ACE 203

Sprint Planning, Scrum, Demos, and Retros

Agile Center of Excellence

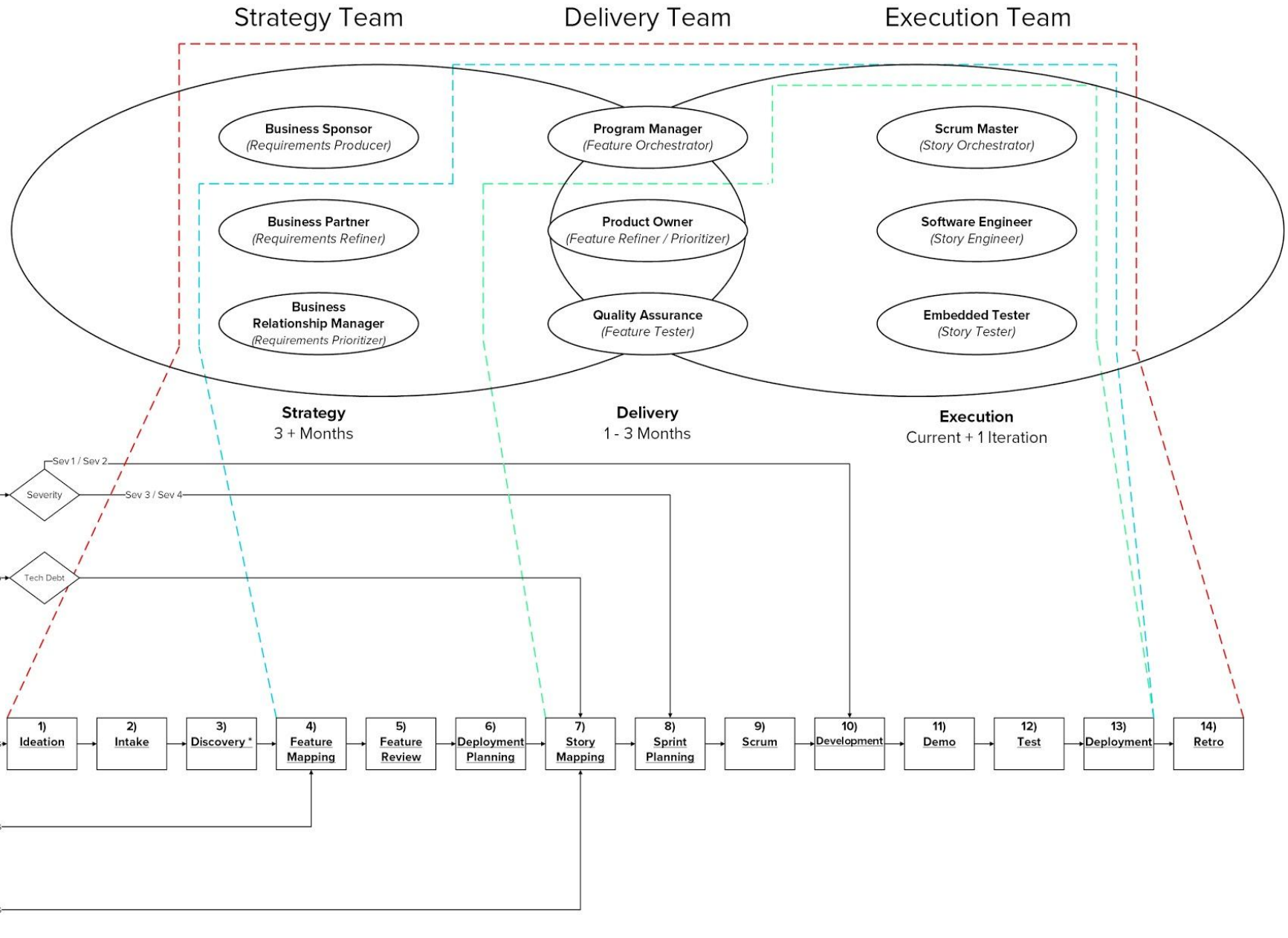
v_1.0

Overview

- Introduction
- Agenda
  - Sprint Planning
  - Scrum
  - Demo
  - Retro

Software Delivery Framework

The type, size, and complexity of a request will determine who is impacted and what steps in the process need to be followed



