Introduction to Agile/Scrum
Module #1 – Agile In a Nut Shell

Goal

- Provide insight into Agile methodologies leveraged by CWDS
- To learn what it means to be a part of a Agile team and what responsibilities exist within said team
- To enjoy some hands on practice
- Answer some of your questions and prioritize those for which we don’t have time

Today's Guidelines for Success

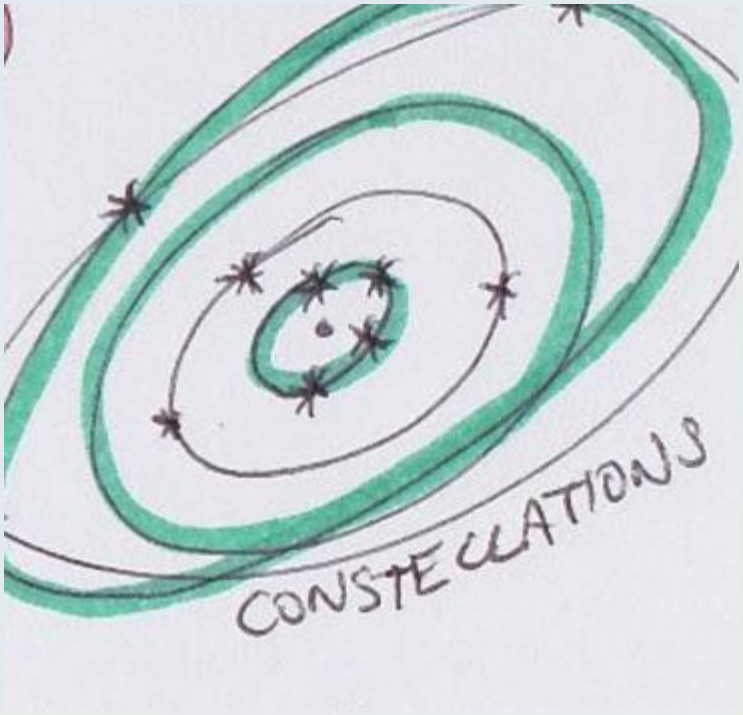
- Cell phones off
- Questions
- Process can be mastered
- Going Agile is a culture change
- Have fun!
- Return the supplies at the end

Today's Schedule

- 8:30 Introductions
- 9:45 Break
- 10:00 Training resumes
- 11:30 Lunch
- 12:30 Training resumes
- 3:00 Break
- 3:15 Training resumes
- 4:30 Training ends

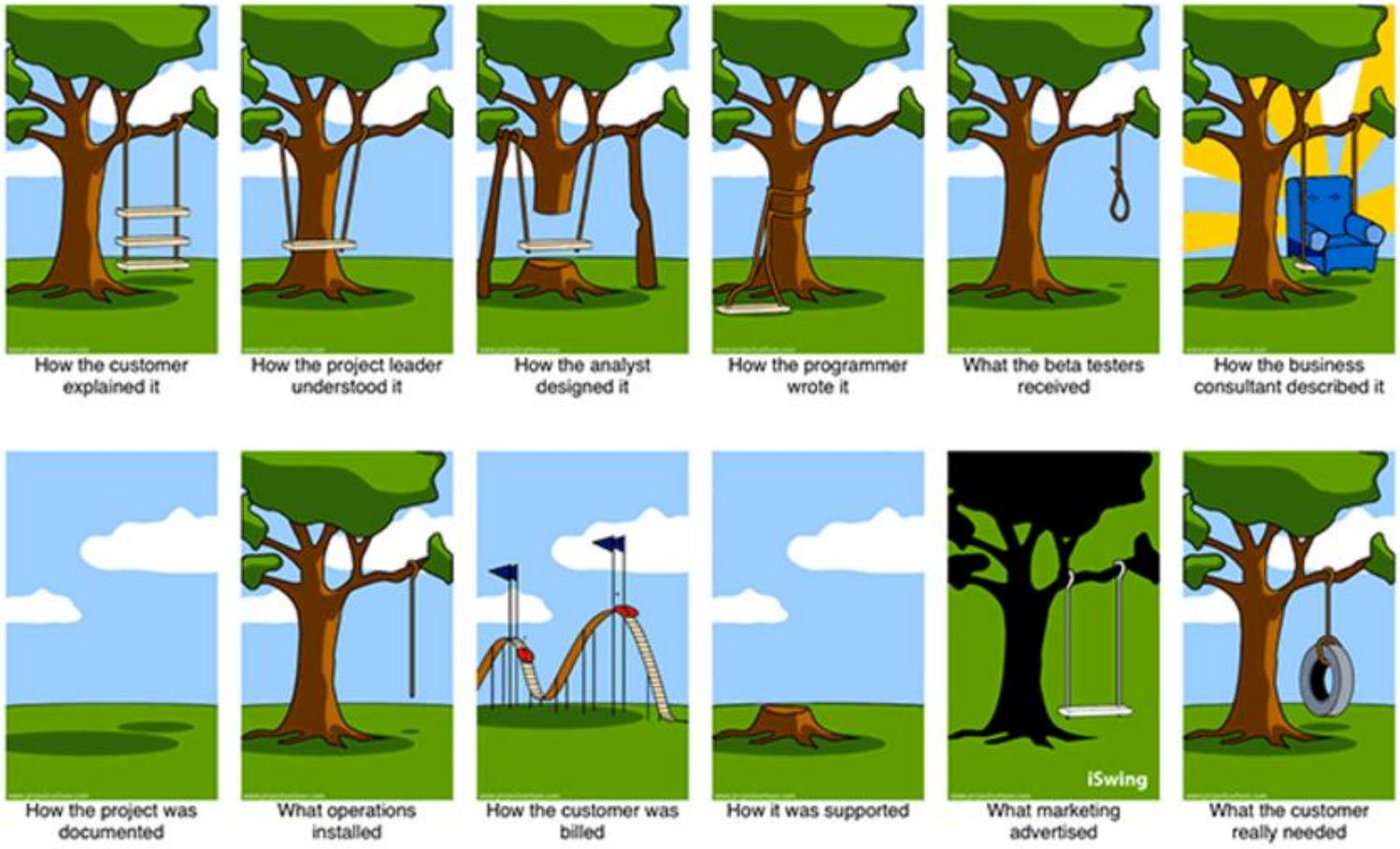
Exercise

Constellations



How Projects Really Work (version 1.5)

Create your own cartoon at www.projectcartoon.com



So You're on a New Agile Team, Now What?!?!?

What is Agile?

- Digital Delivery Service Model that aligns development with customer needs and company goals, for the life of the project and beyond.
- Promotes a disciplined process of transparency, frequent inspection and adaptation
- Leadership philosophy that encourages teamwork, self-organization and accountability
- Set of engineering best practices intended to allow for rapid delivery of high-quality software, but which can also be used in non-software teams

Why Agile?

