprint length and structure

	Day 1	Day Z	Day 3	Day 4	Day 5
Week 1: > Sprint Planning > Start Sprint	Sprint Flanning Crimiticans) Product Owner residence stories with team	Sprint Planning (2.16-4 hours) • Deen defines and estimate tasks • Chalte Acceptance Criterio • Deencestre baild dequeste für sprint • Trade Commits!	Balld/Inst/Accent Write test cases Execute tests Accept Scores Daily Standup	Build/Inst/Accent Write test coses Execute tests Accept Stortes Daily Standup	Bobb/Text/Access Write test cases Execute tests Accept Stories Daily Standup
Week 2: > Build > Test > Accept	Build/Test/Accept Write text cases Execute texts Accept Stories Daily Standup	Balls/Best/Accept Write test cases Exercise tests Accept Stories Dally Standup	Build/Test/Accept Write text cases Execute texts Accept Stories Daily Standup	Build/Test/Accept Write text cases Consulte texts Accept Stories Daily Stanckup	Build/Text/Asseut Write Text cases Execute texts Accept Stones Dairy Standup
Week 3: > Build > Test > Accept	Build/host/Assent Write test cases Encode tests Accept Stories Daily Standup	Baild/bail/bail Write test cases Enecute tests Accept Stories Daily Standop	Baild/Bail/Room Write test cases Encode tests Annept Stories Daily Standup	Bahl/Inst/Accest Write test cases Execute tests Accept Mories Only Stendup	Excht/Inst/Assest Standage Grooming Meeting for Next Sprint
Week 3: > Finalize > Demo > Learn	Beiti/hist/Assent Write test cases Execute tests Assent Stanias Daily Standup	Harden, no new stories to OA Daily Standop Code Review	Hardon, no new stornes to OA Daily Standup	Sorint Wrap-Up (2 to 4 hours) Review velocity SWAD Stories for next sprint	Sprint Wrap-Up (2 to 4 hours) Review and Denso Update regression tests spras release part

6) Estimate the requirements and load into sprints

- > Do relative estimation
- > P.O. describes, team estimates
- > This plan is for strategic purposes it is **not** a team commitment!

Priority	User Story Name	Story Points				
Critical	Simple Search	2 - Small				
Critical	Post item up for bid	3 - Medium				
Critical	Bid on an auction	8 - Extra Large				
Critical	Register on site	5 - Large		6		
High	Flag problem postings	3 - Medium	Carlink 1	Constant 2	Furint 2	Carried A
High	Contact the seller	8 - Extra Large	2 - Simple Search	4 - Flag problem postings	8 - Advanced search	9 - Auction browser toolba
High	Item alerts	1 - Extra Small	4 - Post item up for bid	8 - Contact the seller	1 - Email to a friend	16 - Retract a bid
High	Online help	2 - Small	8 - Bid on an auction	2 - Item alerts	16 - Customize my view	
Madium	Deserved seller feedback	2 Madium	8 - Register on site	2 - Online help		
meaium	Record seller reedback	3 - Medium		4 - Record seller feedback		
Medium	Review seller feedback	3 - Medium		4 - Review seller feedback		
	A discovered as a surface	E Losso	22	24	25	24

7) Plan a sprint – pull stories from top of backlog

- > Some pre-work may happen before it gets to planning
- > Stories get detailed design even screens
- > Detailed acceptance criteria



> We leave story point behind, now we only care about hours

Story - Ability to bid on an item		
HTML for screens	UI	16
Design data model	Dev	16
Code insert, update, delete	Dev	8
Interface with user registration	Dev	4
Create tests	QA	2
Error Handling	Dev	8
Ctony Ability to place on item up for bid		

> If the work fits the team commits and starts the sprint

8) Code, test, and accept a sprint

- > Team tries to complete a story at a time
- > Daily standup to allow quick triage of issues
- > Perfect world P.O. approves each story as completed



Week 1: Sprint Planning &	FOx nevers stories with Dec 8, Q5	Jess and QA Tecimate Determine Dutit	nali 1	nule Code Story 1	2010
Stort		 Englise Englise Acceptance Crise Is COMINITI 	A Write Samme	Test Case Story	1 North
Week 2: BUILD	Text Sto	71 /	Accept Story 1	The tex cises	Sold see. accept, stan
TEST	Ca	ide Story 2	2	st Story 2	Greaning Abouting for
ACCEPT	Write T	est Case Story 2	2	Standup	Read Sprint
Week 3 Finalize	Accept St	ory 2 16	Flanders, no ress etertec no ct/A	Beview with Massehalders	facture and
Demo	furning texts Accept Stories			Finalize Stories for Nast Sprint	repression a
	manop			Bruine velocity	Update tels

9) Review, adapt, re-plan

- > Review with PO & Stakeholders/Review with Executives
- > Perform process retrospective
- > Adjust plan, estimate stories for next sprint, return to step 7



19

1) The Team Reviews the Business Case

Online User Story Training Application

By Lauren Smith – Product Owner

Project Concept

Here at GS Solutions Group, we do the majority of our training and coaching in-person. With recent changes in the world, we must move to a totally online coaching and training model.