Legend

New Initiative
- Impact 1+ Product
- Impact 1+ Team
- Deployed in 1+ Milestone

Existing Initiative
New Feature
- Impact 1+ Team
- Deployed in 1 Milestone
- >300 Hours of Development

Existing Initiative
Existing Feature
New Story
- Impact 1 Team
- Deployed in 1 Milestone
- >20 Hours of Development

Process Steps by Project Phase

1) New Projects created in Ideation
2) Impacted Teams identified at Intake
3) Requirements identified in Discovery
4) Features defined in Feature Mapping
5) Feature Scope accepted in Feature Review
6) Incremental Feature delivery priority finalized at Production Increment Planning
7) Required Stories identified to meet Feature Acceptance Criteria at Story Mapping
8) Execution Team commits to top priority Stories based on capacity at Sprint Planning
9) Review of what has been completed, what is in progress, and any blocking issues
10) Execution Teams completes committed Stories during Development
11) Features are presented for approval at Demo
12) Features pass TQA / UAT / DIT testing during Test
13) Code Deployed to Production
14) Teams utilize Retros for continuous improvement

Glossary

- **Acceptance Criteria** - Specific conditions, with clearly defined pass / fail results, that must be met, before a Feature can be Accepted
- **Artifact -** Theme, Initiative, Feature, User Story, or Task
- **Assumption** (*a.k.a. Pre-Condition)* - Specific criteria that must be met before the end user is eligible to achieve the desired outcome identified in the Requirements or Acceptance Criteria
- **Backlog** - Stack-ranked, prioritized list of work items to be completed
  - **Portfolio Backlog** - Project / Initiative artifacts prioritized by the Business Sponsor or Business Partner
  - **Product Backlog** - Feature artifacts prioritized by the Product Owner
  - **Team Backlog** - User Story artifacts prioritized by the Scrum Master
  - **Iteration Backlog** - User Story artifacts for a specific team, for a specific Iteration, prioritized by the Scrum Master
- **Bug** - Error found in the Development environment
- **Business Case** - Issue or opportunity to be solved and expected outcome the solution will provide
- **Capacity** - Maximum amount of work a team is capable of delivering in a given time period
- **Customer Journey Mapping** - Exercise of visually documenting the process a customer goes through to achieve the desired outcome specified in the Requirements or Acceptance Criteria
- **Cycle Time** - Time between when work begins on an Artifact and when work is complete on an Artifact
- **Defect** - Error found in the Test environment
- **Definition of Done** - Regardless of the specific Artifact, what must be completed for any Artifact of that type to be considered 'Done'
- **Dependencies** - Completion of one Artifact is reliant upon the completion of another Artifact
- **Deployment** - Artifact has met the Definition of Done and is deployed into the Production environment

* Recommended if appropriate

Glossary

- **Dev-Ready** - Both the Business and IT agree upon the Value Statement, Scope, and Acceptance Criteria of an Artifact and there is sufficient detail for development to begin
- **Feature** - Independently deployable functionality that satisfies the Definition of Done for the Requirement
- **Impacted Teams** - Individual teams that will be required to complete work in order to meet the Definition of Done for the Artifact
- **Incident** - Error found in Production
- **Initiative** - Group of one or many Features that collectively deliver a specific outcome for the end user
- **Iteration** *(a.k.a. - Sprint)* - Pre-defined consistent time box, during which Impacted Teams develop and deliver User Stories, typically between 1 - 3 weeks
- **Lead Time** - Time between when an Artifact is created and when work is complete on an Artifact
- **Level of Effort** *(LoE)* - Estimated time (+/- 25%) required to deliver an Artifact
- **Milestone** - Used to track the completion of a set of functionality
- **Release** - Pre-defined consistent time box, during which Impacted Teams develop and deliver Features, typically between 1 month - 1 quarter
- **Requirement** - Desired outcome, output, or functionality requested by the Business on behalf of the End User
- **Return on Investment** (*RoI*) - Benefit *(Sales / Savings)* divided by the Expense *(Labor / Resources)* required to deliver an Artifact
- **Roadmap** - Strategic plan that identifies the key priorities major milestones required to deliver specified outcomes
    - **Portfolio Roadmap** - 6 - 18 month plan that identifies critical Business outcomes, typically at the Project or Initiative level
    - **Product Roadmap** - 1 - 6 month plan that identifies critical Product outcomes, typically at the Feature level