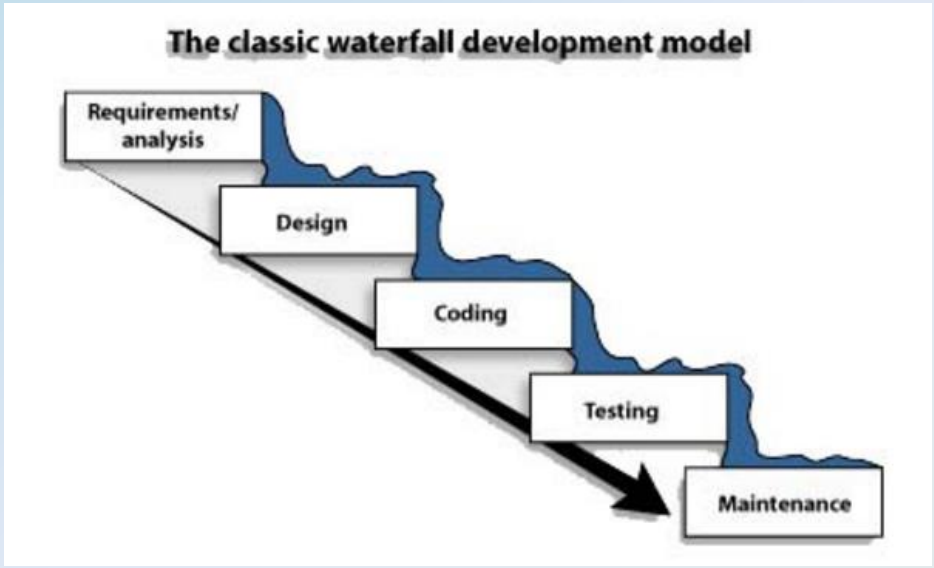
Waterfall

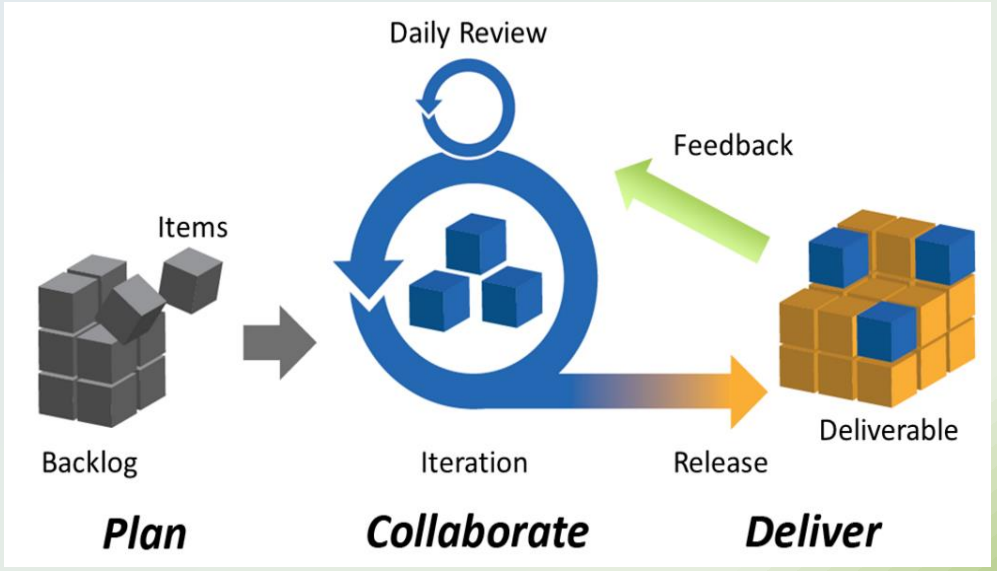
- Long development cycles
- Unable to deal with large feature sets
- Requirements - a moving target
- Features & deadline already determined before effort
- Result - overproduction of feature

Agile

- Timely delivery / sooner to market
- Easier to manage change
- More visibility/transparency
- Immediate & better collaboration between the Business and Development groups
- Develop the 20% of the features that deliver 80% of the value

Waterfall vs. Agile





CWDS Scrum Framework in a Nut Shell

- Scrum Master
- Product Owner
- Performance Analyst
- Development Team
- Extras

4+ Roles

- Vision
- Team Agreements
- Definition of Done
- Roadmap
- Product Backlog
- Release Plan
- Sprint Backlog
- Product Increment

8 Artifacts

- Daily Scrum
- Sprint Planning
- Backlog Refinement
- Sprint Review
- Sprint Retrospective
- Release Planning
- Release Retrospective

7 Activities

Agile Manifesto

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiations

Responding to change over following a plan

- While there is value on the right, Agile values items on the left **(blue)** more

Basic Agile Concepts

- A Roadmap is developed by the Business
- Features requested by the Business are handed to the Product Owner
- Product Owner creates an epic or feature, but works with the team to break down the epic into user stories
- A User Story is a simple expression of functionality that will be valued by the customer
- A collection of User Stories is a Product Backlog
- The Product Owner prioritizes the Product Backlog to ensure top priority items are delivered first

Basic Agile Concepts

- A typical Agile Team is comprised of a Product Owner, Scrum Master, Analyst and delivery teams
- The Team ***pulls*** User Stories from the Product Backlog into the Sprint Backlog
- The Team spends the next iteration/sprint developing the Sprint Backlog
  - Daily Standups
  - Review
  - Retrospective
  - Sprint Planning
- The team participates in Release Planning 1-2 days every quarter.

Scrum Values

- Commitment
- Focus
- Openness
- Respect
- Courage

Scrum Project Roles

Activity	Role	Responsibilities
Manages the Vision	Product Owner*	Establishes and communicates product vision. Creates release plan and backlog with the team.
Manages the Customer Feedback Loop	Performance Analyst**	Research, market analysis, data quality. Liaison between the business vision and the technical execution; asks the question, “Does this service as designed and built meet the need of the customer?”
Manages the Development Iteration	Team	Develops the highest priority features. Defines specific tasks, self organizes to manage its iteration commitments.
Manages the Process	Scrum Master	Enforces the Scrum process. Facilitates daily stand up, backlog grooming, sprint planning, demos, retrospectives. Removes impediments. Liaison with other departments.
Manages the Release	Product Owner (in harmony with the Scrum Master)	Plans the releases and communicates product release plan.

* Service Manager

** Performance Analyst

User Story Format

As a (who needs this?)

I want (what is needed?)

So that (what is the value?)

Acceptance Criteria

- Helps define what the business needs
- Helps Developers know they have satisfied the requirements
- Helps build the right products & features

Exercise

- Pick a group (not your current team)
- Pick a project

Exercise

Working alone, write 1 user story using the format of:

As a (who needs this?)

I want (what is needed?)

So that (what is the value?)

Include acceptance criteria.

Exercise

Working in your team, write 1 user story using the format of:

As a (who needs this?)

I want (what is needed?)

So that (what is the value?)

Include acceptance criteria.

