

WRITING USER STORIES AND ACCEPTANCE CRITERIA

Professor Ernesto Lee

WELCOME TO THE
COURSE

INTRODUCTION

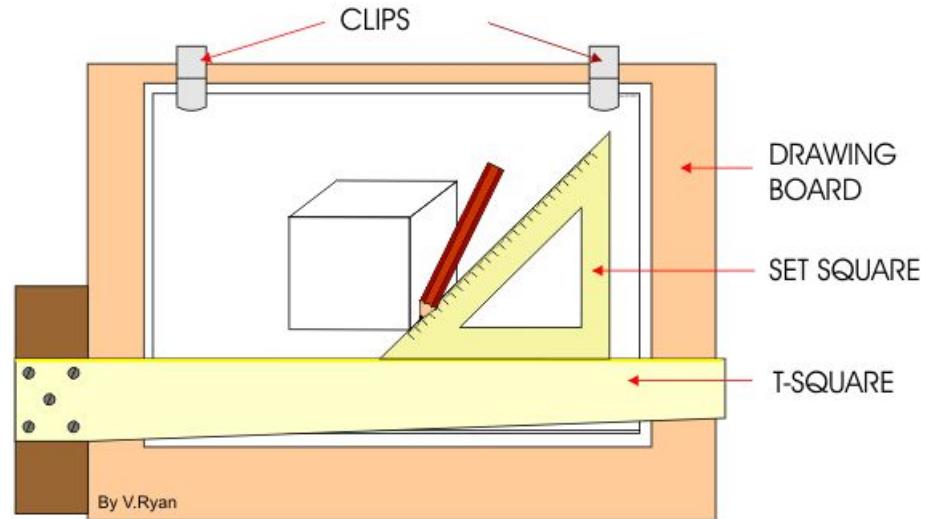
- Agile Recap
- User Story Fundamentals
- Acceptance Criteria
- Advanced Topics

USER STORIES ARE NOT JUST FOR AGILE

System Perspective

Versus

User Perspective

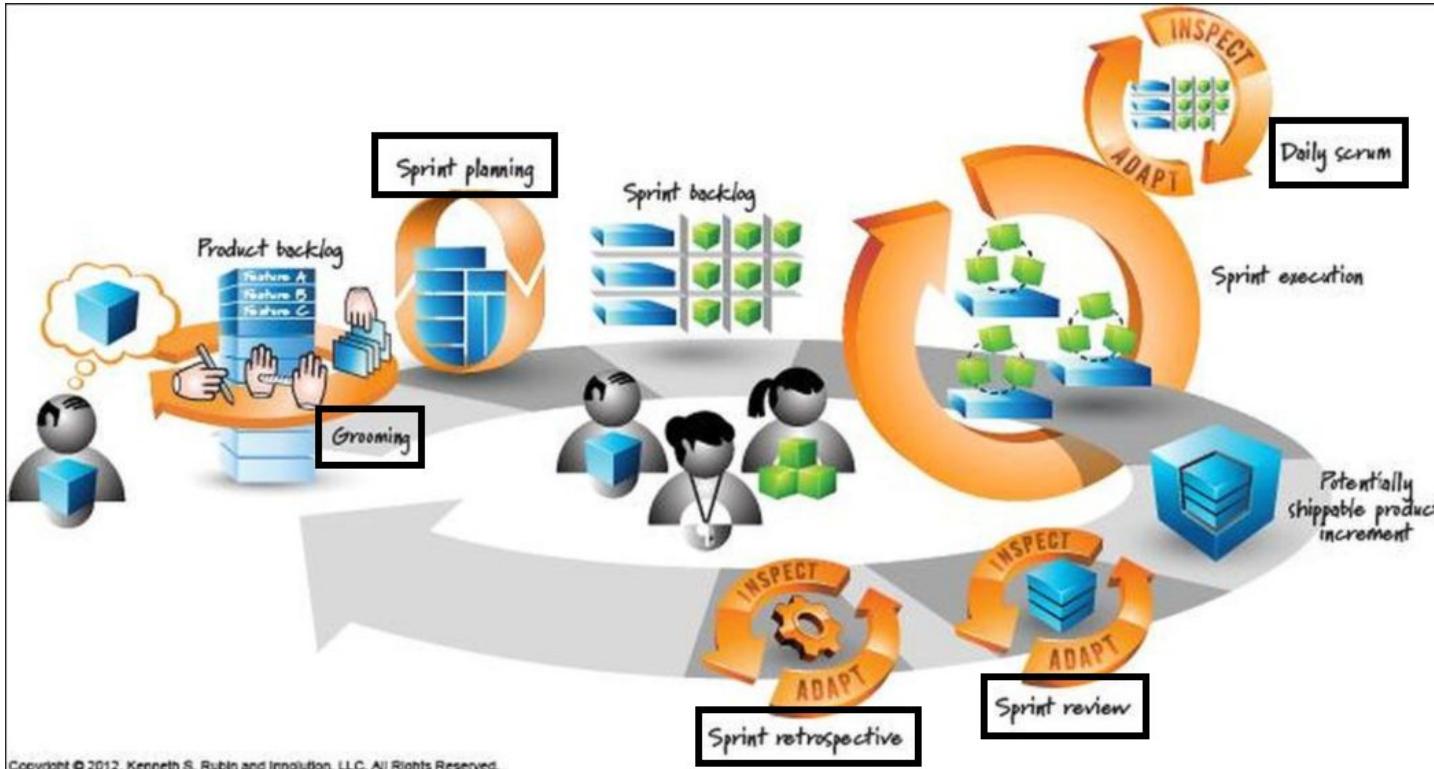


RECAP OF AGILE

Agility & Ways of Working

- Tony Goulart and Ernesto Lee

Overall Scrum Process



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

Backlog Grooming



Backlog Grooming

Purpose and Context

- Prepare User Stories for sprint planning
- Develop shared understanding of Customer needs as represented by user Stories
- Partner with Product Owner to optimize delivery of business value

Level of Facilitation

- Full-scale facilitation by the Product Owner
- Well-designed, closely facilitated task oriented, outcome driven, time-boxed

Outcomes / Results / Deliverables

- Well Groomed User Stories
Include:
 - Business Acceptance Criteria
 - Understand how to demo

Tools & Techniques

- Triangulation
- Powerful Questions
- Listening
- ATDD (Acceptance Test Driven Development)

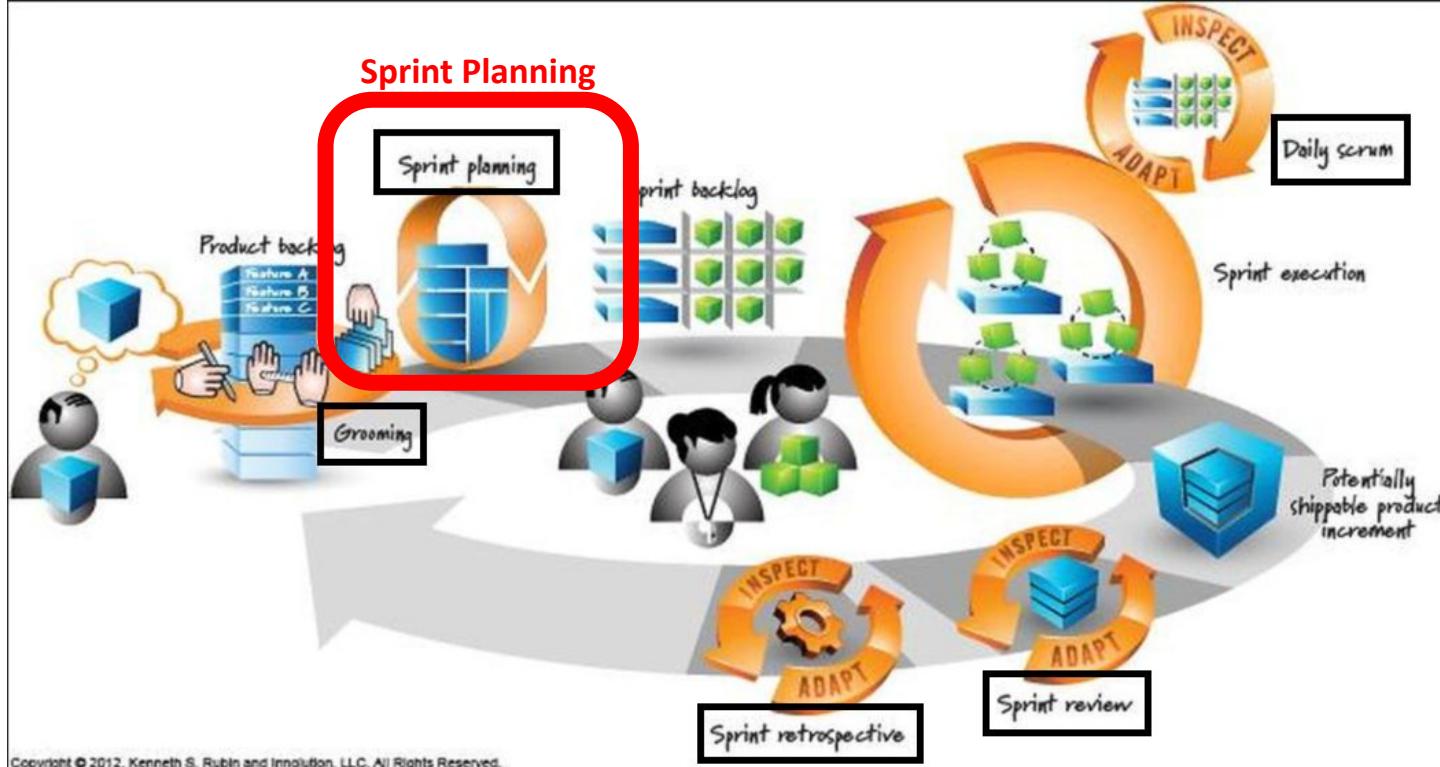
Success Factors

- Engaging & authentic
- Stories moving towards Definition of Ready
- Encourage conversation
- Transparency/Safety

Pitfalls

- Not actually done
- Going thru the motions
- Fear
- Insufficient support from the business stakeholders

Sprint Planning



Sprint Planning

Purpose and Context

- Aligns team and Product Owner on work plan / commitment
- Management gets transparency into the team's thinking instead of needing a detailed long range plan
- Helps team focus and control their destiny
- Provides empirical measure of teams capability / capacity