* Recommended if appropriate

Glossary

- **Spike** - A type of User Story used to track the work to research a solution to a Feature or Story when the Impacted Team does not know how to meet the Definition of Done
- **Story Point** - Abstract metric used to measure the complexity, risk, and time required to deliver a User Story. Is used as the baseline to determine a team's Capacity and Velocity. Modified Fibonacci Sequence is one of the more popular methods
- **Task** - Smallest unit of work, owned by an individual, and typically no smaller than 2 hours and no larger than 4 hours
- **Technical Debt** - Rework incurred for the future due to implementation of the easier, but not the best solution in the present
- **Test -** Ensure the execution of the Artifact satisfies the Acceptance Criteria and Steps to Test
  - **Automated** - Test cases are executed automatically with the assistance of a software tool
  - **Manual** - Tests cases executed manually by an individual
  - **Regression** - Ensure that deployment of new Feature does not negatively impact existing functionality
  - **Functional** - Ensure the deployment of new Feature satisfies the functional specifications of the Feature
- **User Story** - Unit of work for a specific Impacted Team that satisfies specific Acceptance Criteria within a Feature, completed within one Iteration

* Recommended if appropriate

Glossary

- **Value** - Objective measure used to prioritize an Artifact based on:
  - Business Rank
  - Return on Investment
  - Dependencies
  - Legal Requirements
  - Market Conditions
  - Other Factors
- **Value Statement** - Describes the intended outcome or value received by the end user as a result of a specific action
  - As a _____*(End User)*
  - I want to _____*(Action Performed)*
  - In order to _____*(Outcome / Value Received)*
- **Velocity** - Rolling average number of Artifacts delivered over a given number of Iterations *(User Stories)* or Releases *(Features)*

* Recommended if appropriate

Ceremonies

Participants / (*Owner*)

- ***Responsible for setting up the meeting, inviting the correct people, and ensuring the Entry Criteria is met***
- Involved Party

Entry Criteria:

- Specific conditions which must be satisfied before the Ceremony may begin

Key Activities:

- Specific activities that must be completed by participants

Exit Criteria / Output:

- Specific outcome or deliverable that is produced at the conclusion of the meeting

Metrics:

- Measurements to determine if the Ceremony was successful

* Recommended if appropriate

Sprint Planning

Participants: (*Owner*)

- Product Owner *
- ***Scrum Master***
- Software Engineer
- Embedded Tester

Entry Criteria:

- Prioritized User Story Backlog

Key Activities:

- Review previous Iterations to determine Velocity
- Review current and future Iterations to determine capacity
- Pull list of User Stories, based on priority, the team may be able to complete in next Iteration
- Review each User Story and get consensus from team on Story Points

Exit Criteria / Output:

- Based on capacity, team commits to delivering specific User Stories in the next Iteration
- User Stories are stack-ranked in the Iteration Backlog based on priority

Metrics:

- % of User Stories with an Iteration Assigned
- % of User Stories with Planned Estimate
- % of Scope Creep during an Iteration

* Recommended if appropriate

Sprint Planning

REAL WORLD

Prep Work

- Review previous Iterations to determine Velocity
    - Number of accepted User Stories and Story Points
- Review current and future circumstances to determine capacity
    - Holidays, vacations, etc...
- Refine Feature Backlog
    - Stack-rank based on value
- Pull list of User Stories the team may be able to complete in next Iteration

Activity

- Review the Definition of Done
- Review each User Story to ensure they are 'Dev-Ready'
- Estimate and get consensus on Story Points for each User Story
    - Planning Poker Video