Put your user stories on the task board

Backlog

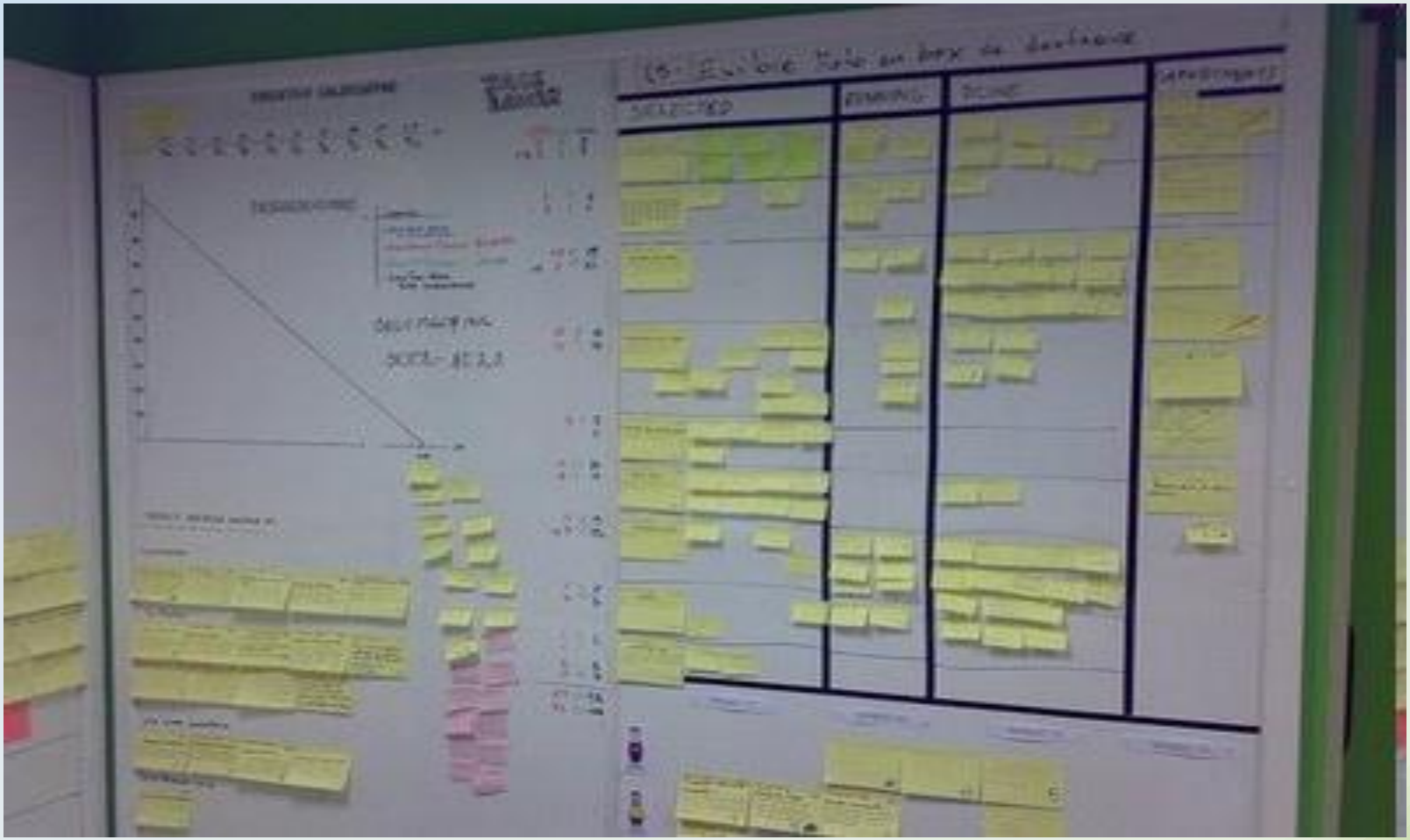
- User stories focus on the "***what***" not the "how" (never, never, never the "how")
- Product Owner prioritizes backlog based on business value
- Team advises on technical priority

- Completed backlog items are displayed on the Great Wall of Done. What's the Great Wall of Done? I'm glad you asked.

Sprint Planning

- Every iteration begins with the sprint planning session
- CWDS has adopted a two week sprint schedule for most teams
- Product Owner decides which stories are of the highest priority to the release and will generate the highest business value
- Team pulls work from the prioritized backlog into the sprint backlog and breaks them into tasks

Storyboard



"With Agile we are adding so many meetings."

- Former Team Member

In truth, there is only communication and planning sessions.

“There are no meetings in Agile”

“But there are daily stand ups…”

…Yes, but stand up isn’t a meeting.

The daily stand up is used for the team to check in with one another and ***plan*** their day with regard to the progress of their committed work.

Daily Scrum/Stand Up Format

- Time-boxed meeting: 15 minutes
- Team Members answer three questions
  - What have you done *for our sprint* since we last met
  - What will you do you *for our sprint* before we meet again
  - Is there anything standing in your way of completing work *for our sprint*

“There are no meetings in Agile”

“But there is a Sprint Demo/Review…”

…Yes, but the Sprint Demo/Review isn’t a meeting.

The Sprint Demo/Review is used for:

- the team to communicate to Stakeholders the progress they made on their forecasted work during the iteration/sprint
- a time to communicate iteration/sprint challenges
- a time to open further communication with the business

