Scrum

Scrum Artifacts

Definition of Done

- "when the code has been written" (programmer)
- "all of the tests have passed" (tester)
- "it's been loaded onto the production servers" (operations)
- "we can now sell it to customers" (business person)

- Each team creates its own "definition of done"

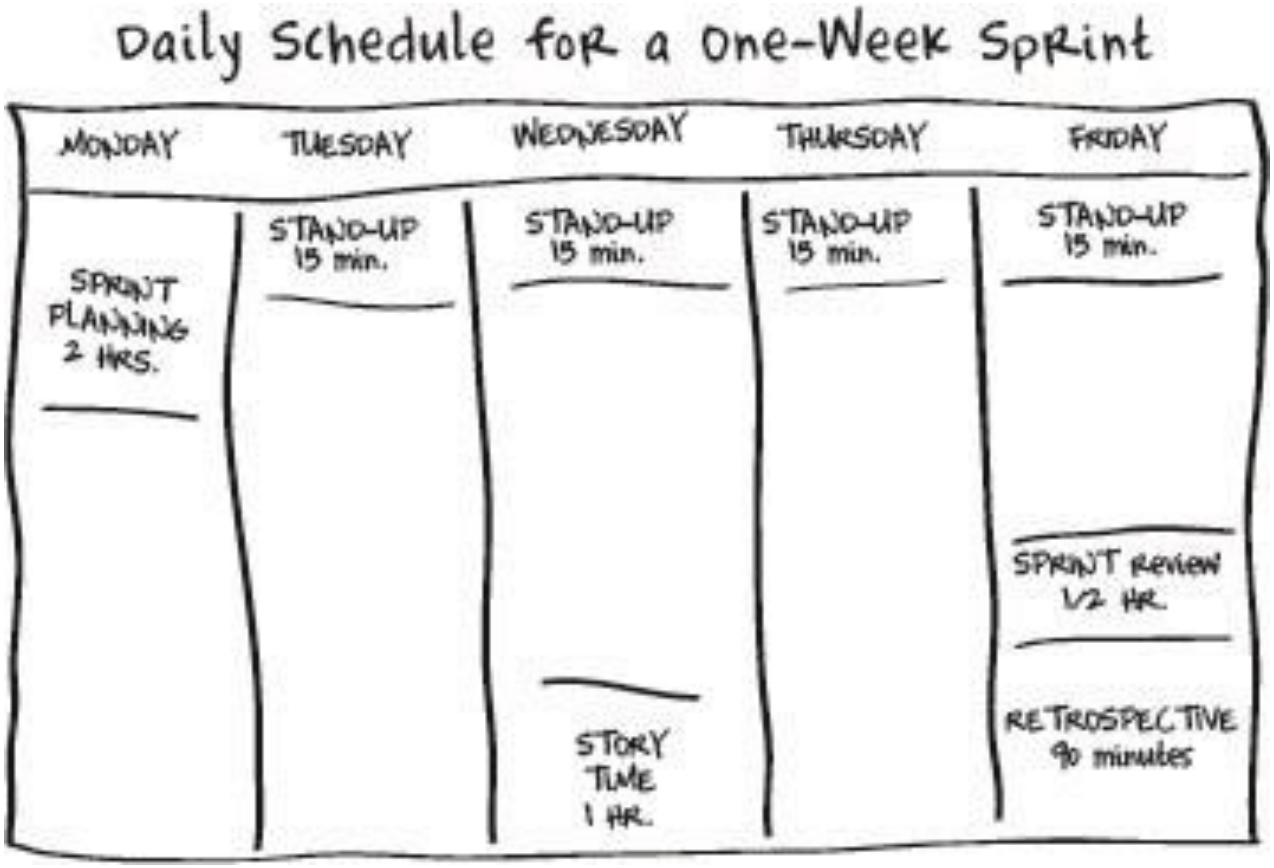
Definition of Done

- is a checklist of activities and artifacts that must be completed for every user story
- allows the team to focus on the truly important activities
- DoD is created prior to the first iteration
- DoD should be written down and kept visible throughout the project
- DoD ≠ acceptance criteria
- Nonfunctional requirements (performance, user experience) can be included in DoD

Definition of Done

- The Definition of Done checklist includes (at a minimum):
  - coding
  - testing
  - integration
  - addressing all issues/bugs
  - deploying to an appropriate environment
  - creating automated tests
  - conforming to design/coding standards
- Stories that do not meet the DoD by the end of the iteration are not considered complete, and do not count towards velocity