To that end, the most frequent request from our clients is to understand how user stories are identified, enriched, defined, coded, tested, and accepted. Our first online workshop will be focused on user stories.

The main driver for this initiative is to address the needs of our clients, and to ensure sustainability of the company.

The online Agile training market is very crowded, with numerous options from large, Agile and Project Management focused training companies. To stand out in this market, we must stay true to our brand, and create a training module that can be customized for each audience and client we work with. Contextual training will provide more value to our clients, and allow them to apply the training as soon as they return to work.

We envision two phases of this project.

- Phase 1 Create the core functionality, aimed at a generic public audience, such as a PMI chapter.
- Phase 2 Based on the feedback from testing Phase 1 with PMI chapters, determine what attributes of the training module need to be customized, and make those areas configurable.

For today's purposes, this business case will be focused on Phase 1. We will create a subsequent business case for phase 2. In terms of constraints, we believe we need this module up and running in beta form by May 14th, 2020. We believe expenses will mainly be our time to create the app. Quality/robustness is important, but can be de-emphasized to support the target date.



We will use our current Agile methodology to create the application (foundational work plus Scrum).

Target Users

- Generic Agile team members
- Product Owners
- Business Analysts
- Scrum Masters
- Project Managers

In Scope

- · Understanding user stories versus other requirement methods
- How to evolve stories
- · Epics compared to stories
- Story estimation
- How to test stories
- Non-functional requirements
- Story hierarchy and taxonomy
- Story syntax
- Stories and governance
- Refining stories
- Registering students
- Testing Students
- Sharing test results
- Certificate of attendance
- Payment and invoicing
- Course discounts
- PDUs

Out of Scope or Phase 2

- Certification
- Workshop customization
- Self-paced training
- Mobile device support

Workshop Duration

We are envisioning a two-part workshop, with each session running 90 minutes.



Review of The Business Case for Our Simulation

Project Details: Microsoft 365 Implementation For this exercise: Project subset Office Pro Plus Implementation

Background:

Enterprise Services (DES) is undertaking an IT transformation project to transition to a Microsoft 365 (M365) cloud-based enterprise. This will be a phased project to plan, implement and roll out M365 applications and services to all DES programs.

There are many drivers for this transformation, both technical and business driven. For example, a technical driver is that WaTech has plans to decommission the Vault storage for our email (IronPort) in June, 2021, and is also going to be decommissioning SharePoint in June 2022. DES will need to move to Exchange Online and SharePoint Online to mitigate those decommission changes. Transitioning off of older technology will enable our agency to transform to a more modern, flexible workforce environment, a clear business driver now.

The transformation will focus on providing the essential components for the implementation of M365 at DES. The foundational components will provide baselines that are necessary prerequisites for successful M365 and public cloud workload migration. This project will also require strong change management to ensure successful adoption throughout DES programs.

This project will provide new solutions and enhancements of existing ones for the DES enterprise, supporting security compliance, organizational efficiencies for DES, enabling users to access business applications through a mobile platform, via cloud application and file access.

For this exercise: subset would be rollout Office pro Plus for the agency

Vision: on overall project

Implement a solution that provides overall compliance and governance for all of the cloud services running in the new M365 platform, providing a secure, compliant cloud-based environment based on industry identified best practices.

The M365 platform provides all of the necessary enterprise components to support DES end users using the new M365 platform. ETS will be the first division to pilot and undergo this transformation and migration of workload to M365.

This effort starts with implementing a solid foundational approach to reference architectures, security frameworks, and organizational effectiveness. The project team will identify processes and develop governance with DES business programs that will provide the foundation necessary to successfully deliver M365.

For this exercise

Vision: - push out the new M365 Office Pro plus to all 800 plus users in DES. Create user documentation, training materials and communicate impacts and changes. This will bring all staff up to date with the latest Microsoft Office software.

Risks - User downtime. Change in how they do things

2) The Team and P.O. Create an Elevator Statement

Define your position based on the target market and how you will provide value to capture the market. Go on to discuss your competition and what your product will do to separate itself from similar products.

For (target customer)

Who (statement of the need or opportunity)

The (product or project name) is a (category)

That (statement of key benefit – that is, compelling reason to proceed)

Unlike (primary competitive alternative)

Our Product (statement of primary differentiation)

Elevator Statements

For	Requestors (Capitol Security's customers)					
Who	Desire a more efficient (intuitive, role/identity-focused) and robust					
	(consolidated, one-stop shop) system to make and manage physical access requests to facilitate the fulfilment of their orders.					
The	Building Enterprise Access Request System (BEARS)					
	1- For (target custom	r)				
ls a	Single system to make access requests 2- Who (statement of 3- The (product or product or	the need or opportunity) ject name) is a (category)				
	4- That (statement of	(ey benefit – that is, compelling reason to proceed)				
That	Allows us to make correct orders on the first attempt 6- Our Product (state	nent of primary differentiation)				
Unlike	The current card key request system and other manual request processes					
Our						
	BEARS system will allow us to make access requests more easily and accurately, which will reduce turnaround time.					