Sprint Review

There are no meetings in Agile

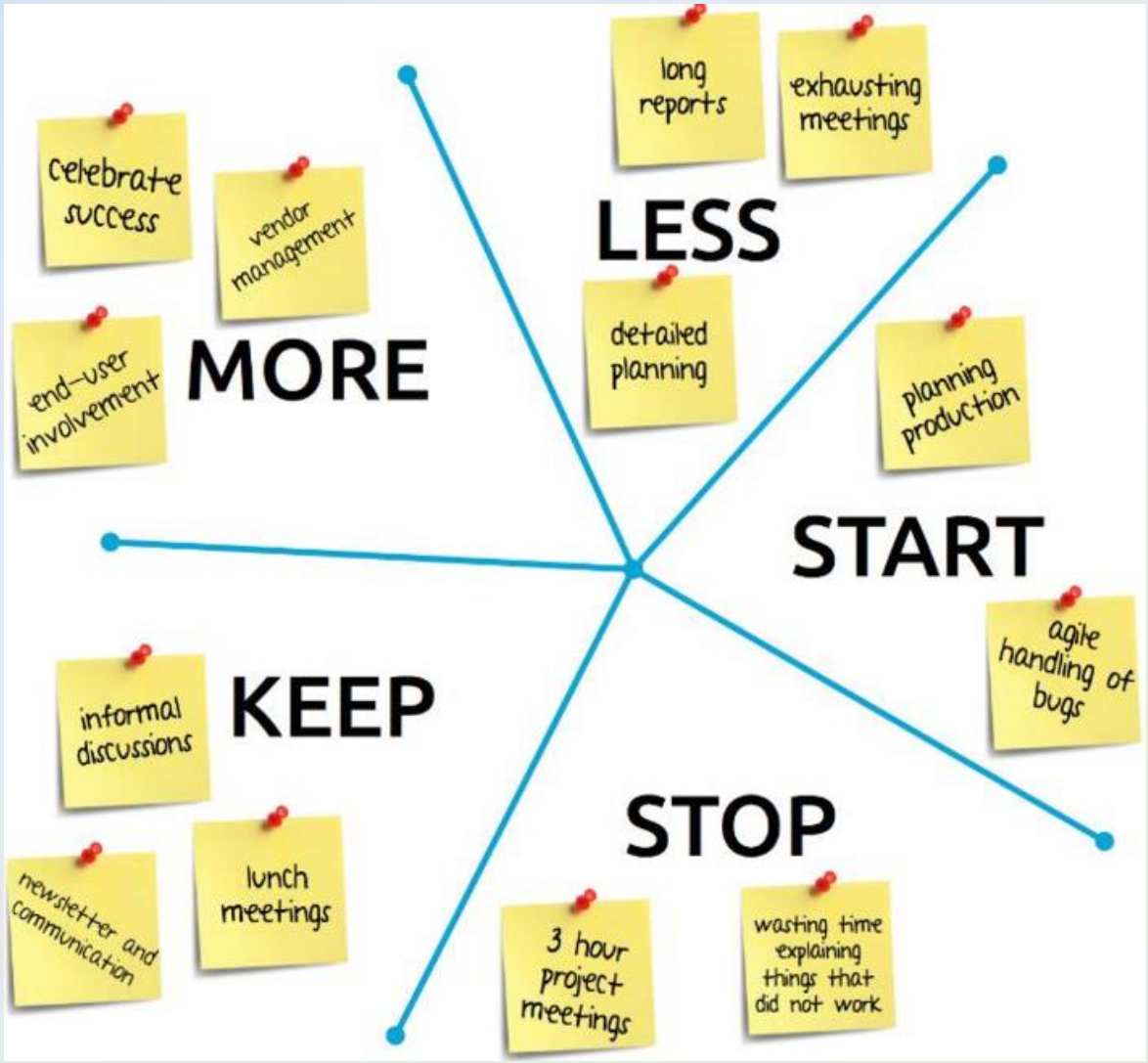
"But there is a Sprint Retrospective…"

…Yes, but the Sprint Retrospective isn't a meeting.

The Sprint Retrospective is used for:

- the team to talk (communicate) about their sprint in an open, unfettered manner
- a time to communicate iteration/sprint challenges
- a time to identify, commit to and communicate strategies that will help to make them a better team

Sprint Retrospective



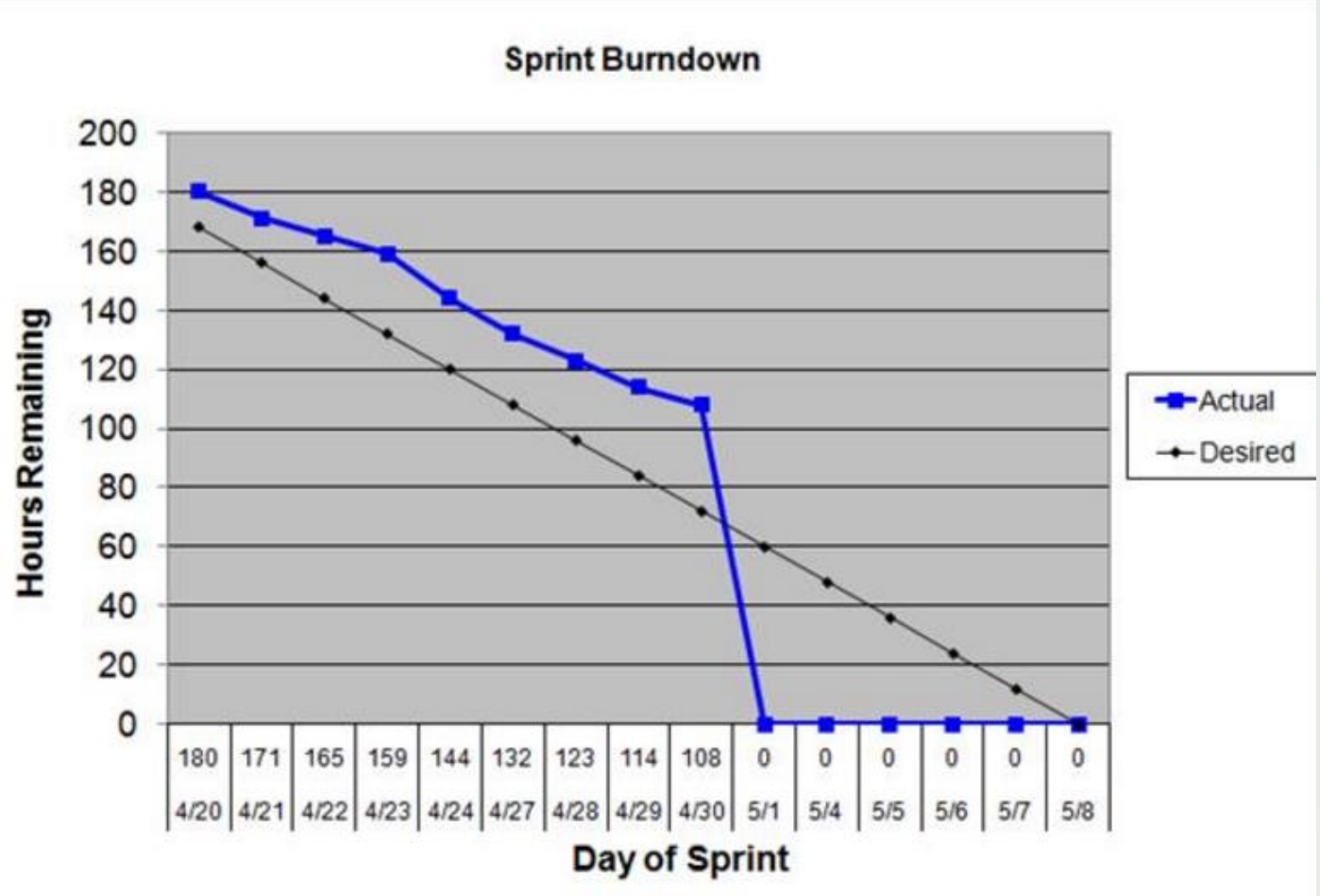
Other Agile Terms CWDS Will Use

- Team Vision
- Team Agreements
- Working Agreements/Shared Values
- Definition of Ready
- Definition of Done
- Daily Burndown Chart
- Release Burnup Chart
- Velocity

Burndown Chart

- A **burn down chart** is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run **chart** of outstanding work. It is useful for predicting when all of the work will be completed.