Scrum Ceremonies



Scrum Ceremonies

- Iteration Planning
- Daily Standup
- Iteration Review
- Retrospective

Iteration Planning

also called Iteration Kickoff

- Purpose:
  1. Discuss detailed requirements of the stories to be built in the iteration.
  2. Review and refine the acceptance criteria for each story
  3. Discuss the technical design and agree on the implementation approach
  4. Create a task-level plan for executing the work of the iteration.
  5. Make a commitment regarding what will be completed

Iteration Planning Agenda

1. Preview the iteration's stories
   - Jointly review the list of stories assigned to the team for the current iteration

2. Discuss and agree on acceptance criteria for each story
   - Discuss the detailed requirements for each story one by one
   - Take notes about the acceptance tests for each story
   - Agree on the basic acceptance criteria for each story

Iteration Planning Agenda

3. Discuss the design for the iteration's stories

- Determine how the design of the system must be altered or enhanced to support the functionality of the new stories
- Discuss infrastructure work, refactoring, research
- Agree on the basic design for each story

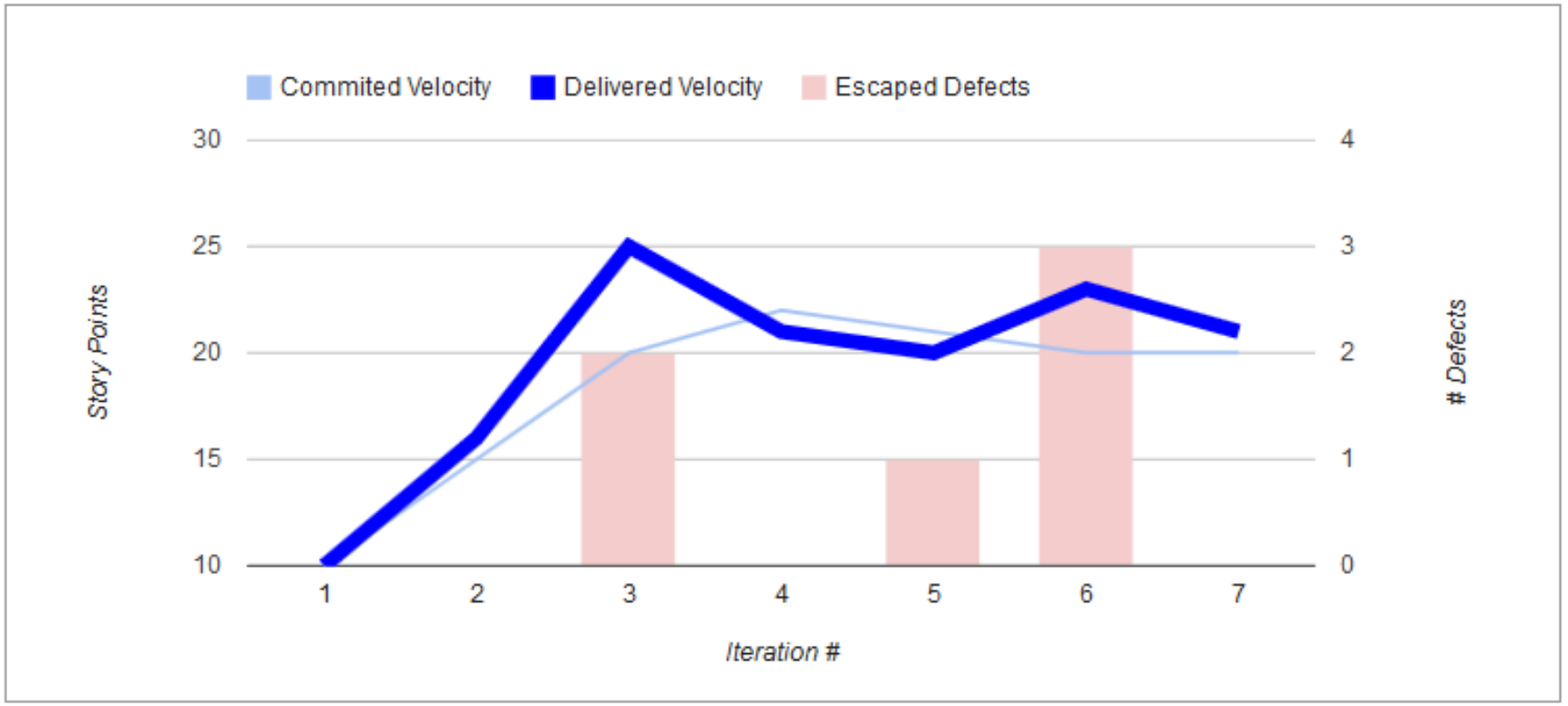
Iteration Planning Agenda

4. Create tasks for each story
   - Break each story into tasks, based on the design
   - Discuss any dependencies

5. Make a commitment
   - The team revisits and commits to the stories to be completed
   - The scope is fixed for the duration of the iteration