Level of Facilitation

- Full-scale facilitation by the Scrum Master with Product Owner
- Well-designed, closely facilitated task oriented, outcome driven, time-boxed

Outcomes / Results / Deliverables

- Defined Iteration Goals
- Iteration Backlog with acceptance criteria
- Team iteration Commitment
- Deeper shared understanding
- User Story Task Breakdown
- User Story Size Estimates

Tools & Techniques

- User Story Grooming
- Silent Work
- Brainstorming
- Story Refactoring
- Task Bucket Estimating
- Time Boxing
- Working Edge Cases First
- Just Enough / Just In Time

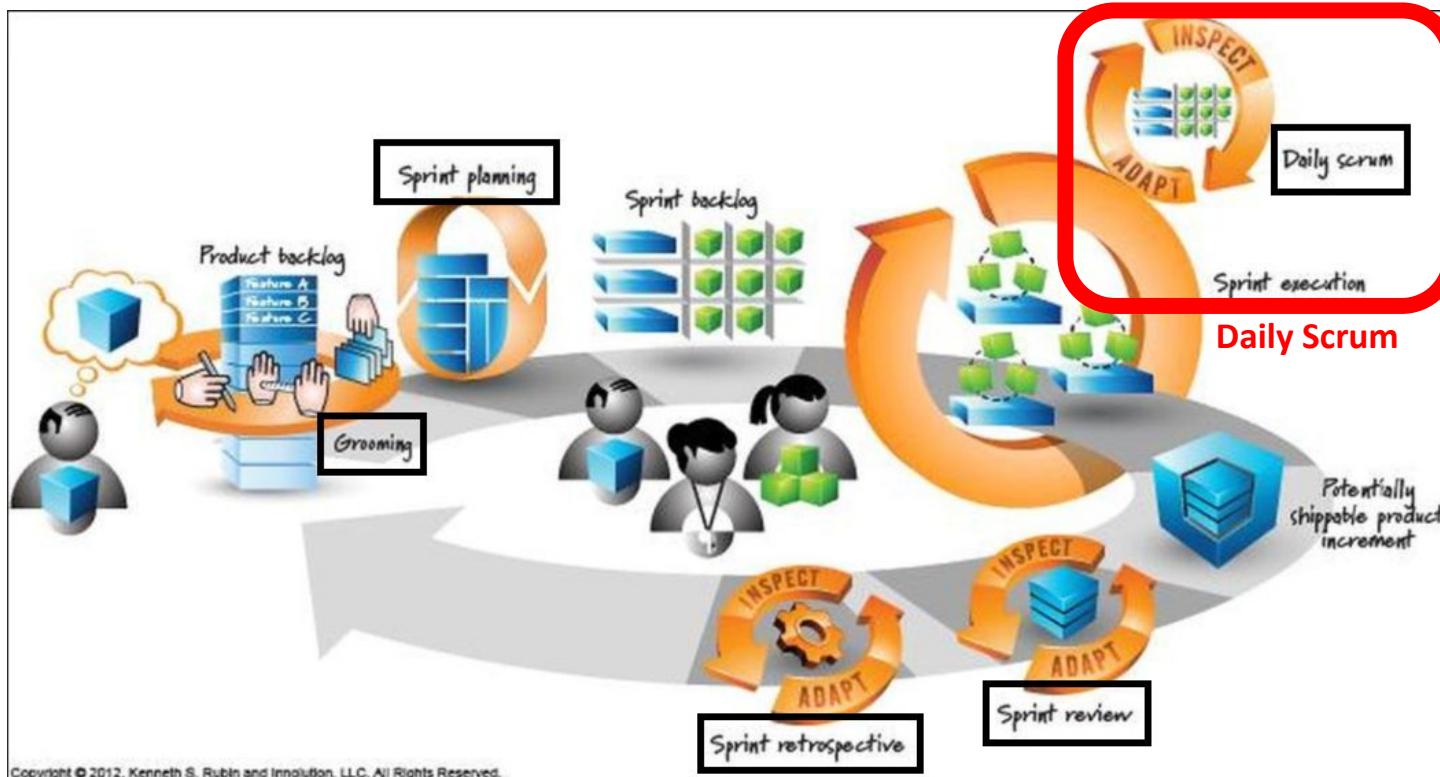
Success Factors

- Highly Engaging
- Team Alignment
- Clear desired outcome

Pitfalls

- Ungroomed user Stories
- Wishful Thinking
- Undue Influence from senior team members

Daily Scrum



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

Daily Scrum

Purpose and Context

- Helps Team coordinate their actions in detail
- Allows micro-planning and emergent discoveries
- Daily team commitment meeting (Tasks <1 day)
- Provides visibility into team's process

Level of Facilitation

- Varies based on team maturity (Shu, Ha, Ri), task oriented, outcome driven, time-boxed
- Facilitated by the Scrum Master and be rotated by members on the team

Outcomes / Results / Deliverables

- Team coordination
- Early identification of impediments
- Visibility for non- team members
- Team mutual accountability

Tools & Techniques

- Time Boxing
- Bottom lining
- Just Enough / Just In Time
- “Talking Stick”
- Working the Task Board

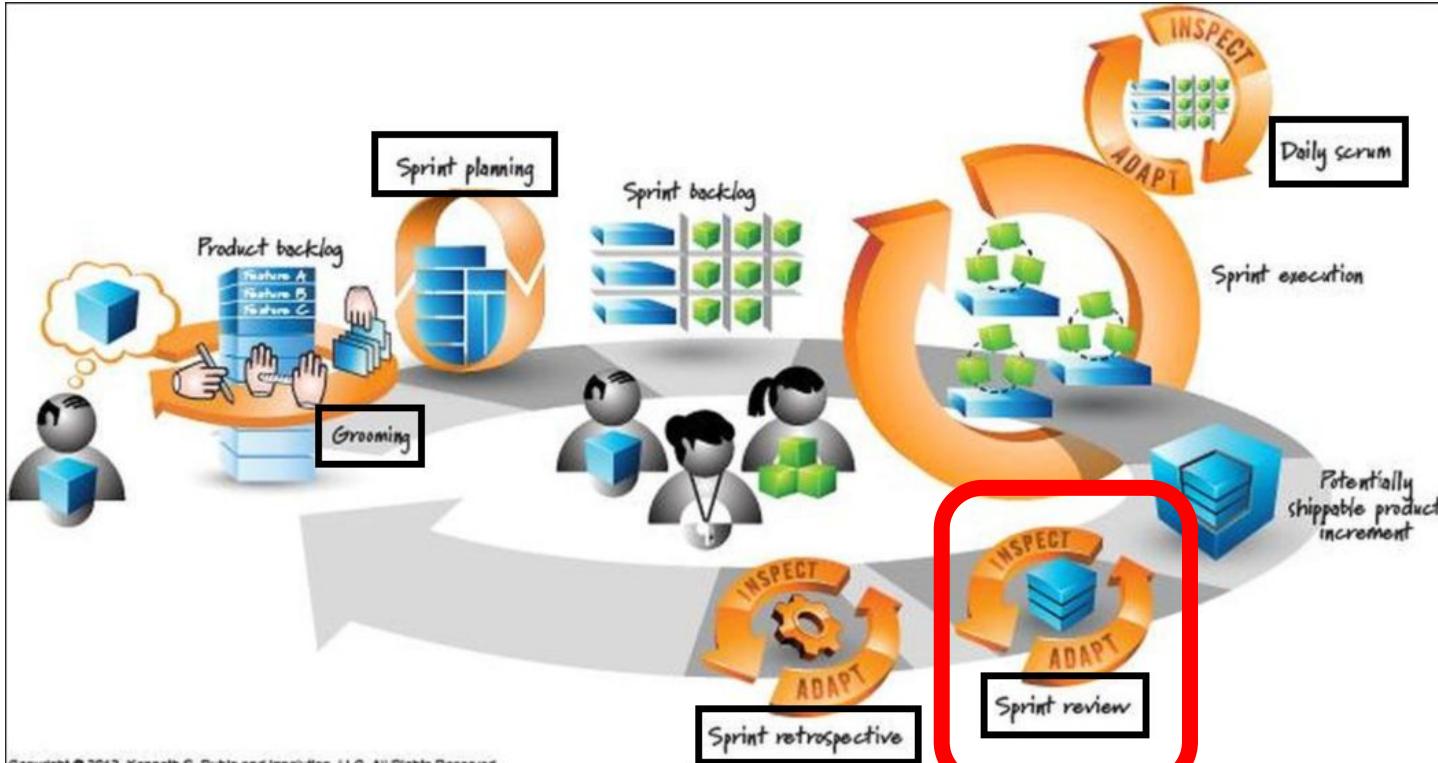
Success Factors

- Start on time
- Rigorous Time-boxing
- Commitment based conversation
- Full permission to interrupt

Pitfalls

- Going through the motions
- Digression into details
- Sitting down
- Entire Team not present

Sprint Review



Sprint Review

Sprint Review

Purpose and Context

- Review the goal of the sprint and compare to what actually got accomplished during sprint
- Demonstrate progress “show off” the product
- Real-time customer feedback from all stakeholders
- Allows management to give feedback & make requests
- Accept the iteration

Level of Facilitation

- Full-scale facilitation by Scrum Master until a team member wants to, and is capable of, facilitating; then facilitation can rotate.