You Can Still Keep and Use Remaining Benefitstatements

Additional/Secondary benefits

- Facilitates modern, intuitive, and efficient ordering workflow system
- Is a robust reporting solution for ordering activities
- Is a notification system for orders and key activities
- Will improve access request communication

Elevator Steps

- Identify the key audiences who get value (audience is a category of users, not specific roles)
- 2. Document the main need of the audience
- 3. What is the name of your project or product? What type (category) of project is it?

Before going to steps 4, 5, and 6 – brainstorm all possible benefits to the audience, if we do this project

When done proceed.

- 4. Put a "good" benefit here. Maybe a benefit that makes you equal to your competition
- 5. For 5 and 6, put your separator here. What makes your product better than your competitor or your current product? What will the audience be most excited about?

- 1- For (target customer)
- 2- Who (statement of the need or opportunity)
- 3- The (product or project name) is a (category)
- 4- That (statement of key benefit that is, compelling reason to proceed)
- 5- Unlike (primary competitive alternative)
- 6- Our Product (statement of primary differentiation)
 - Music lovers For: Who: desire a simple way to listen to and manage their songs The: iPod portable digital music player Is a: provides intuitive, easy to use That: controls. Unlike: other MP3 players product provides seamless Our: integration with a world class music store

(iTunes).

HCA Example

Stop: Brainstorm all benefits we can see For: HCA Executives Who: Need Quick Access to Provider Data The: "Provider Dashboard" 1) > Self - Service (SASM) > Data will be current > Better Visualization 15 A: Dashboard and modeled set of provider data > Better Manipulation > Data will be Consistent That: Reduce PDRS 2) > Reduce Public Disclosure Requests (Because it is publicly queitable) And > Reduce Human Enors on Reporting Unlike: Having to make requests today, Our project will provide real-time Self-Sorvices

Mural Exercise

For	
Who	
The	
ls a	
That	
Unlike	
Our	

Elevator Statement – Our Project Compass

- The elevator statement tells us which way to go when prioritizing
- It helps us make tradeoff decisions
- It reminds the team who we are serving and how
- It synchronizes the team on the value



3) The Focus Matrix

- Your stakeholders want everything for a project
- But if the team can only guarantee one attribute, what do the stakeholders want?
- The Focus matrix shows the relative importance of each factor for a given project, as determined by the stakeholders of the project request.
- **"Fixed"** means the factor cannot be changed. **"Emphasized"** means it is our main focus since we have some control. **"Flexible"** means the factor can be heavily adjusted.

	Fixed (1)	Emphasized (1)	Flexible (2 to Infinity)
Resources	X		
Schedule		Χ	
Scope			Χ

4) The Team and the P.O. Identify the Users (Mural, Teams, Other)



Some Synonyms for "User"

- Actor
- Role
- Persona
- Profile
- Audience
- Demographic

BEARS Profiles

Requestor (Capitol Security's customers):

- May be dedicated to facilities work only, or have tasks that are primarily unrelated to facilities
 Requestors' jobs and tasks can vary greatly
- Likely has a limited understanding of access controlspecific terms
 - Often reaches out via phone or e-mail to Capitol Security
- May requests sparingly/occasionally, or may be in charge of making large bulk requests
 - E.g., May need to coordinate group access when their program moves to a new location.
- If there's more than one requestor from a specific program/agency, may need to double-check the requests that another requestor made.

Capitol Security - General

- Performs recurring access control tasks
- Resolves simple/common customer service issues/complaints
- Escalates more complex customer service issues/complaints

Capitol Security - System Admin:

- Understands the big picture of access control and how the pieces fit together
- Fills in the program where needed
- Pulled in many directions (projects, tasks, etc.)
- Resolves more complex customer service issues/complaints

BEARS Profiles

Capitol Security - Investigator:

- Capitol Security manager
- Responsible for input into program budget
 - $\,\circ\,$ Conscious about program costs
 - Interested in auditing/investigating internal processes to gather information and validate effectiveness
- Ensures high-level processes are effective

Finance Staff:

- Ultimately receives Capitol Security's customer payments
- Supports multiple programs
- Finance experts, but likely unfamiliar with the intricacies of Capitol Security's business processes
- Occasionally needs to research specific order details
- Needs to see big-picture financial information and high-level reports

Who Would Be Our Users for Office Pro Plus? (Mural Exercise)



5) Once We Know the Users, We Need to Identify The Stories They Need



Stories are "Tasks" that the User Performs with the System

<u>As a</u> buyer <u>I want to</u> review seller feedback <u>so that I</u> can see if I trust this seller