CLASS ACTIVITY

1. Select Scrum Master
2. Review Features
3. Identify 3 - 5 User Stories for each Feature
    a. *HINT* - Keep User Stories the same color as Feature to help organize and start title with 'US'
4. Prioritize ***ALL*** User Stories
    a. *HINT* - You do not have to complete User Stories for a Feature before starting User Stories for another Feature
5. Estimate how many User Stories the team could complete in a 1 week Sprint
    a. Move these to the Sprint Backlog
6. Review and vote on Story Points for every User Story in the Sprint Backlog
    a. Pick a size 1 or 2 User Story and a size 5 or 8 User Story to establish a relative baseline
    b. Creator of User Story presents their User Story
    c. Each person must vote on the estimated Story Points required to deliver that User Story
    d. Discuss any Story Point estimates that are outliers
    e. Teams agrees on final Story Point estimate
7. Team sets Team Goal (deliverable) for the Sprint

Scrum / Daily Stand Up

Participants: (*Owner*)

Scrum of Scrums

- Business Partner / Product Manager *
- Business Relationship Manager
- ***Program Manager***
- Product Owner
- Quality Assurance
- Scrum Master

Daily Scrum

- ***Scrum Master***
- Software Engineer
- Embedded Tester

* Recommended if appropriate

Entry Criteria:

- Participants have access to current accurate data

Key Activities:

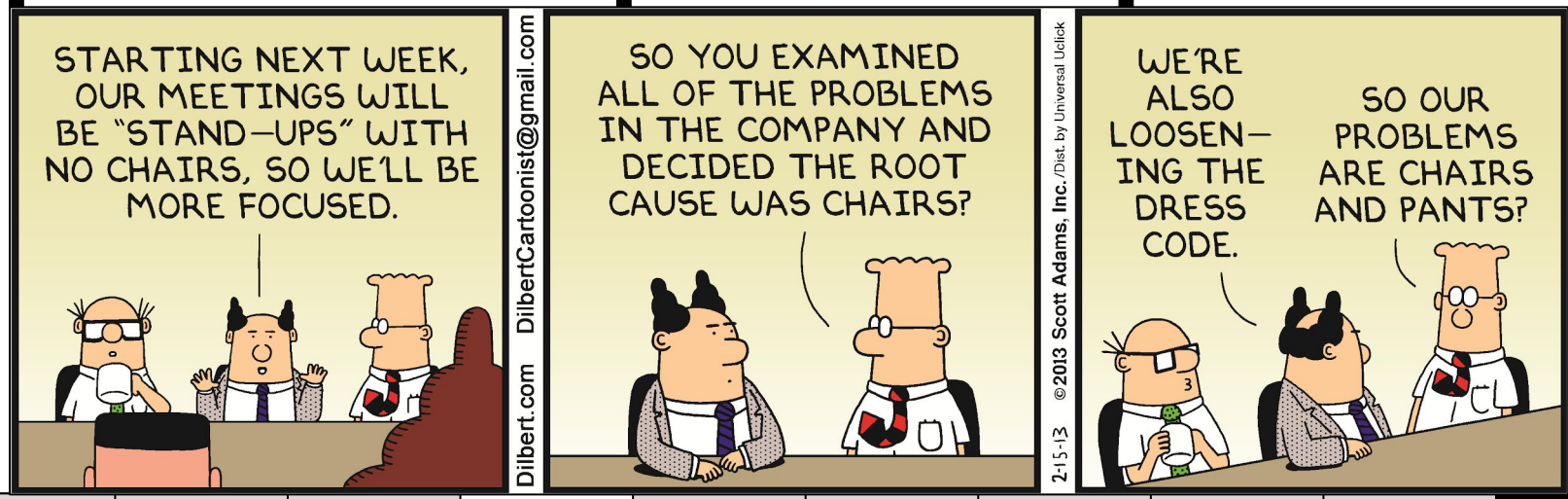
- Review what has been completed since previous Scrum
- Review what will be completed before next Scrum
- Review blocking issues and dependencies

Exit Criteria / Output:

- Program Manager (SoS) / Scrum Master / Product Owner (DSUP) has a list of blocking issues to resolve

Metrics:

- # of Blocked Artifacts



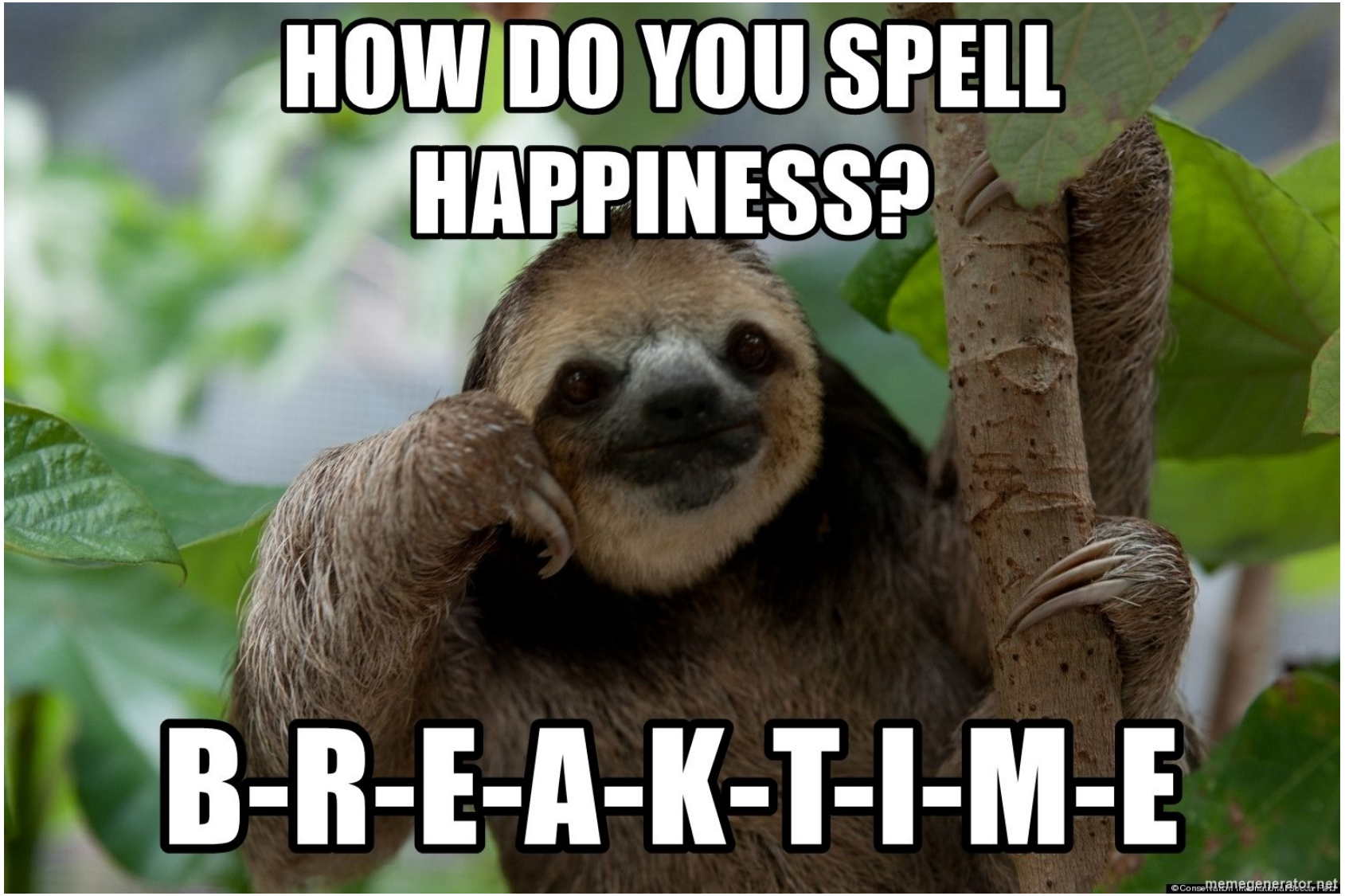
Scrum / Daily Stand Up

REAL WORLD - Daily Stand Up

- Each team member reviews
  - What they have completed since the previous Scrum
  - What they plan to complete before the next Scrum
  - Any blocking issues, dependencies, or impediments
- NOTES
  - It is best to work directly from the team's kanban board
  - The Scrum Master should facilitate and ensure the meeting takes place, but this should be self-lead by the team
  - This is not a time to solution
    - Any design or resolution sessions should be identified at the Scrum, but not take place during the Scrum
    - These will be action items to follow up on after