Outcomes / Results / Deliverables

- | | |
|---|---|
| <ul style="list-style-type: none"> • Feedback on work completed vs goal • Demonstrate work completed • Shared understanding of value delivered | <ul style="list-style-type: none"> • Improved team morale • Team takes public responsibility for success/failure (bonds them) |
|---|---|

Tools & Techniques

- Time Boxing
- Bottom lining
- Information sharing
- Powerful Questions
- Listening

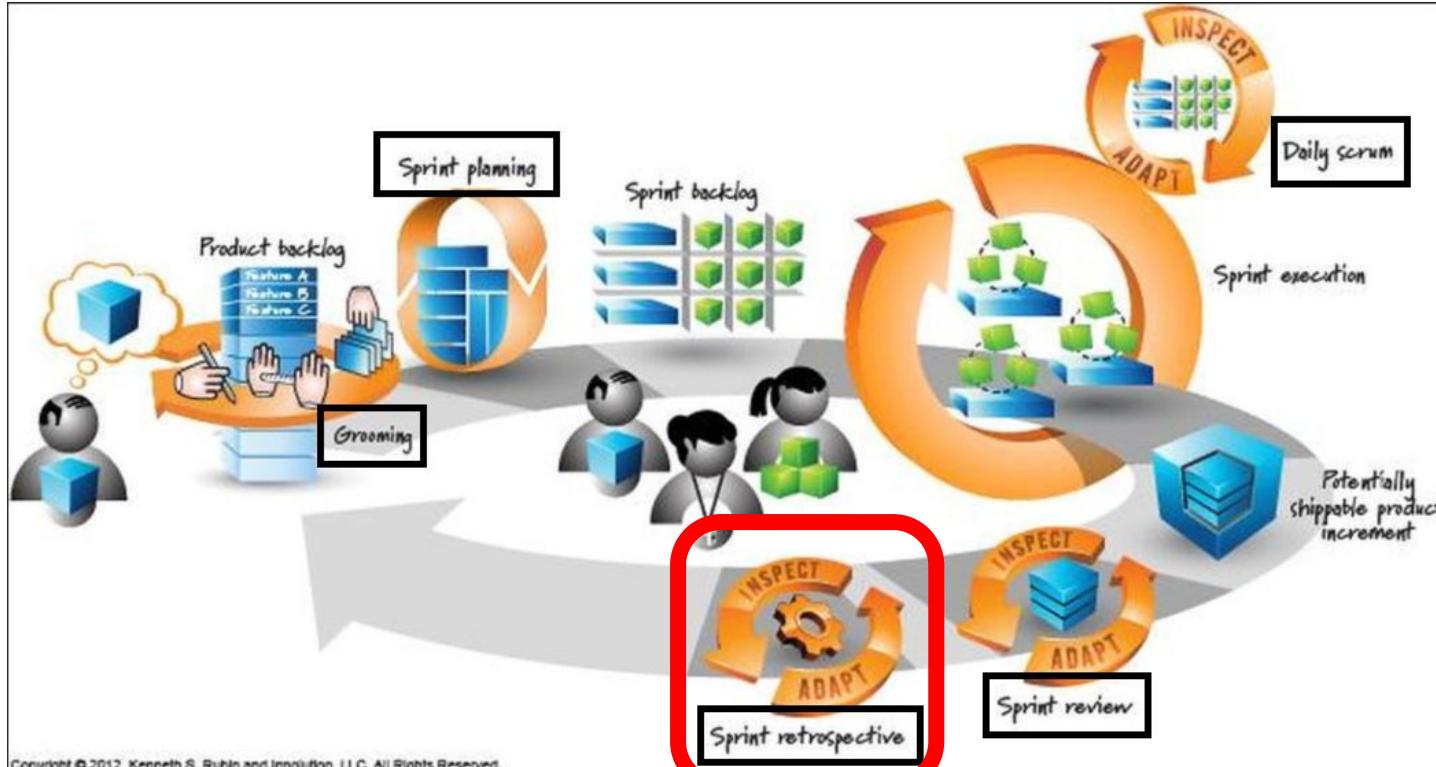
Success Factors

- Engaging & authentic
- “Real” working software
- Encourage conversation
- Transparency/Safety

Pitfalls

- Going through the motions
- PowerPoint
- Fear
- One way information flow
- Lack of interest from business stakeholders

Retrospective



Sprint Retrospective

Retrospective

Purpose and Context

- Primary focal point of the Inspect and Adapt cycle
- The way for the team to take responsibility for their process
- Focuses on three major areas: productivity, process & team

Level of Facilitation

- Full-scale facilitation owned by the Scrum Master until a team member wants to, and is capable of, facilitating; then facilitation can rotate

Outcomes / Results / Deliverables

- Greater team understanding of what actually happened
- Short list (e.g. 1 – 3) of “actions” to improve
- Inspect and Adapt regarding Productivity/process/team
- Team bonding (we’re in this together)

Tools & Techniques

- Steps from *Agile Retrospectives* Book
- Meeting is owned by the team, therefore, they may invite non-team members
- Prod Owner may attend IF they are member of team (rather than an external customer)
- Light and frequent follow up is crucial during sprint

Success Factors

- Planned ahead
- Skillfully facilitated
- Team interacts with the team directly
- All voices are heard

Pitfalls

- Same old – same old
- Everything “is outside the team’s control” (never completely true)

How do we adopt this?

How

Training

Coaching

Mentoring

Shu

Person follows rules
until they sink in

Ha

Person reflects on the
rules,, looks for
exceptions and
“breaks the rules

Ri

Rules are forgotten as
person has developed
mastery, and grasped the
essence and underlying
forces

Who

Student

Apprentice

Master

Agility – Stages of Learning

- 1. Shu (Following Precisely without Modification) – 2. Ha – 3. Ri



Agility – Stages of Learning

- 1. Shu - 2. Ha (Shifting Between Techniques) - 3. Ri



Agility – Stages of Learning

- 1. Shu - 2. Ha- 3. Ri (New Techniques, even unconsciously)



How do we adopt this?

How

Training

Coaching

Mentoring

Shu

Person follows rules
until they sink in

Ha

Person reflects on the
rules,, looks for
exceptions and
“breaks the rules”

Ri

Rules are forgotten as
person has developed
mastery, and grasped the
essence and underlying
forces

Who

Student

Apprentice

Master

Team and Scope

The Team

- Roles:
 - Product Manager (PdM)
 - Scrum Master + Team = Development Team
- All skills needed to implement
 - Generalists, versatile
 - May include specialists
- Egos are put aside
 - High Trust
 - Is it good enough? Not looking for perfecti..
- Empowered and Self Organized
 - Members accountable for planning and delivery as a team