<u>As a</u> buyer <u>I want to compare prices</u> <u>so that I</u> can make sure I am buying at the lowest possible price



I start with this simple structure, when we are just scoping:

User:

Verb/Action:

Noun:
Can We Identify Some User Stories for Office Pro Plus? (Mural Exercise)

ETS Employee Share Document I.T. Monitor Application Administrator Assign Roles

Help Desk Support Office Pro Users

6) Once We Have the Stories/Backlog, We Could Find MVP

- A Minimum Viable Product has just those features/stories that allow the product to be deployed, and no more
- I like to say that if even one MVP story is removed, we do not have a deployable product for any audience
- It is a strategy to avoiding delivering what customers do not want
- The MVP often serves as a learning experience to tell us what to add to the product next
- MVP also helps us get to market while the need still exists



Once We Identified All of the Stories, We Could Identify MVP



7) Team Estimation of Stories using Story Points

Priority	Requirements/Stories	Size			
Critical/MVP	Document Sharing	2 - Small			
Critical/MVP	Messaging	3 - Medium	T-Shirt Size	US Points	Hours
			X-Small	1	<15
Critical/MVP	Create Chat Groups	8 - Extra Large	Small	3	16 - 29
Critical/MVP	Notifications	5 - Large	Medium	5	30 - 59
			Large	8	60 - 100
High	Online Calls	3 - Medium	X-Large	13	>100
Medium	Download documents	5 - Large			

Planning Poker

- The team discusses the story with the customer
- Each member secretly chooses a card with either 1, 2, 3, 5, or 8 for size
- On command everyone expose their cards
- The differences leads to discussion
- Keep discussing and voting until the estimates are somewhat close to each other

In the New Days, An App We Can All Access



The Scrum Master could facilitate and record the agreed to estimate.



The Scrum Master Would Enter the Stories and Estimates into Azure DevOps

Priority Critical/MVP	Requirements/Stories Document Sharing	Size 2 - Small	
Critical/MVP	Messaging	3 - Medium	
Critical/MVP	Create Chat Groups	8 - Extra Large	
Critical/MVP	Notifications	5 - Large	
High	Online Calls	3 - Medium	
Medium	Download documents	5 - Large	

Another Option – Affinity Estimation



A process where team members put the product backlog in sequence, from smallest to largest.



Sprint 1	Sprint 2	Sprint 3	Sprint 4
1 – MSN Governance	5 – Module testing	8 – Advanced search	8 – View average grade
3 – ACM Governance	8 – Contact instructor	1 – Email to a friend	13 – Workshop refund
8 – Story testing	2 – Online Help	13 – Customize my view	3 – Venmo payment
8 – Simple Search	5 – Provide app feedback	2 – Pause training session	
3 – Contact student	5 – Review app feedback		
24	25	24	24

We will assume the development team has averaged 25 points per sprint in the past (velocity).



Long Term Planning Review

- Envisioning gives your release or project a strong foundation, with common understanding. It can still be performed remotely.
- Elevator statements can still be created collaboratively.
- We can still perform Story Mapping and find MVP collaboratively, when we are remote.
- Planning Poker is still valuable, and still easy to do remotely
- Almost all Agile/Scrum tools will support creating releases and defining MVP
- This is a good time to initialize a light charter





Break





Agile at the Team Level – Via Scrum





Simplified Scrum Overview

- Sprints time boxes
- 3 kinds of Artifacts
 - Product backlog list (a list of user stories)
 - Sprint backlog list (a list of tasks)
 - Sprint burn down chart (graph showing progress and how much remains to be done)
- 5 kinds of Meetings
 - Sprint planning meetings
 - Daily scrum (and scrum-of-scrums for larger projects)
 - Sprint refinement
 - Sprint review
 - Sprint Retrospective
- 3 Roles
 - Product Owner
 - Scrum Master
 - Team (development)



Sprint Structure Example

	Monday	Tuesday	Wednesday	Thursday	Friday
 Week 1: Sprint Planning Start Sprint 	8:30 AM Sprint Planning Big Picture Review Sprint Goals Planning Commit or Adjust Sprint Begins	Build/Test/AcceptWrite test casesWrite CodeAccept StoriesDaily StandupCode Reviews/RefactorRefine Future Stories	Build/Test/AcceptWrite test casesWrite CodeAccept StoriesDaily StandupCode Reviews/RefactorRefine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories Team Refinement Meeting 9 AM	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories
Week 2: Build Test Accept Complete	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories Team Refinement Meeting 9 AM	Sprint Wrap-Up Stabilize code Finish off bugs Complete documentation Review Definition of Done 1:00 PM: Demo/Review 2:00 PM Retrospective



What are Sprints? (Mini-projects)





Sprints

- The software could be custom code, or a configured SaaS such as Salesforce
- A team is **dedicated** to a sprint. They all work together to deliver the software by the end of the sprint.
- A customer representative, called a product owner, interacts with the team during the sprint to answer questions and to clarify requirements as needed.
- Scrum can be used for non-tech projects

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1: > Sprint Planning > Start Sprint	8:30 AM Sprint Planning Big Picture Review Sprint Goals Planning Commit or Adjust Sprint Begins	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories
Week 2: > Build > Test > Accept > Complete	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories	Build/Test/Accept Write test cases Write Code Accept Stories Daily Standup Code Reviews/Refactor Refine Future Stories Team Refinement Meeting 9 AM	Sprint Wrap-Up Stabilize code Finish off bugs Complete documentation Review Definition of Done

Sprints are typically 1 to 4 weeks long, but I have seen them go as long as 8 weeks and still be beneficial.