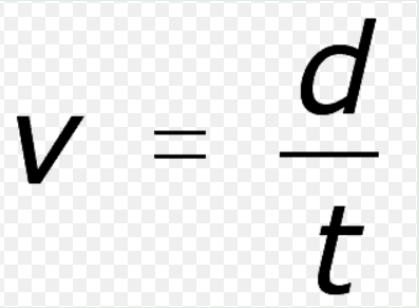
Burndown Chart



What is Velocity?

Some things to know about Velocity:

- For Planning
- Average Velocity
- Comparing velocities
- Velocity and management
- Velocity will be impacted:
  - Loss or addition of a team member
  - Change in technology



Exercise

Puzzle time

What we've learned about Agile

- Features requested by the Business are handed to the Product Owner
- Product Owner creates an epic or feature, but works with the team to break down the epic into user stories
- A User Story is a simple expression of functionality that will be valued by the customer
- A collection of User Stories is a Product Backlog
- The Product Owner prioritizes the Product Backlog to ensure top priority items are delivered first

What we've learned about Agile

- A typical Agile Team is comprised of a Product Owner, Scrum Master, Analyst and delivery teams
- The Team ***pulls*** User Stories from the Product Backlog into the Sprint Backlog
- The Team spends the next iteration/sprint developing the Sprint Backlog
  - Daily Standups
  - Review
  - Retrospective
  - Sprint Planning
- Requires teamwork

Introduction to Agile/Scrum
Module #2 – Team Structure

Team Characteristics

- Seven members (+/- two)
- Cross functional team members
- Self-organizing
- Preferably co-located
  - Facilitates communication
- Dedicated Product Owner
- Dedicated Scrum Master
- Add teams, not team members

Team Artifacts

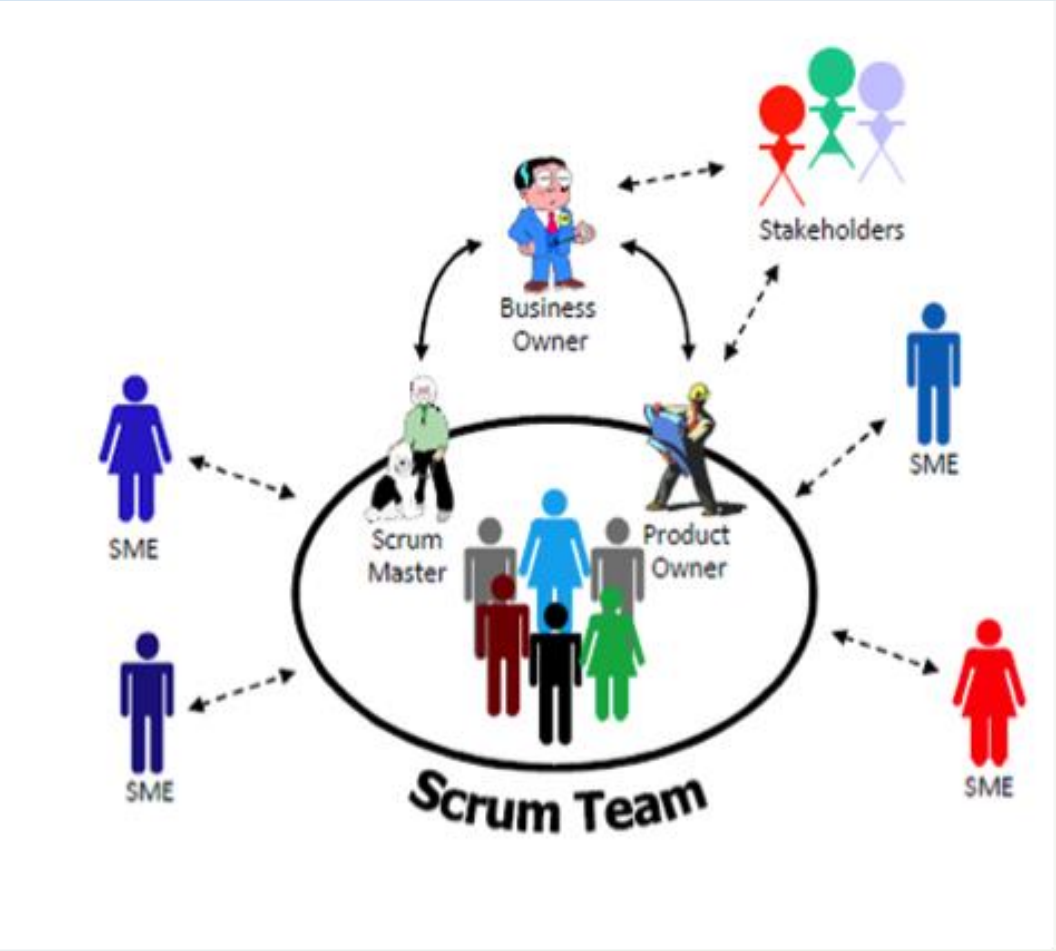
Team Agreements

* Core hours 9 a.m.-4 p.m.
* Daily stand-ups 10 a.m. sharp
* Done includes testing
* Respect the build
* When someone asks you for help say "yes"
* Weekly demo Tues 11am
* Customer available 1-3 p.m.

Working Agreements

* We don't cut corners
* No broken windows
* It's OK to disagree
* We can handle the truth
* Don't assume—ask
* When in doubt—write a test
* Crave feedback
* Check your ego at the door

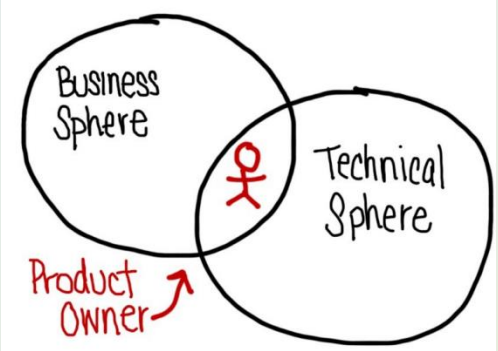
Scrum Team



www.adapty.com

CWDS Product Owner*

- Defines and delivers the product vision
- Set priorities and makes/logs decisions about features and technical implementation details based on user, policy, technical and business requirements
- With the Core team, conducts user based testing
- Facilitates the weekly or bi-weekly service team meeting, reviewing burn down charts, reviewing the icebox, etc.