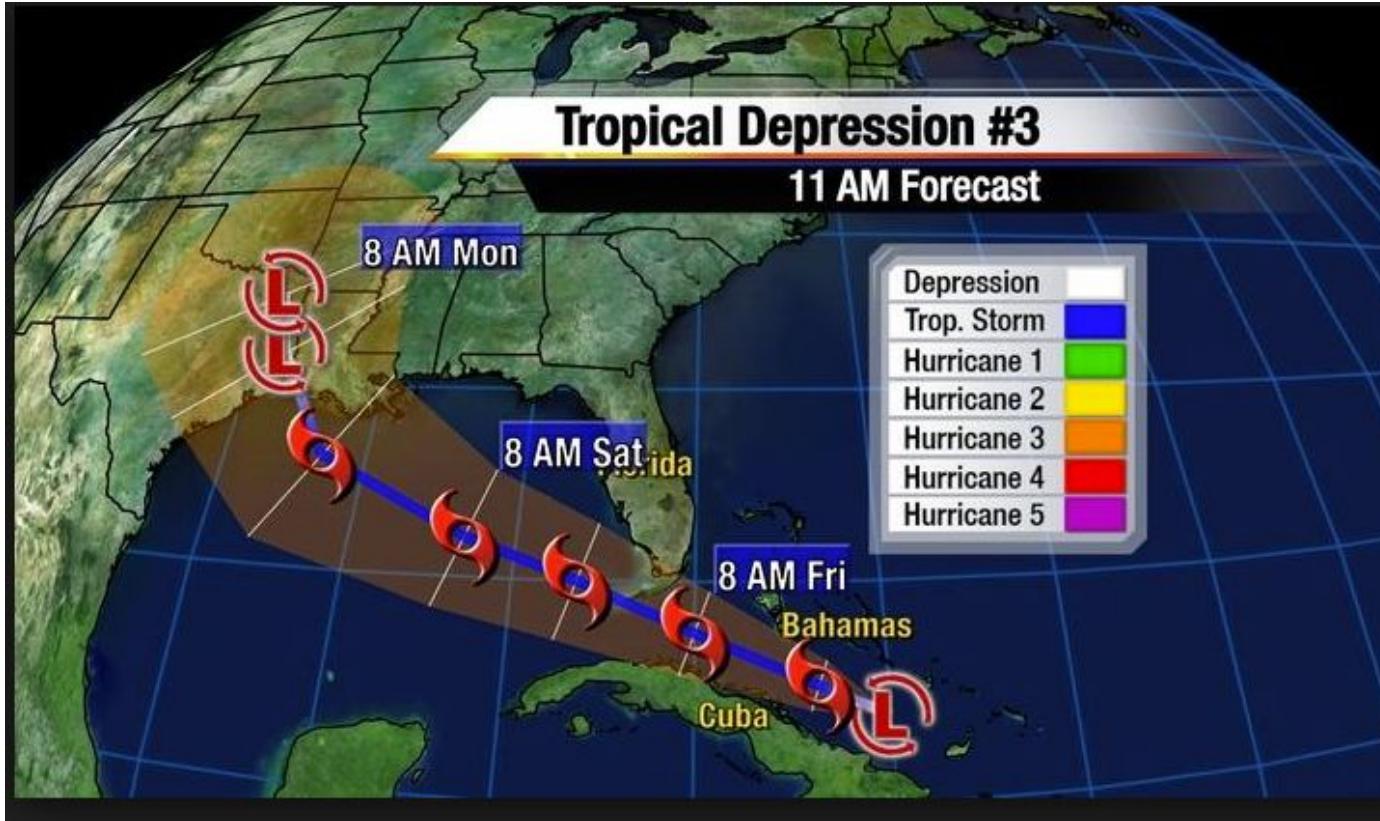


Team and Scope

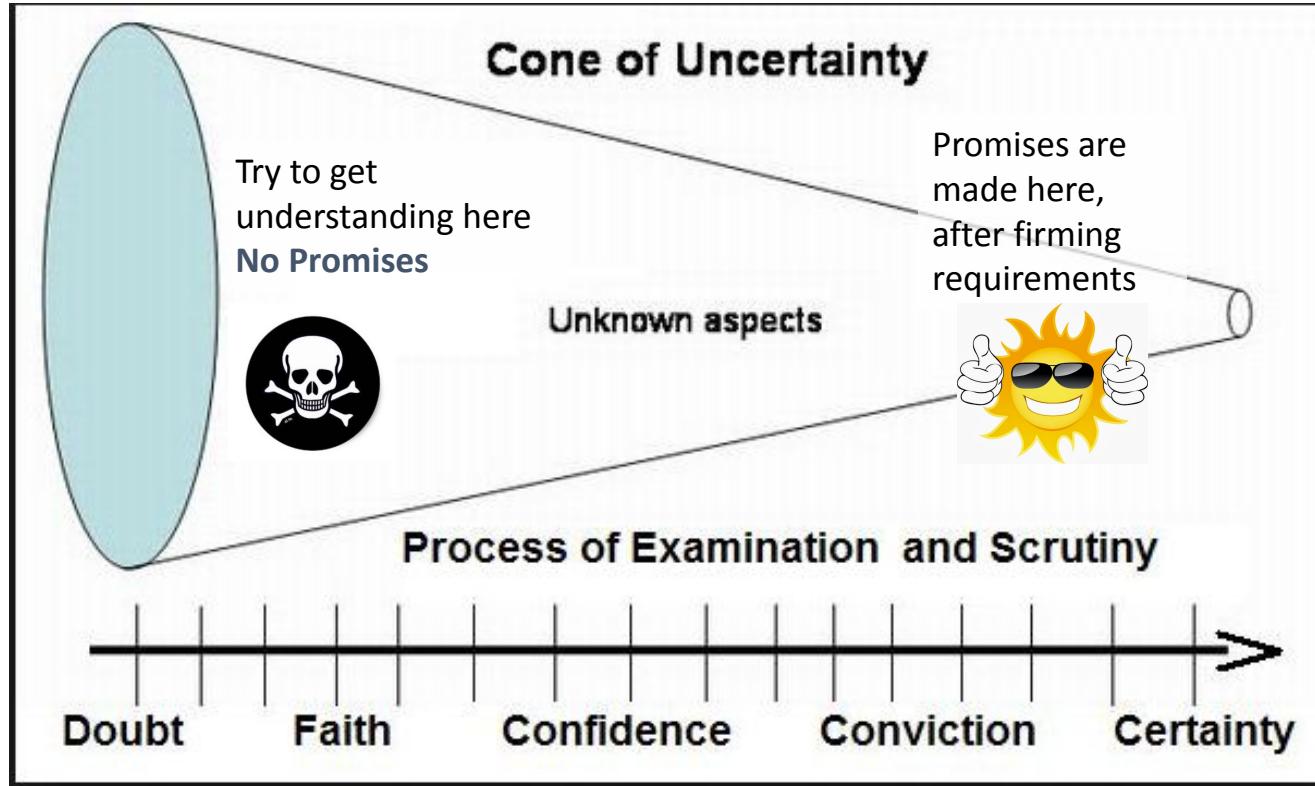
Team Roles

Product Manager	Scrum Master	Development
<ul style="list-style-type: none"> • Owns Product vision • Defines features, decides on release date and content • Responsible for market success • Prioritizes features according to market value • Can change features and priorities every Sprint • Advocate for Business 	<ul style="list-style-type: none"> • Responsible for facilitating process • Focuses Team and protects them from external interruption • Stabilizes Current Sprint • Looks for ways to enhance productivity • Assists Product Owner in leveraging Scrum • Advocate for the Engineer 	<ul style="list-style-type: none"> • Small group containing all necessary people and project skills • Focuses on steady delivery of high quality features • Generates options for delivery • Manages own work within Sprints

Cone of Uncertainty

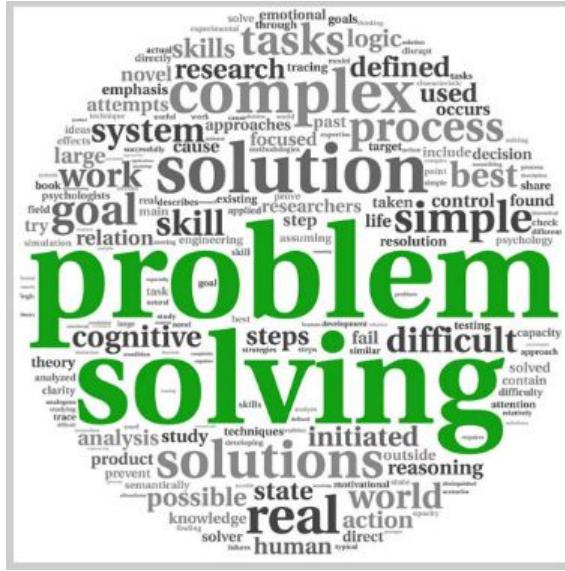


Cone of Uncertainty



User Stories

Helping to solve the Business Problem...



Start with an Epic...



Story Writing

- A user story is not a specification, it is a communication and collaboration tool to help explain the why.
 - *Ron Jefferies - A Card, Conversation and Confirmation.*
- The Product Owner should not hand stories off to the development team:
 - They should rather be part of a conversation together or better yet write them as a team.
 - Share and Leverage each others knowledge and will result in better user stories.



Structure of a Story INVEST

Independent

Can deliver value by itself. We want to be able to develop in any sequence.

Negotiable

Avoid too much detail; keep them flexible so the team can adjust how much of the story to implement.

Valuable

Users or customers get some value from the story.

Estimatable

The team must be able to use them for planning

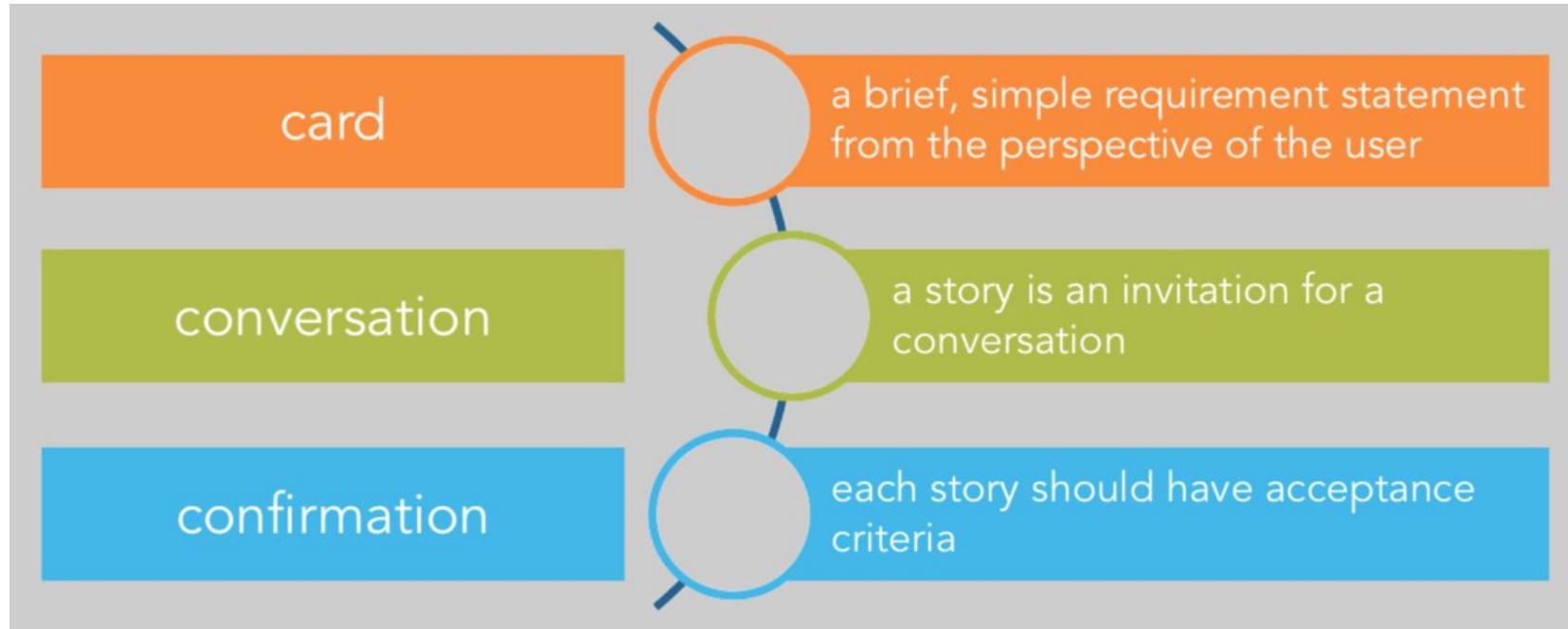
Small

Large stories are harder to estimate and plan. By the time of sprint planning, the story should be able to be designed, coded and tested within the sprint.