Why Sprints?

- We can delivering value to the customer throughout a project, as opposed to just the end
- We do a better job of delivering accurately due to a shorter gap between analysis, coding, and testing
- We have a logical place to adapt to discoveries throughout a project (at the end of a sprint)
- We have a great tool for measuring the most important project metric – how much code is truly complete (production ready)?





The Definition of Done for a Sprint

- Code meets the Story Acceptance Criteria
- Code completed, unit tested (when possible), checked in and run against current version in source control
- Code is peer reviewed and meets development standards
- Builds without errors
- □ No priority 1 or priority 2 bugs are open
- QA validates that the functionality has been met against the stated acceptance criteria.
- □ Story and tasks are marked as 'complete' in JIRA
- Technical Documents Complete (When Necessary)
- Release Steps documented (When Necessary)



Sprint Planning

The Goals: Understand the scope of work for a sprint, so that it can be the planned with high confidence. Related:

- Understand the work well enough to know if it fits
- Understand the work well enough so that each team member can make the call on what they can commit to
- Have a general feel for handoffs during the sprint
- Focus on completing and not needing stories to rollover

Rating	Practice
Ideal	Team discusses stories with Product
	Owner/BA, and tasks out the work in Azure
	DevOps
Not	Team discusses stories with Product
Advised	Owner/BA, and they tell the Scrum Master
	what to put into Azure DevOps
Ideal	Identify target dates within the sprint, for
	each story to be complete.
Ideal	Manage work in progress, focus on
	completing stories one at a time. Limit how
	much work is in progress.





A Story After Sprint Planning

Product Backlog

Epic	→ Story	Acceptance Scenarios	Build Tasks
Registration	Title: Create Account Detailed Story: As a student, I want create an account, so that I can register for the user story workshop 5 Story Points (estimated during grooming)	Scenario 1 – User name does not exist: Given the student enters a unique user name When they submit their account request Then create a new account And Then say "Account Successfully Created" And Then email a confirmation to student's email address Scenario 2 – User name does exist: Given the student enters a user name that already exists When they submit their account request Then say "Account Name Already Exists, Please Try Again" And Then place the cursor in the account name field	 Create account screen – 10 hr Write query logic – 14 hr. Write test case – 2 hr. Execute Test Case – 1 hr. Create Error Handling – 6 hr. (estimated and defined during sprint planning)

SOLUTIONS

<u>As a</u> buyer (who)

I want to review seller feedback (what)

so that I can see if I trust this seller (why?)



Other Artifacts to Enrich

Screen Designs



Acceptance Criteria:

Scenario 1 – User name does not exist: Given the student enters a unique user name When they submit their account request Then create a new account And Then say "Account Successfully Created" And Then email a confirmation to student's email address

Scenario 2 – User name does exist:

Given the student enters a user name that already exists When they submit their account request Then say "Account Name Already Exists, Please Try Again" And Then place the cursor in the account name field



You can still use tools like:

- Figma
- Sketch
- Mira
- Mural



Some Teams Do Tasks and Hours During Sprint Planning, and Leave Story Points Behind

Defining the Work Needed to Build the Stories

Output: The team creates a sprint backlog. (stories and the tasks needed to complete them)

- □ The team identifies the tasks needed to complete each story.
- □ We frequently have tasks in these areas:
 - Research/Requirements
 - Design
 - Coding
 - Service Creation
 - Peer Review (PR)
 - Testing (QA or UAT)

Azure should show the tasks and their estimates, tied to a story.

Add an information form

- When the team has identified all of the work, they should discuss the total amount of work by role, and see if there is enough capacity to complete the work. If there is not, the team may split a story, or return one to the Product Backlog.
- If only one role has extra capacity, the P.O. may choose to have that role work on fixes, enhancements, or do research on future stories.