* Synonymous with Service Manager in some cases

CWDS Product Owner Cont'd

- Provides needed clarification regarding business or technical scope/direction
- Develops, prioritizes and refines the backlog as needed
- Drives business change, but does not drive process
- Accepts or rejects work results throughout the Sprint
- Manages & communicates the release plans

Performance Analyst*

- Conducts analysis before/during/post implementation to ensure that their digital service meets performance requirements
- Data analytics (not team analytics) for feature development and review after implementation
- Make recommendations based on the quantitative and qualitative evidence gathered via web analytics, financial data and user feedback

* Not in Agile/Scrum

Performance Analyst cont'd

- Collect and present the key performance data and analysis for their service to Stakeholders
- Be a revolutionary in the way in which government continuously measures, assesses, and improves performance in transacting with the public
- Support service managers by generating useful information and translating it into actions that will allow them to iteratively improve their service for users
- Understand how the services, activities and strategies in the area work together to create value for the user
- Monitor emerging issues and trends, which might impact or benefit team's work

Business Core Team *

- Attend the daily Scrum Meeting, the weekly service team meetings, the Sprint Planning & Review sessions and the Sprint Retrospective session
- Conduct user acceptance testing with Product Owner
- Develop, revise and review artifacts in an Agile manner
- Provide expert guidance and support for business and technical decisions
- Participate in Backlog Refinement with the team
- Communicate availability during Sprint Planning

* Resource Pool = Business Core Team

Scrum Master

- Coaches the team
- Protects the team
- Facilitates the team events
- Removes impediments for the team
- Protects the Scrum process
- Monitors team Scrum analytics
- Partner with Product Owner for backlog development, refinement, release planning and roadmap development

Delivery/Scrum Team

- Self organized within the Agile framework
- Partners with the Product Owner, Performance Analyst and Business Team for backlog development and refinement
- Responsible for developing features within acceptance criteria
- Defines tasks, forecasts and manages commitments
- Demo’s completed stories to Product Owner throughout the sprint
- Reports impediments to the Scrum Master
- Works in a cross-functional capacity

Delivery/Scrum Teams

- Are self-disciplined
- Make collaborative decisions
- Share roles, responsibilities & cooperate
- Are accountable for Sprint goals
- Move from divergence to convergence
  - Forming
  - Storming
  - Norming
  - Performing

Delivery/Scrum Teams

May consist of

- Developers
- Testers
- Technical Writers
- Architects
- Analysts (Business, Market, Performance)
- Whomever is needed to complete the sprint
  - User, Resource Pool, Configuration Mgmt, SME(s), Dev Ops

What we've learned about Team Structure

- Seven team members (+/- two)
- Cross-functional within their team
- Self-organized within Agile
- Need to communicate, collaborate and plan
- Team members should be dedicated and co-located

Introduction to Agile/Scrum
Module #3 – Stories & Epics

Concept to Project

Ideas gathered

- Focus groups, interviews, feedback from users
- Listen to customers and be prepared for undeveloped ideas

Refine Ideas

- Interpret user statements into user needs
- Focus on “WHAT” not “HOW”
- Avoid assumptions and adding extra detail

What is a User Story

- A User Story is a simple statement describing the value of functionality from a user's perspective

- User Story format

  As a (who needs this?)

  I want (what is needed?)

  So that (what is the value?)

Why User Stories

- It is impossible to know all the requirements for a product or project in advance
- Emergent requirements are those that the users cannot identify in advance
- Every project has emergent requirements

User Story Concept

User stories focus on the “what” not the “how”

The 3 C’s of a User Story

Card

The story is documented on a card representing the requirement or use case.

Conversation

The requirement is communicated from business to technology.

Confirmation

The acceptance criteria makes the simple card/conversation approach possible.

INVEST Model

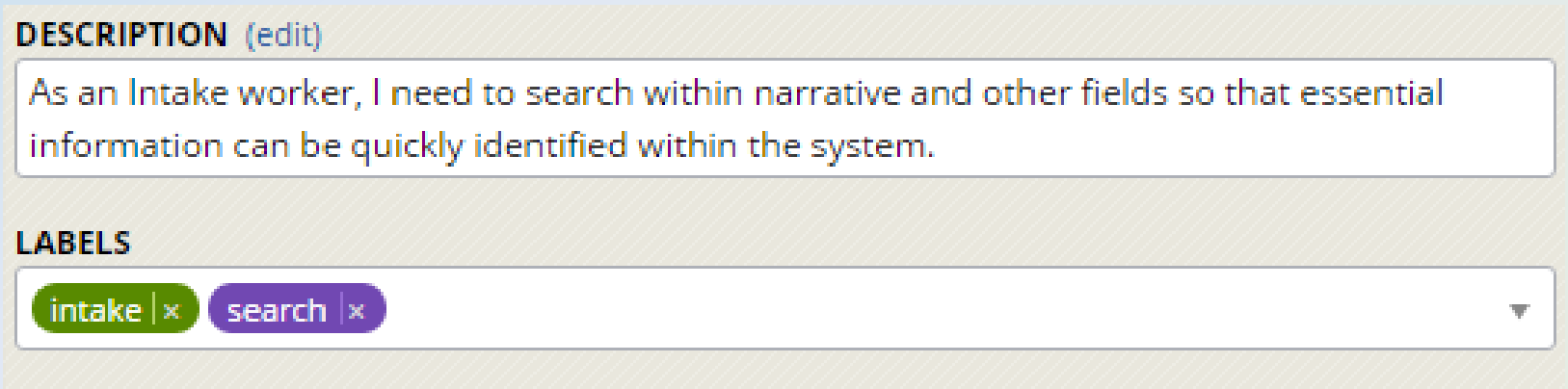
http://agilesoftwaredevelopment.com/blog/vaibhav/good-user-story-invest

- Independent
- Negotiable
- Valuable
- Estimable *
- Small
- Testable

All *User* Stories should strive to follow the INVEST model.

User Story Example

Current user story in Pivotal



User Story Example

As a mom, I want your room cleaned, so that I am not embarrassed to have company over and so that a Hazz Mat team does not quarantine our house.

Acceptance Criteria/Tests

- Acceptance criteria (test) is the confirmation that the user story has been satisfied. This is *not* the Definition of Done.
  - Acceptance criteria are short and easy to understand statements that are testable
  - The statements detail what the user expects the product/feature to do
  - Acceptance criteria enable developers to understand that they have satisfied the user requirement

Acceptance Criteria Example

As a mom, I want your room cleaned, so that I am not embarrassed to have company over and so that a Hazz Mat team doesn’t quarantine our house.

Acceptance Criteria:

- There are no dishes in the room
- There is no laundry on the floor
- The bed is made properly
- There is no trash in the room

Decomposing the Epic

EPIC: As a mom, I want your room cleaned, so that I am not embarrassed to have company over and so that a Hazz Mat team doesn't quarantine our house.

Acceptance criteria of an epic becomes lower level User Stories in the break down process:

- There is no laundry on the floor (story 1)
- There are no dishes in the room (story 2)
- The bed is made (story 3)
- There is no trash in the room (story 4)

Decomposing the Epic

User Story 1

As a mom, I want all the *laundry* removed from your room and taken care of properly so that you have clean clothes, don't smell and aren't wrinkled.

Acceptance Criteria:

- No laundry is under the bed, on top of the dresser, on the night stand or floor.
- All laundry that was removed from your room has been washed, dried, and put away properly.
- All clean laundry has been folded properly, has been ironed or is hanging on hangars and is in the dresser or hanging in the closet. NOTE: Sniffing does not count as cleaning the laundry.