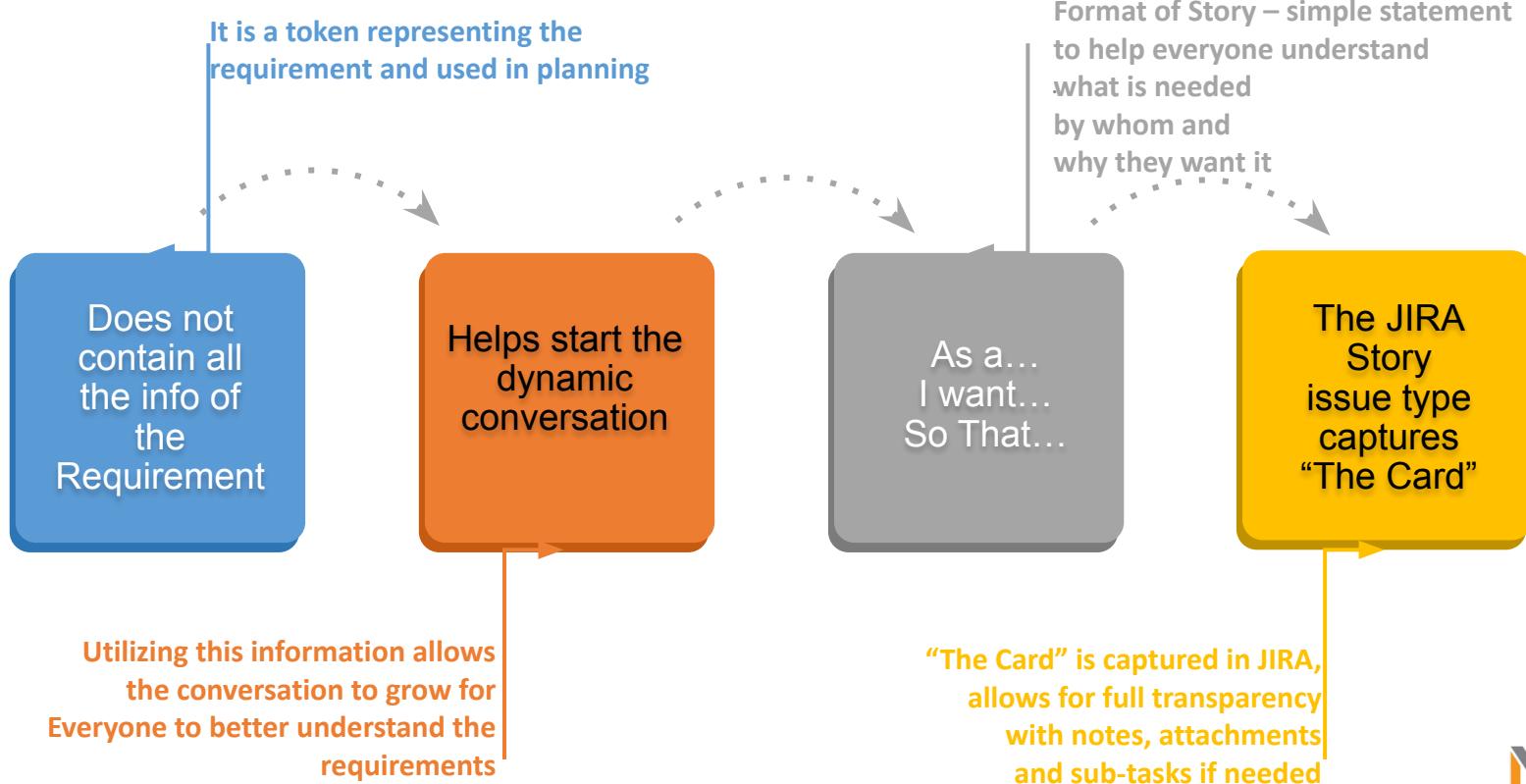
Testable

Document acceptance criteria, or the definition of done for the story which leads to test cases

Structure of a Story



The Card



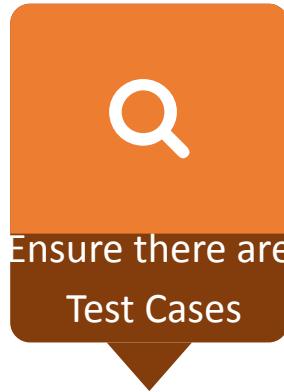
The Conversation



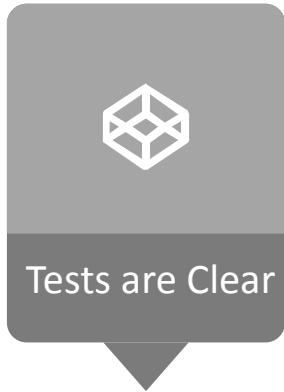
The Confirmation



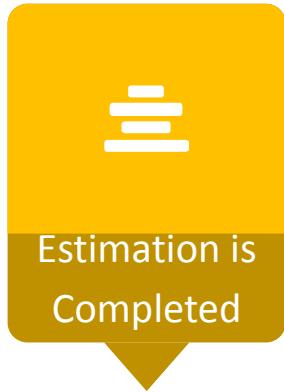
Story Meets
INVEST



Ensure there are
Test Cases



Tests are Clear



Estimation is
Completed



Agreement
by All

The story needs to meet the INVEST Criteria that was discussed earlier in the presentation

Having Test Cases will allow the team to understand what needs to be completed and how will everyone understand how they will validate this.

By having clear tests will allow the Developer Test and Customer or User perform UAT

Estimation will allow the team to ensure they understand the size, complexity and effort that is being asked of them. In addition it will allow to see the Burndown and also Velocity of the Team

Agreement by everyone on the team on what will be completed within the sprint will allow everyone to be on the same page.

Story Overview



Story Overview cont.

1st Feedback from the field

“Customers would like the shredder to have a power switch.”



Story Overview cont.

1st Feedback from the field

As a user, I want a power switch so I can turn the shredder off and on.



Story Overview cont.

2nd Feedback from the field

Customers love the power switch but they would like it on the back of the shredder so it cannot be seen from the front.



Story Overview cont.

2nd Feedback from the field

As a user, I want the power switch on the back so it cannot be seen from the front.



Story Overview cont.

3rd Feedback from the field

Customers love the new switch location, but they would like a light on the front that indicates if the shredder is powered on.



Story Overview cont.

3rd Feedback from the field

As a user, I want an indicator light so I know if the shredder is on.



Story Overview cont.

Why does the customer want . . .

- A power switch?
- The power switch on the back of the shredder?
- A power indicator light?



Story Overview cont.



Story Overview cont.

Original user story

As a user, I want a power switch so I can turn the shredder off and on.



Story Overview cont.

Better Story

As a parent, I want a shredder on which my toddler will not accidentally hurt herself so that I do not have to remember to unplug it from the wall when not in use.

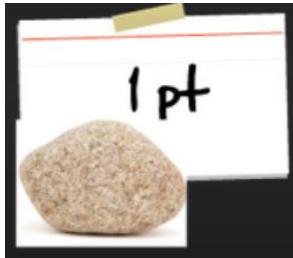


Estimating

Agile Estimation (Fibonacci) and Planning

- Fibonacci Numbers
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,
 - We use the following numbers in Agile
 - 0, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?

- Let's look at the following series
- | | | | | | | | | | | |
|---|---|---|---|---|---|----|----|----|-----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 1 | 2 | 3 | 5 | 8 | 13 | 20 | 40 | 100 | ? |



Backlogs

Product Backlog is DEEP

Goal - Groom user stories in the backlog for future sprints - “story time” (get them to a “ready” state)

Detailed Appropriately 

 **Estimated**

Emergent 

 **Prioritized**

Detailed Appropriately

More detail is known about items that will be done sooner and are ordered higher in the backlog

Emergent

Not everything is known upfront. We acknowledge that change happens, so planning is continual

Estimated

Some form of estimate has been given for each product backlog item before a team commits to implement

Prioritized

The backlog is prioritized, so that stories are ordered by value to the business continually as learning progresses

Definition of Ready and Done



Example - Definition of Ready

Definition of Ready (DOR) – checked prior to starting a story in a sprint

Description and Acceptance Criteria are clearly defined

UX Documentation has been reviewed and approved by Team

All functional requirements, non-functional requirements, documents, designs and use cases have been included or linked in ticket

Dependencies are identified and documented or linked in ticket

Ticket must be 100% filled out where required by JIRA

This is suggested DoR.

Each Team needs to define their own DoR.



Definition of Done



Example - Definition of Done

Example – Definition of Done

Product Owner accepts the delivered story and signs off

Example Definition of Done (DOD)

The story meets all listed Acceptance Criteria

The story visually matches UX documentation (if applicable)

The code for the story is reviewed, suggestions incorporated and checked in

Unit tests and test cases are written and executed successfully

The code has been merged to master, and pushed to source control

Product owner has reviewed the story to validate that it meets the Acceptance Criteria

This is suggested DoD.

Each Team needs to define their own DoD.



Scrum Framework - Artifacts

Product Backlog

All work items related to a product/project, ordered by a Product Owner.

Burndown Chart

Visual aid for tracking team progress and forecasting expected completion dates.

Sprint Report

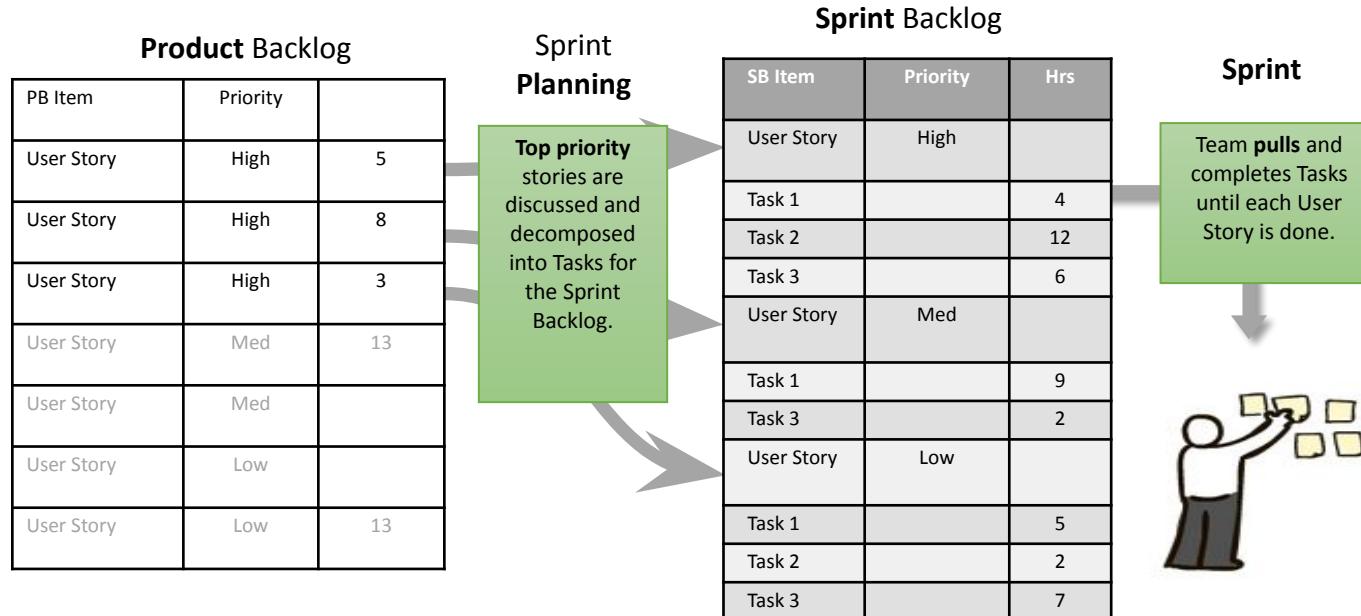
The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks..

Velocity Chart

Tracks rate of completion for team. Shows the sum of promises per sprint vs. the work that was delivered

Product and Sprint Backlog

- Priority is looked at from the Product Backlog and moved to the Sprint Backlog
- Some teams look at hours at the task level, others only look at Story Points of stories



How to read Burndown Chart

- Visual aid for tracking team progress and forecasting expected completion dates.
- Burndown Charts can provide context to make tough decisions for sprints



Backlog

Story 1
Story 2
Story 3
Story 4
Story 5
Story 6
Story 7
Story 8
Story 9
Story 10

Product Owner Speaking

To date, we have completed Story 1 through Story 4.