CLASS ACTIVITY

1. Each team member selects 2 - 3 User Stories that they did not write
2. Each team member selects one of their User Stories to
   a. Discuss what they have accomplished
      - Team member moves their sticky into next lane
   b. What they will complete today
   c. What blocking issues, dependencies, or impediments are slowing them down
   d. BE CREATIVE!!!
3. Scrum Master makes sure no one dominates the meeting and stays on topic
4. Scrum Master to document any impediments or action items


HOW DO YOU SPELL
HAPPINESS?
B-R-E-A-K-T-I-M-E
memegenerator.net

Demo

Participants: (*Owner*)

- Business Sponsor *
- Business Partner
- Business Relationship Manager *
- Program Manager
- ***Product Owner***
- Quality Assurance
- Scrum Master
- Software Engineer
- Embedded Tester

Entry Criteria:

- Feature in ‘Test’ state and all child User Stories in ‘Accepted’ state
- Feature and all child User Stories have successfully passed testing

Key Activities:

- Live, working demo of the Feature functionality is presented to the Business ‘Decision Maker’ and Impacted Stakeholders for approval
- Based on feedback, decision is made:
  - **Ready to Move to Production**, no changes required
  - **Ready to Move to Production**, changes to be deployed at a later time through a new Feature
  - **Not Ready to Move to Production**, additional Features and User Stores to be created

Exit Criteria / Output:

- Approved Features are staged for Production

Metrics:

- TBD

* Recommended if appropriate

Demo

REAL WORLD

- Teams provide a working demonstration to stakeholders of what has been completed
- Stakeholders make a decision to:
    - **Move to Production** - No Changes Required
    - **Move to Production** - Changes Deployed Later
    - **Do Not Move to Production** - Additional Work Required

CLASS ACTIVITY

1. Team members meet and sketch out what they think they would have accomplished during the Sprint
2. Spokesperson for each team is chosen to present
3. All teams return to main session and each team take a turn to present, including what was their original goal and what did they actually deliver
4. After each team presents, the groups decides:
    a. **Move to Production** - No Changes Required
    b. **Move to Production** - Changes Deployed Later
    c. **Do Not Move to Production** - Additional Work Required

Retro

Participants: (*Owner*)

- Business Partner / Product Manager *
- Business Relationship Manager *
- Program Manager *
- Product Owner
- Quality Assurance
- Scrum Master
- Software Engineer
- Embedded Tester

Participants and Owner determined based on scope of Retro

* Recommended if appropriate

Entry Criteria:

- Entry Criteria and level of detail based on scope of Retro
- Examples:
  - Team Retro = Post Iteration
  - Product Retro = Post Feature Deployment
  - Project Retro = Post Initiative Deployment
  - Deployment Retro = Post Milestone Deployment

Key Activities:

- Review open items from prior Retro
- Team discusses what went well, what was acceptable, and what could be improved
- Each team member is expected to contribute at least one item in each category
- Team members may use checks to indicate they agree with another's input
- Each item is presented and reviewed by the individual who submitted the item

Exit Criteria / Output:

- Action items are created for areas of improvement and assigned to team members to follow up
- Action Item list that is shared with all team members

Metrics

- % of Action Items completed from previous Retro

Retro

REAL WORLD

- Review open items from prior Retro
- Team discusses what went well, what was acceptable, and what could be improved
- Each team member is expected to contribute at least one item in each category
- Team members may use checks to indicate they agree with another's input
- Each item is presented and reviewed by the individual who submitted the item

CLASS ACTIVITY

1. Team discusses what went well, what was acceptable, and what could be improved
2. Each team member is expected to contribute at least one item in each category
3. Team members may use checks to indicate they agree with another's input
4. Each item is presented and reviewed by the individual who submitted the item
5. Pick two action items to create User Stories to assign to the next Sprint


L8R