Decomposing the Epic

User Story 2

As a mom, I want all the ***dishes removed*** from your room and taken care of properly so that we can use our dishes.

Acceptance Criteria:

- No dishes are under the bed, in/on the dresser, on the night stand or floor.
- No drinking glasses, cups, silverware, any size plate, any size bowl, even paper dishes and plastic spoons are in the room.
- All dishes that were removed from your room have been washed, dried, and put away in the correct cabinet. All paper and plastic dishes have been put in the proper recycling bin.

Decomposing the epic

User Story 3

As a mom, I want your bed made properly so that the room looks nice.

Acceptance Criteria:

- The sheets have been washed and dried.
- The sheets were put back on the bed properly.
- The blankets and bedspread are neatly laid out on the bed.

Are there stories other than “User Stories?”

Research or Spike Stories

- A Research Story is when a team wants to do some web research, poking around in existing code. Outputs are generally research/decision documents.
- A Spike Story is when a team does some coding to bring life to their Research. Outputs are generally a proof of concept. It is perfectly acceptable to have ‘throw away code’ at the end of the sprint.

Research (←Action) Smart Distance algorithm for multi-paragraph web pages **so that** (←Value) we learn what search functionality exists

Prototype (←Action) a Smart Distance algorithm for multi-paragraph web pages **so that** (←Value) we understand what search functionality will work best for our customer

Are there stories other than “User Stories?”

Device/System User Story

- While the user story is the primary, and most desirable form of story, not every system under development interacts with an end user. Sometimes, the “user” is a *device* (example: printer) or other *system* (example: transaction server).
- The goal is to continue to ask who the story is for to try to get to the perspective of the end user/customer.
- NFR’s fit nicely into this type of story (security, scalability, usability)

As a web crawler, **I want** a URL dictionary without duplicates to dead links **so that** the crawling process is faster.

Backlog Refinement

- User stories need to be simple enough that they can be accomplished in one iteration
- Large, complicated stories are called Epics
- Epics need to be refined (decomposed) into smaller, less complicated stories that can be accomplished in a single iteration

Backlog Refinement cont'd

- Team meets regularly to 'refine' the stories in the backlog; to bring them to the Definition of Ready
- Team works on the highest priority stories that are not in sprint
- Clarifies things like, "and", "or", "properly", "but", "etc."

Example of an Epic Story

Succeeding With Agile: Software Development Using Scrum by Mike Cohn, p.246

- As a user, I want to log into the system so that my information can only be accessed by me

Exercise

In your groups, decompose the following Epic:

As a user, I want to log into the system so that my information can only be accessed by me

Example of an Epic Story

Succeeding With Agile: Software Development Using Scrum by Mike Cohn, p.247

Decompose Epics into smaller stories

- As a registered user, I can log into with my user name and password so that I can trust the system
- As a new user, I want to register by creating a user name and password so that the system can remember my personal information
- As a registered user, I can change my password so that I can keep it secure or make it easier to remember
- As a registered user, I want the system to warn me if my password is easy to guess so that my account is harder to break into
- As a forgetful user, I want to be able to request a new password so that I am not permanently locked out if I forget it

Exercise

- Product Owner, please identify the epics; team can discuss, challenge the decision
- Product Owner, please prioritize the backlog; team can challenge with technical priority

Exercise

Working in your team, write 1 user story using the format of:

As a (who needs this?)

I want (what is needed?)

So that (what is the value?)

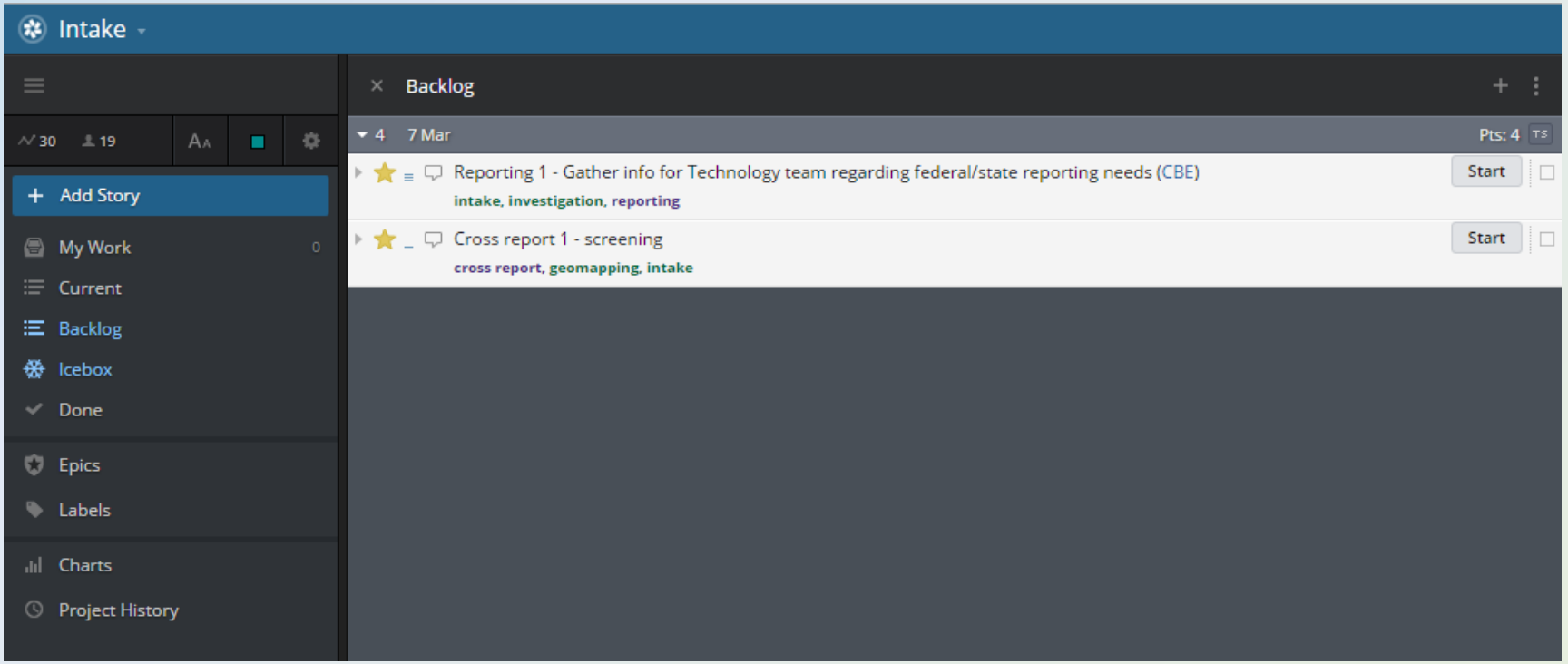
Include acceptance criteria.