Unfortunately, we lost several key members of our team during iteration 6 and **we are unlikely to get all planned Stories done for this release**, unless we execute with perfection.

We will likely **delay Story 9 and 10 until the next release**, unless we make some tradeoffs.

We already started discussions with sales and marketing and we may limit our work on Story 5 and 6 in the next Sprint.

JIRA Sprint Report

The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks.

ATT_Scrum
Sprint Report -

ATT 7/19 - 7/31 It 62 ▾ [How to read this chart](#)

Active Sprint 27/Aug/18 8:56 PM - 10/Sep/18 8:56 PM [Linked pages](#)

[View ATT 7/19 - 7/31 It 62 in Issue Navigator](#)

Status Report

* Issue added to sprint after start time

[View in Issue Navigator](#)

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (3)
ATT-89	Pick a theme for the party	Story	<input type="checkbox"/> None	DONE	3

[View in Issue Navigator](#)

Issues Not Completed

Key	Summary	Issue Type	Priority	Status	Story Points (5)
ATT-88	Delegate Responsibilities	Story	<input type="checkbox"/> None	BACKLOG	3
ATT-183 *	Determine Theme of Party	Story	<input type="checkbox"/> None	BACKLOG	2

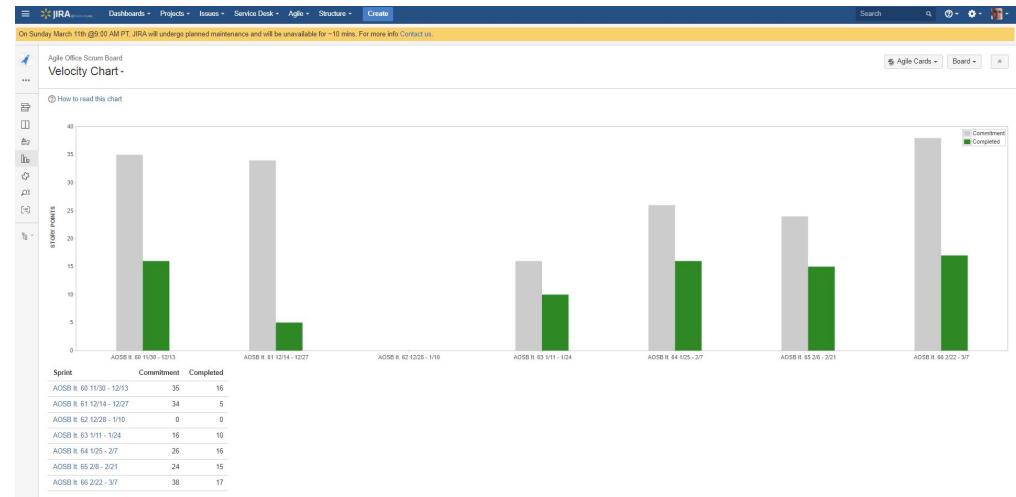
[View in Issue Navigator](#)

Issues Removed From Sprint

Key	Summary	Issue Type	Priority	Status	Story Points (5)
ATT-32	Determine core group of people	Story	<input type="checkbox"/> None	BACKLOG	5

Using Velocity for Planning

- Velocity is a measure of how much work can be delivered in a timebox
- It is based on the historical performance of the team for recent sprints
- We use established velocity for planning sprints and for use in estimating releases



Retrospective

Retrospectives = Focus?



Retrospective – Purpose and When

The PAST is where you
learned the lesson

The FUTURE is where you
apply the lesson,

Don't GIVE UP
in the middle!

CALENDAR

					1	2	3
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	

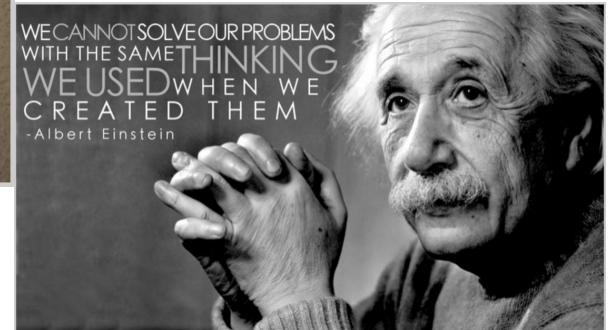
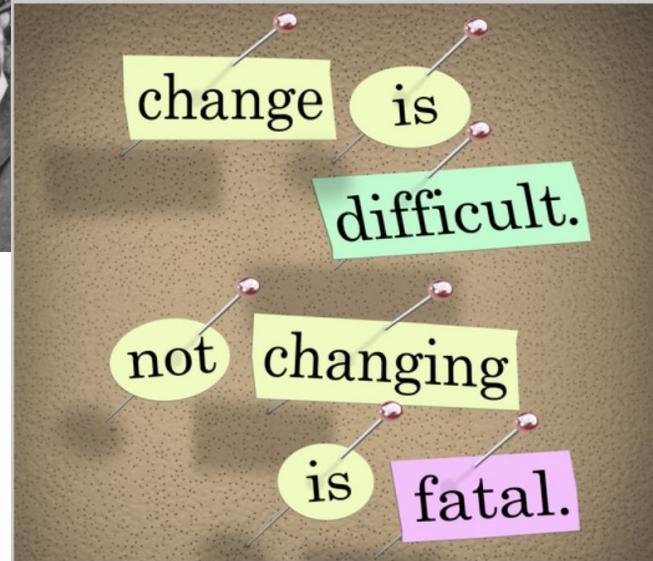
Why do Retrospectives

**"You may never
know what results
come from
your action.
But if you do
nothing,
there will be no
result."**

Mohandas Karamchand Gandhi

happytoinspire

Mindset during Retrospective

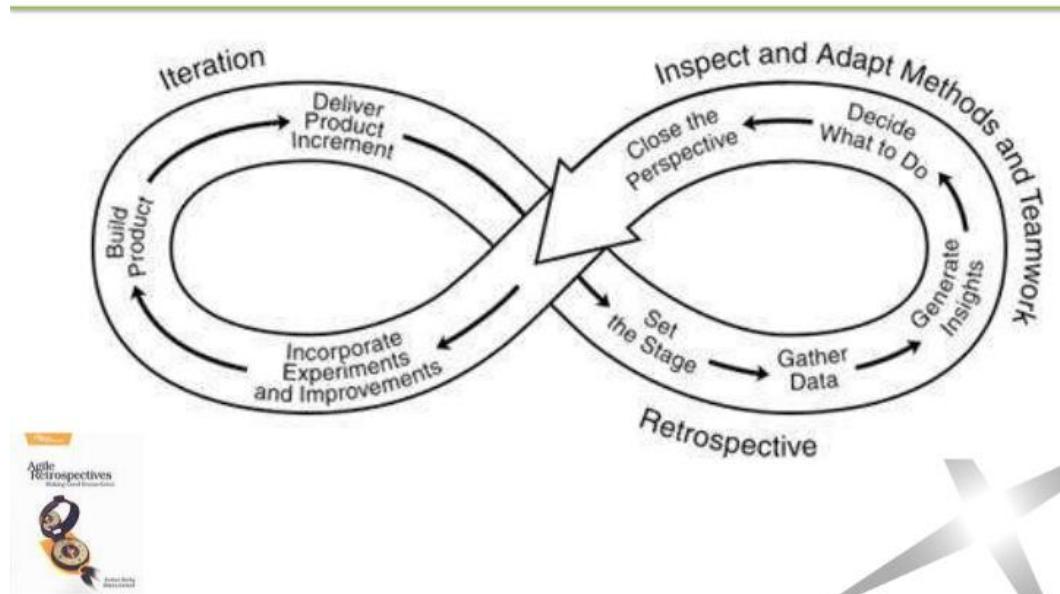


Tip prior to Retrospective



Making good teams Great!!!

Why Retrospect



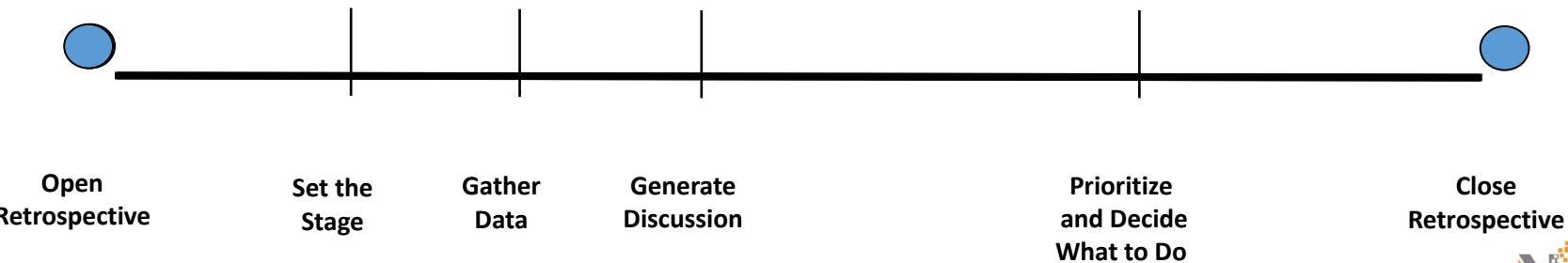
Credit: Agile Retrospectives – Making good teams Great by Esther Derby & Diana Larsen

<http://pragprog.com/book/dlret/agile-retrospectives>

AgileSparks

Structuring the Retrospective – 6 points

- Open Retrospective
 - Set the Stage
 - Gather Data
 - Generate Discussion
 - Prioritize and Decide What to Do
 - Close Retrospective
-
- Below is timeline of meeting:



Gathering Data – Learning Matrix



Happy Face – aspects you liked
should be repeated



Sad Face – aspects you disliked –
should be changed

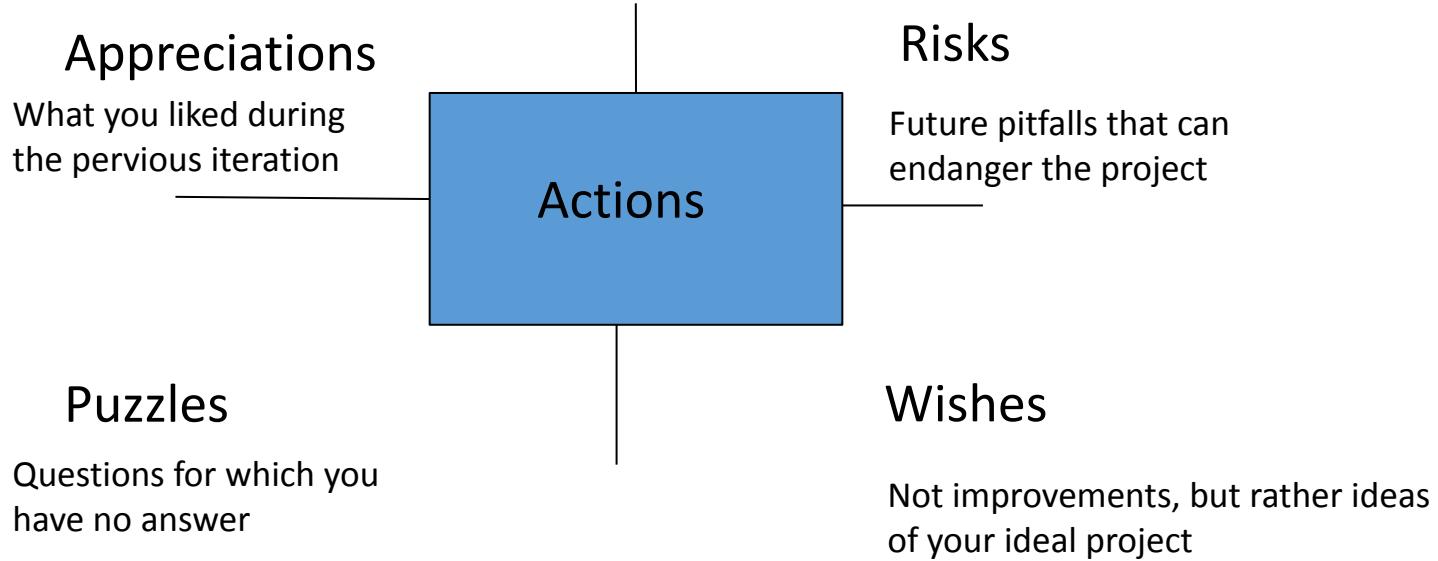


Flower bouquet for people
you wish to appreciate

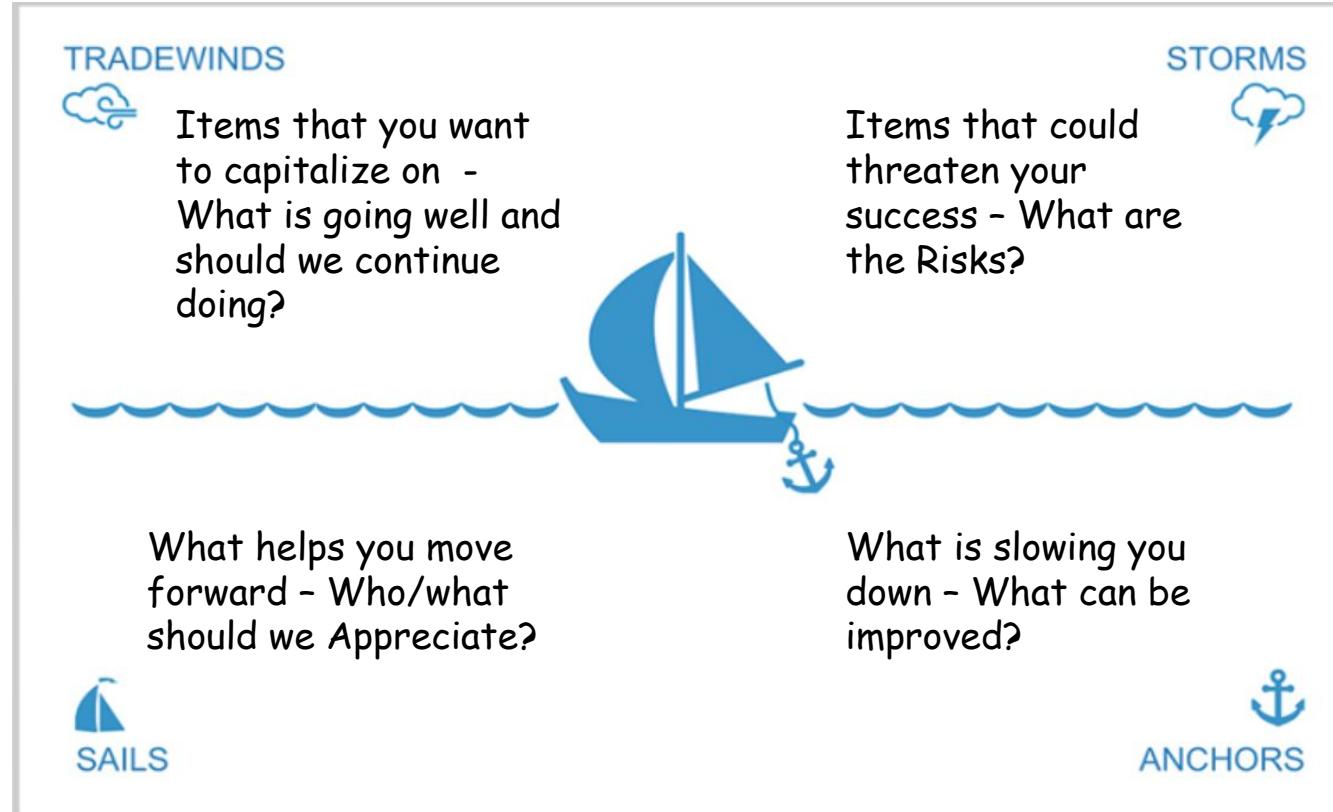


Light bulb for new ideas to try

Gathering Data - Actions



Gathering Data - Sailboat



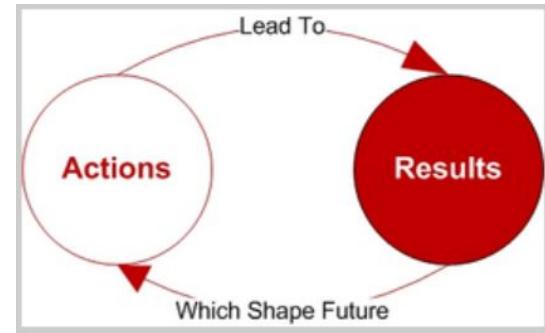
Gathering Data – Start, Stop, Continue



- Simplest retrospective method – very effective
- Reflect on three things
- Facilitator has team focus on each question separately in 3 short “rounds”
- Team votes on specific items to focus on during the coming sprint
- Review these items at the beginning of next retrospective

Prioritize and Close Retrospective

Reiterate actions and follow-up



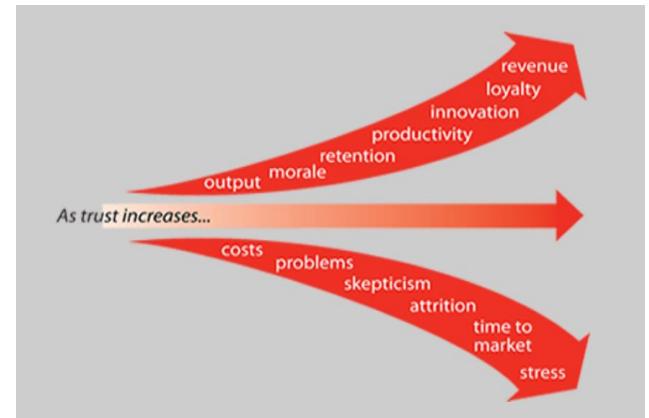
Appreciate contributions



Identify ways to make the next retrospective better

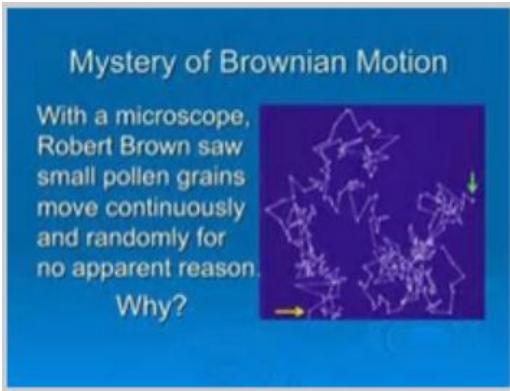


Benefits of doing Retrospective



Leave out...what happens

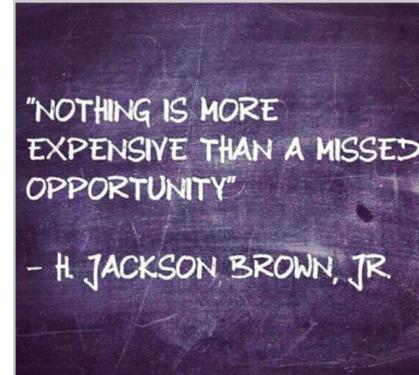
Setting the Stage:



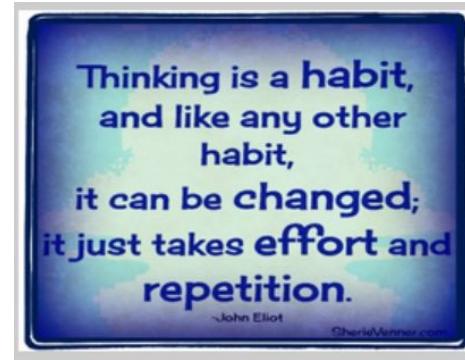
Gathering Data:



Closing the Retrospective:



Generating Discussion:



Deciding What to Do:



EXIT NOW

How do we adopt this?