Taskboard Ba	acklog Capacity	+ New	v Work Item	ሷ Sprint 1 😅
🛨 🖃 Order	Assigned To	Remaining	Title	
+ 1	Jamal Hartnett 🛛 🚥	8	🗸 🛄 GSP locator interface	Drag and drop wor
	Raisa Pokrovskaya	8	훋 Initial design	balance work acros
2	Jamal Hartnett		🛄 Check service status	
3	Christie Church	12	🗸 🛄 Hello World Web Site	Work
	Christie Church	12	휟 Design welcome screen	leam
4	Raisa Pokrovskaya	30	🗸 🛄 Cancel order form	(97 of 117 h)
	Christie Church	12	💈 Research slow response ti	Work By: Activit
	Jamal Hartnett	18	🖻 Auto-save	Design
5	Jamal Hartnett	16	~ 📕 Request support	(24 of 27 h)
	Jamal Hartnett	16	🖻 Develop form	Development
6	Raisa Pokrovskaya	18	v 🛄 Cancel order form	(73 of 90 h)
	Johnnie McLeod	8	💈 Auto-complete user's na	Work By: Assign
	Raisa Pokrovskaya	10	💈 initial work	🜖 Christie Chur
7	Raisa Pokrovskaya	13	🗸 🛄 Phone sign in	(24 - 6 27 h)
	Raisa Pokrovskaya	13	evelopment work	(24 of 27 h)
				lohnnie Mcl



57

ŝ

 $\overline{}$

If It Fits - Commit

Determining if the Work Fits

Output: The team commits to the sprint.

 A good last step is to sequence the work across the sprint, and verify the work fits when team members are available.

	Day 1	Day 2	Day 3	Day 4	Day 5
Week 1:	Sciencelanairez Frantistificaries Interne Antonific	anteloning of el Carero Codesi		Lair Sury?	kilentissen Kinninges
> Sprint	Two .	Write Feel Co	ar 91 1	Martin Tree Case Ste	**** Set
Planning > Start Sprint	Nacio Sergiale Constitució e Sectorials regulacionals Text coverso	P#/4+6,	N#95+47	Terr Skery:)	Second Stary 2
Week 2: > Auld > test > Accept > Complete	Call Martin Ca Martin Ca Der y Ster Str	Anny X Anny X Anny X Anny A Cally State &	Rad ATTery Arrest Magazine court Metal Cole Trans Linkal Auto do New Manual Solary A	Deploy Deploy to two proclassed encaptionarts	Carbol Wag Uni C 2 Ser Examp Paratolation Carbon Archy 2 WE States Archy 2 WE States Archy 2 WE States Archive
		Next Spr	ht Pre-Work		
	and the second second				

□ The sequence sheet may show when we are targeting mid-sprint demos, and also show key milestones for the sprint.

111

- The last step is for the team to vote on whether they can commit to the sprint.
 Scope may be reduced to get consensus support to go forward.
- □ *Fist of five* may be used for the consensus vote.
- A sprint planning retrospective is held at the end of sprint planning.

(review the exit questions on page 1, at the end of sprint planning)

Taskboard	Backlog Capacity	+ Ne	ew Work Item	ሷ Sprint 1 😤 🍸 🍪 🖉
+ - Orde	er Assigned To	Remaining	Title	
+ 1	Jamal Hartnett 🛛 🚥	8	🗸 🛄 GSP locator interface	Drag and drop work items to
	Raisa Pokrovskaya	8	🖻 Initial design	balance work across your team.
2	Jamal Hartnett		🛄 Check service status	
3	Christie Church	12	🗸 🛄 Hello World Web Site	Work *
	Christie Church	12	휟 Design welcome screen	
4	Raisa Pokrovskaya	30	🗸 🛄 Cancel order form	(97 of 117 h)
	Christie Church	12	휟 Research slow response ti	Work By: Activity -
	Jamal Hartnett	18	휟 Auto-save	Design
5	Jamal Hartnett	16	🗸 🛄 Request support	(24 of 27 h)
	Jamal Hartnett	16	훋 Develop form	Development
6	Raisa Pokrovskaya	18	🗸 🛄 Cancel order form	(73 of 90 h)
	Johnnie McLeod	8	훋 Auto-complete user's na	Work By: Assigned To 🛛 🔻
	Raisa Pokrovskaya	10	훋 initial work	ġ Christie Church
7	Raisa Pokrovskaya	13	🗸 🛄 Phone sign in	(24 of 27 h)
	Raisa Pokrovskaya	13	훋 development work	Jamal Hartnett
				Julia Harnet
				(34 of 36 h)
				😨 Johnnie McLeod
				(8 of 18 h)

Raisa Pokrovskava

(31 of 36 h)

Advanced Teams - Layout the Sequence





Copyright Greg Smith

Daily Stand-up

The Goals: Review the status of the sprint. See if there are any risks to delivery. Work the risks and increase the chances of sprint delivery. Discuss discoveries and determine whether there is need to add new stories to the Product backlog.

Rating	Practice
Ideal	15 meeting minute at a physical wall
Acceptable	15 meeting minute while looking at Azure DevOps together
Not Advised	A meeting focused on individuals reporting on personal status, with
	disregard for the health of the entire sprint
Not Advised	Working issues during the standup after 15 minutes have passed.
	Instead have folks work them post meeting.
Ideal	When stories have target completion dates, review how we are
	doing to the dates.