Put your user stories on the task board

Backlogs Revisited

- A Backlog is where the stories live before they are pulled into sprint
- Product Owner continuously prioritizes (refines) backlog based on business value and respects technical prioritization
  - CWDS uses Pivotal Tracker software tool to manage backlog
  - Completed backlog items are displayed on the Great Wall of Done
- The Backlog is owned by the Product Owner

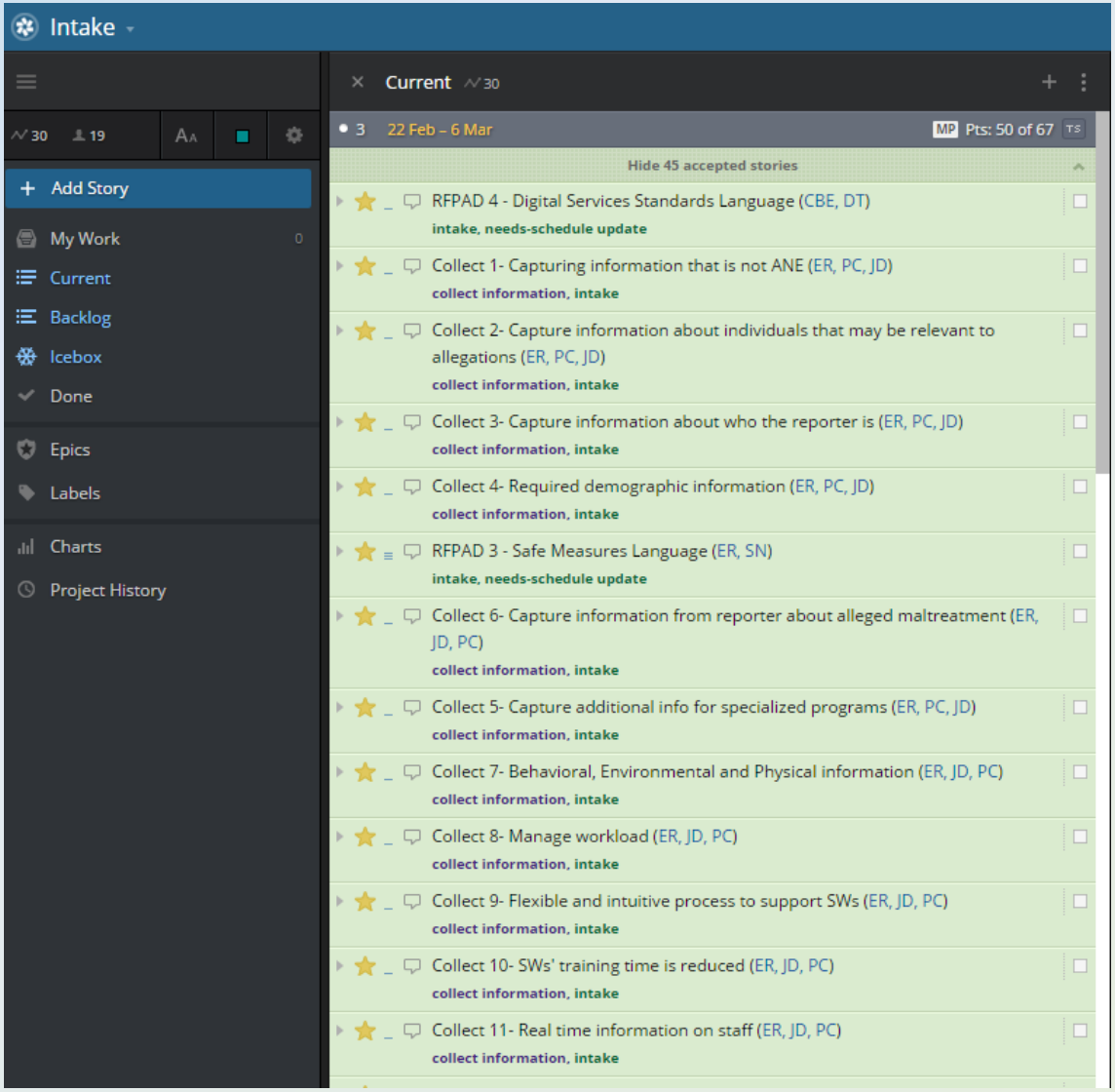
Pivotal Tracker Backlog Example



What is a Sprint Backlog?

- A Sprint Backlog is where the stories live once they are pulled into sprint
- No story should be pulled into sprint until it meets the Definition of Ready
- Once the sprint is started, no one removes a story from sprint; no minds are changed
- Additional stories are not added into the sprint without the commitment of the whole team
- The Sprint Backlog is owned by the team

Pivotal Tracker Sprint Backlog Example



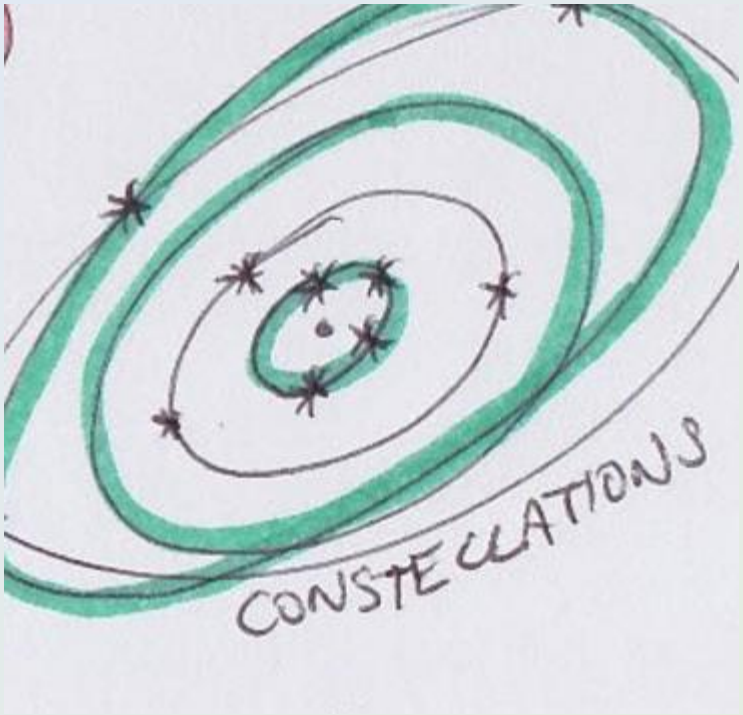
What we've learned about Backlogs

- User Stories describe What is needed, not How to create it
- Epics need to be broken down into small stories that can be completed in a single sprint
- A backlog is simply a collection of User Stories
- A sprint backlog is born when a team member pulls the story into sprint

Parking Lot

Exercise

Constellations



Upcoming Trainings

- Definition of Ready vs. Definition of Done
- What is Kanban?
- User Stories Unmasked
- Different Types of Estimating
- Facilitating the Scrum Events
- Scrum Master as Coach
- Participating in the Scrum Events
- The Oz Principles of Accountability
- Meaningful Working Agreements
- Understanding My Team
- Team Stats; What do they mean and who needs them?
- Release Planning

Questions?