How

Training

Coaching

Mentoring

Shu

Person follows rules
until they sink in

Ha

Person reflects on the
rules,, looks for
exceptions and
“breaks the rules”

Ri

Rules are forgotten as
person has developed
mastery, and grasped the
essence and underlying
forces

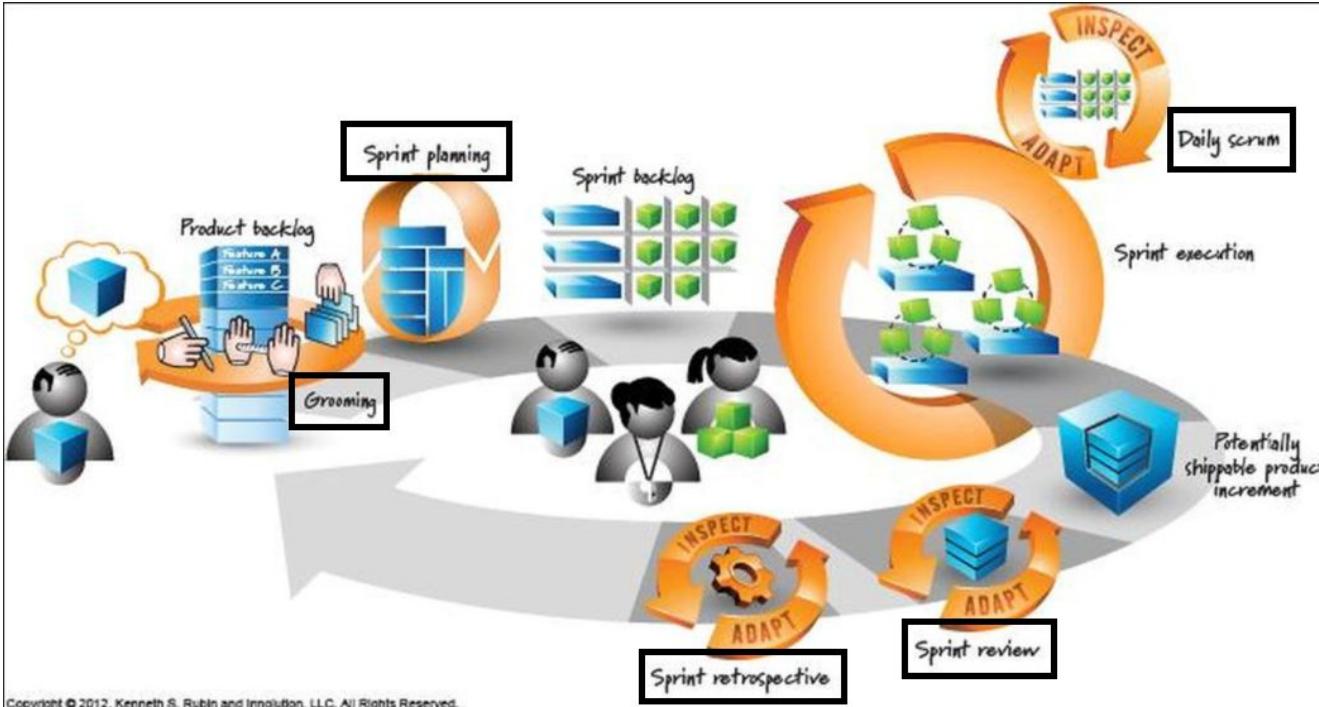
Who

Student

Apprentice

Master

Better Understanding - Scrum Framework



Thanks

Thanks

Quiz

0070

In the Agile Manifesto, what is valued more than Processes and Tools?

- a. Customer Collaboration
- b. Individuals and interactions
- c. Working Software
- d. Responding to Change

Quiz

In the agile manifesto, what is valued more than Contract Negotiation?

- A. Responding to change
- B. Working software
- C. Customer collaboration
- D. Individuals and interactions

Quiz

Question 3:

Fill in two words that are missing from one of the Agile Manifesto's core values

Through this work we have come to value: _____ over following a plan

- A. Responding to change
- B. Working software
- C. Building great product
- D. Satisfying our customer

Quiz

Question 4:

According to the Agile Principles, what is the highest priority?

- A. Work Together Daily
- B. Attention to Detail
- C. Deliver Value faster
- D. Face-to-Face Conversations

Quiz

Question 5:

How often do the agile principles recommend to deliver working software?

- A. Every week
- B. Every year
- C. A couple of weeks to a couple of months with preference to a shorter timescale
- D. Daily

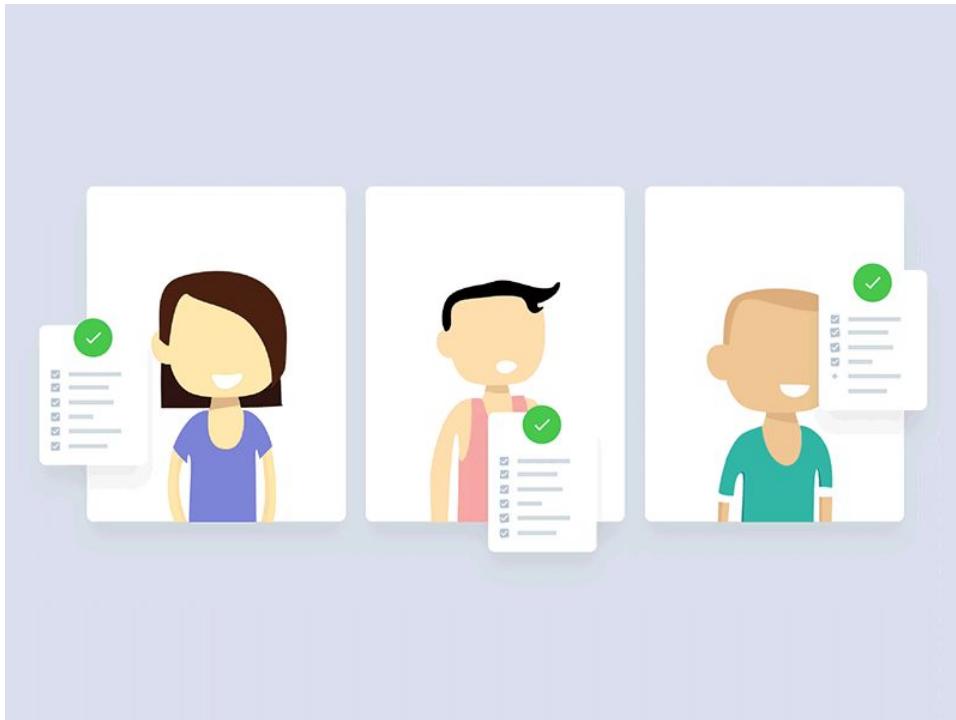
INTRODUCING USER STORIES

WHAT ARE USER STORIES AND WHY ARE THEY USED?

A user story is a small unit of work told from a user's perspective of the person who desires a new functionality.

Explains desired functionality

I believe they are a crucial part of Agile product development, because it shifts the focus from writing requirements to talking about them.



USER STORY FORMAT

PARTS OF A USER STORY

[WHO] As a <type of user>

[WHAT] I would like <goal>

[WHY] So that I <reason>

EXAMPLE 1

As an online student
I would like the ability
to message the instructor directly
So that I can get
one-on-one help for any questions
or advice that I need.

EXAMPLE 3

As an online student
I would like when I close
the video for my spot to save
So that I when I come back
to the video I can continue
from where I left off.

USER STORY FROM AN ONLINE INSTRUCTOR PERSPECTIVE

As an online instructor
I would like the ability to edit lectures
So that I can keep the course accurate and up to date

WHO CREATES THE USER STORY?

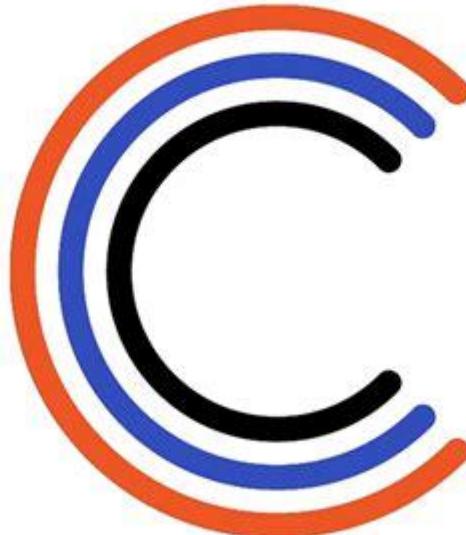


3 C'S OF WRITING USER STORIES

Card

Conversation

Confirmation



WRITE YOUR FIRST USER STORY! - ACTIVITY

0120

DIFFERENCES BETWEEN USER STORY AND A REQUIREMENT

1. Perspective (System versus User)
2. Where these are used (non-agile projects like waterfall or spiral whereas user stories are used in agile projects)
3. How requirements and user stories are drawn out
4. Who owns it?
- 5.

USE CASE VERSUS USER STORY

User Stories: Basic Needs (small scope - one specific user need)

Use Cases: Basic Behaviors (define interactions with the system and are larger scope)

ACTIVITY

ACTIVITY - Use cases, user story, and requirements worksheet

Many terms utilized in business analysis can easily be confused and incorrectly used. This activity will help you to ensure you have a good understanding of these commonly misunderstood terms; **user story**, **use case**, and **requirements**

USER ROLES

USER PERSONAS

ACTIVITY

ACTIVITY - User Roles and Persona activity

Let's put this new knowledge into practice! Please complete this User Roles and Persona activity. 0180

ACTIVITY

ACTIVITY – Iterating on User Stories with new Roles and Personas

Now that you've gone through the exercise to define the user roles and personas, it is always a good idea to review the User Stories you've already written. If you find the <user> is too vague or doesn't provide value in its current state, utilize the identified roles and personas to help you iterate and update the User Story.

Instructions: Walkthrough each of your previously created User Stories, paying close attention to the <user> in each. When you find a User Story that would benefit from an update, use the outputs from the User Role and Persona Activity to help you make it more descriptive and valuable.

A Lesson to Remember: Any time circumstances change or concepts are further defined, it is a good idea to review your User Stories and update them with the newly acquired information.

SUMMARY

ACCEPTANCE CRITERIA

UNDERSTAND ACCEPTANCE CRITERIA

WRITING ACCEPTANCE CRITERIA

WHO AND WHEN TO WRITE ACCEPTANCE CRITERIA

SUMMARY

ACTIVITY

ACTIVITY - Adding acceptance criteria to your user stories

Now that you have a base understanding of Acceptance Criteria, it is time to add more details to the User Stories you have created.

Instructions: Walkthrough each of your previously created User Stories and add details as Acceptance Criteria. Remember, this is an iterative process. If you are stuck on one of them, brainstorm about the questions you would need to have answered and the details necessary to solve that particular problem.

Once you have completed a few iterations of adding and adjusting Acceptance Criteria, take a look at the included Answer Key. This will help you to fully grasp the concepts and potential options.