Daily Standup

Meeting Goals

This meeting is held daily and does not exceed 15 minutes.

Attendees: Scrum Master, Project Team

Each member of the team should explain their work by answering 3 questions:

- 1. What Story was worked on and what task was accomplished yesterday?
- 2. What is planned for today?
- 3. Do you have any obstacles or impediments?

Example 1: I worked on Story 1.1 and finished the testing. I moved it to "done"

Example 2: I started to work on Story 4.3 but could not finish as the template does not match the page. I tried to follow up with the UX designer but they are on vacation. In the meantime, I'll pick up the development task on Story 3.2 and get that started.

Note: If a team member needs a follow up discussion they will ask their teammate to have a conversation immediately after the Standup.



63



Tracking During the Sprint- The Daily Standup

⊅, Fabrikam Fiber Team ∨	r 🛨 x ^R	June 11 - June 29 10 work days remaining
Taskboard Backlog Capacity	$+$ New Work Item \smallsetminus	🗘 Sprint 1 😤 🍸 🚳 🖊
	To do	In progress
Hello Work Web Site	1 not started	
Slow response on welcome page	ge	
Add an information form		
Change the initial view		
Secure sign-in		
363 Welcome back Jamal Hartnett 18 h State New Effort 3	387 Design form Jamal Hartnett 12	388 Code form Raisa Pokrovsk 6

The Scrum Master probably presents this screen.

We go through each story and discuss status.

We an do the normal:

- What you did
- What you will do
- Any obstacles

But really, the standup is about delivering stories and the sprint, not individual performance.

	Day 1	Day 2	Day 3	Day 4	Day 5
Week 1:	Sprint Planning Product Owner reviews stories with team	Code S	itory 1	Code Story 2	te test cases
 Sprint Planning Start Sprint Team estim seque TEAM 		Write Test Ca	se Story 1	Write Test Case Stor	y 2 Code
	Finalize Acceptance Criteria Team defines, estimates, and sequences tasks TEAM COMMITS!	Daily Standup Daily Standup Text Story 1			Test Story 2 Accept Story 2
Week 2: > Build > Test > Accept > Complete	Code Write Forde Write Test C Daily Standup	Story 3 Write Fode ase Story 3 Daily Standup	Build/Test/Accept Write test cases Write Code Execute tests Accept Stories Accept Story 3 Accept Story 3	Deploy Deploy to live production environment	Sprint Wrap-Up (2 to 4 hours) Retrospective Review velocity SWAG Stories for next sprint
	Next Sprint Pre-Work				
	Grooming Meeting		Code Review		

Sprint Refinement/Grooming

The Goal: Envision the next sprint, while the current sprint is being delivered. Refinement accelerates sprint planning.

Benefits and process:

- Usually for an hour one or two times per sprint
- Allows time for the team to acclimate to a story
- Allows time to define the detailed acceptance criteria
- Allows time to discuss the story with multiple stakeholders
- May expose a long lead time dependency





Sprint Review/Demo

The Goals: Review the work with stakeholders and end users. When long term initiatives are involved, provide status on progress with the current sprint, and projections for the initiative (next milestones, ETA for delivery).

Rating	Practice				
Ideal	Review high level status of the sprint, commitment vs actuals.				
Ideal	Key stakeholders attend the review				
Ideal	Select users see a demonstration of functionality (may allow stakeholders				
	to leave)				
Not Advised	A scrum team only review. This implies the work is not considered valuable				
	outside of the team.				
Not Advised	Segueing the review into a retrospective				
Ideal	Adjustments and/or additions made to the backlog, based on feedback				





End of Sprint Review

You should be able to perform a Review and/or Demo with Teams - or any video conferencing tool.

	Story	Status
/	Design	
5	1194 Design Modal Pop Up	All complete
8	1126 Update sprint 8 prototype Round2 usability	All complete
5	1054 Design Error Page	All complete
8	1113 Design Deposit Strategy	Rollover
13	1112 Moderated Member Testing	All complete
8	1057 Update Proto based on Usability test	All Complete
39	Total	
	Development	
5	1127 Mock Service	Rollover
3	900 MFA Spike	All Complete
8	<u> 1056 Login Happy Path (Mobile) – FULL DEV!!!!</u>	All Complete
8	<u>977 Login Unsuccessful (Mobile) – FULL DEV!!!!</u>	All Complete
5	1124 Restricted Error – Help and Support	All Complete
5	899 Analytics Spike	Complete* (Azure Spike to follow)
3	1173 Add Environment Checks to UI Application	All Complete
32	Total	

Retrospective

The Goals: Review the health of the Scrum process. Identify processes working well, identify and prioritize the processes that need improvement. Leave the meeting with a clear improvement plan for top priority items.

Rating	Practice				
Ideal	Have the team break into subgroups and do mini-retros, then regroup and				
	share ideas.				
Ideal	Prioritize the improvements and select the top two or three to address in the				
	next sprint(s)				
Ideal	Create improvement stories in Azure DevOps, and add them to the sprint				
	backlog				
Ideal	Have the Scrum Master Reviewed with the team a week before sprint				
	planning				
Acceptable	Use the Azure DevOps retrospective template to log thoughts, before the				
	retro				
Not Advised	Have the team tell the Scrum Master their thoughts, real time in the meeting				
Ideal	If we are experiencing significant shortfalls to commitment (say less than				
	80% of commitment), bypass the usual stop/start/continue, and immediately				
	move to root cause analysis, and solutioning, for coming up short				

Stop

 Having General description tasks, create detailed tasks with good definition

Start

- More collaboration and teamwork
- Synchronizing JIRA with the physical wall
- Open and impromptu team communications
- Impromptu design meetings

Continue

- Linking the big picture to team work
- Playing nice with others
- Having Conversations in JIRA
- Updating JIRA daily
- Linking related issues in JIRA
- Celebrating success with food
- Identifying constraints (integrated processes)
- Using VSM room more
- Assign points to stories before sprint planning
- Identifying stretch goals





End of Sprint Retrospective

Tools like Microsoft Teams have retrospective templates built in. Team members can enter their thoughts before the meeting starts. We can discuss and vote during the meeting.

Overview	Retrospectives History History Sprint 21 Retrospective V ··· Collect Group Vote Act					
🕄 Boards	What went well	😟 What didn't go well	Q Ideas to Consider	Actions to Take		
🗘 Work items	+ Add new feedback	+ Add new feedback	+ Add new feedback	+ Add new feedback		
Beards	:	:	:			
冒 Backlogs	A relaxed release to production let us	Digital Cloud Migration leveraging POD	Moving from Slack to Teams as	Leave Slack on Monday 4/19.		
Ď₊ Sprints	find and fix some bugs that we could only find there	resourcesGarrett	moving away from Slack and using Teams.			
$=_{\nabla}$ Queries						
Delivery Plans	Apr 9th, 2021 11:44 am	Apr 9th, 2021 9:14 am	Apr 9th, 2021 10:33 am	Apr 9th, 2021 12:03 pm		
⊘ Personas	#0	#0	#1	#0		
Q Portfolio++		: :	> 2 Items			
Retrospectives	Good collab with Erin!	User lesting.com glitch could not launch study on time :(Demo structure / plan / process			
🛗 Calendar						
🖗 Estimation Poker	Apr 9th, 2021 11:42 am	Apr 9th, 2021 11:42 am	Apr 9th, 2021 11:43 am			
🖉 Dependency Tracker	#1	#1	#2			
😢 Repos	:	:				
Pipelines	Alpha 2 Tester feedback has been very positive so far	some of those bugs that slipped by I should have found :(
📥 Test Plans	Apr 9th, 2021 11:43 am	Apr 9th, 2021 11:44 am				
பி Project settings 人	#2	#2				

Where Scrum is Effective

- Innovative environments with emerging requirements
- Think new product development, when experiments are needed to understand the need
- When we can see more than a week or two into the future
- When we do not expect those needs to change on a daily basis



Models Where Scrum is Less Effective

- Interrupt driven work, like a customer support desk, where priorities change often and we cannot see past a week or two for our needs
- Simple, repeatable work, think assembly line – where you can apply the same pattern or steps every time
- Firefighting where immediate triage is needed



• Maintenance work (help desk resolving bugs) Kanban is a better fit here

Attributes That Set You Up Well for Agile

- An entrepreneur mentality (hungry not a monopoly)
- Co-located team and product owner
- Where work is prioritized before it reaches the teams
- Volatile or dynamic environments
- Competitive environments where innovation is valued
- Companies with low turnover and high maturity
- When management understands that project discoveries cannot be eliminated with extensive upfront planning
- Where teams are recognized more than individuals
- Where managers are recognized for what their *teams* do not what *they* do
Attributes That Set You Up Well for Agile (2)

- When the customer is available for frequent feedback and interaction
- When management understands, and supports, estimates vs. commitments
- Companies with work/life balance (40 hour work week, sustainable pace)
- Where customer value is more important than standardized process
- Technical excellence scalable architecture, automated testing, continuous deployment
- A culture of transparency information is there and easily accessed
- When there is an existing culture of continuous improvement and process optimization
- When leadership takes the *team level* Agile training

Kanban Better Fit

When team members cannot be dedicated

When work can be done independently

On production support work

On lower priority work

On teams without Agile training

On work that is not time-boxed

On urgent work that cannot wait for a sprint

Note:

If you have not done Agile or Scrum before – Kanban provides a great foundation

Scrum a Better Fit

Team dedicated to the project

When we can see at least one to two weeks into the future

The deliverables require teamwork

Experienced team

Interactive Customer

Prioritized work

Critical project or time-boxed

Few dependencies on other groups/departments

Team trained on Agile

Questions