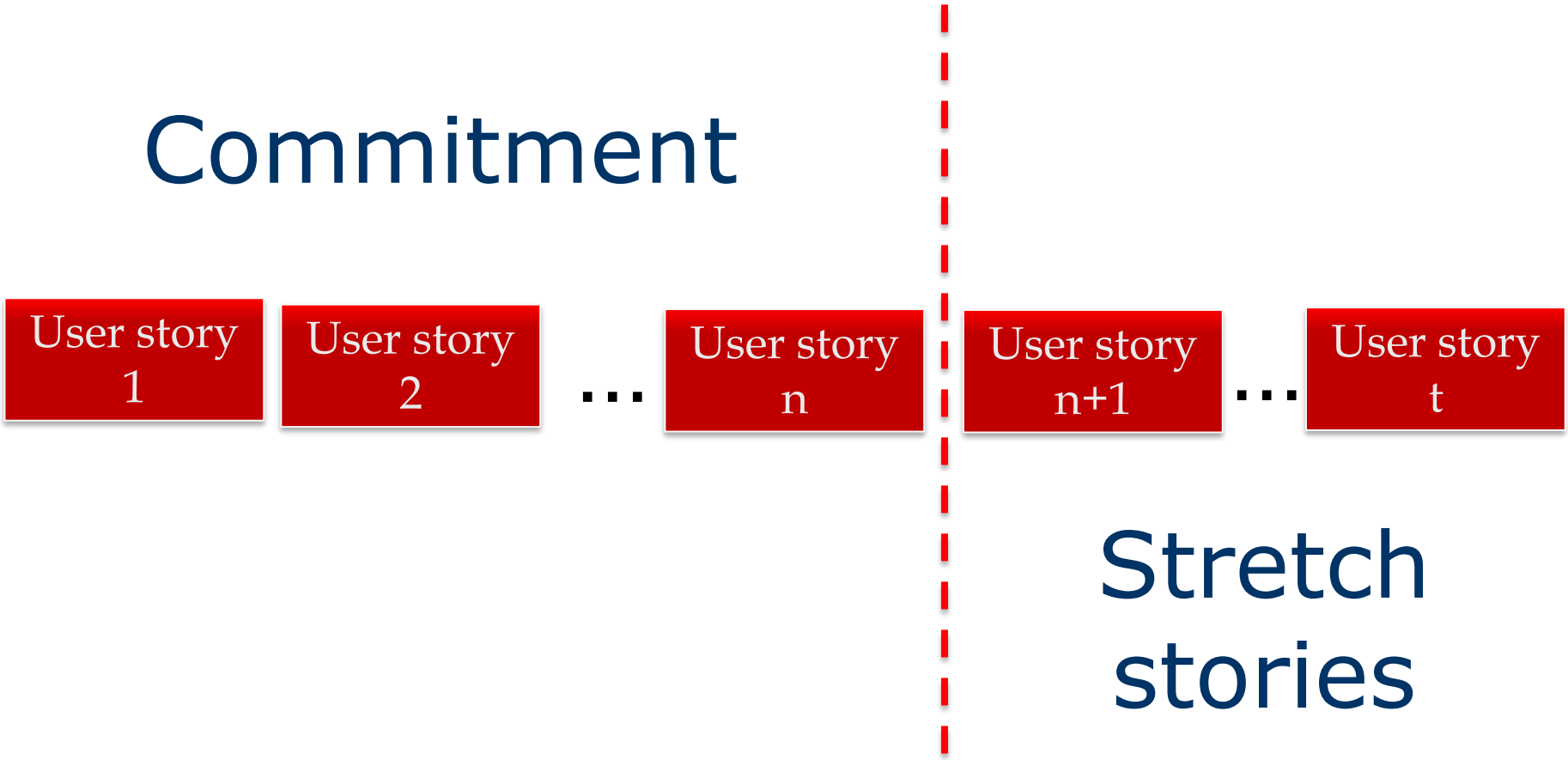
Awesome

Engineering Manager	Team Size	Release Planning	PSR Score	Iteration Duration	Last Update
Jeff Nielsen	8			2	5/12/2014



Iteration No.	Commited Velocity	Delivered Velocity	Commitment Deviation	Escaped Defects	Story Fitness	Sprint Review	Standups
7	20	21	5.00%	0			
6	20	23	15.00%	3			
5	21	20	-4.76%	1			
4	22	21	-4.55%	0			
3	20	25	25.00%	2			
2	15	16	6.67%	0			
1	10	10	0.00%	0			

Velocity prediction



Daily Standup

- is the team's opportunity to coordinate, synchronize, and plan its work for the next 24 hours
- the meeting is done in front of a task board
- each team member share what they have completed and any challenges they are facing
- timeboxed to 10-15 minutes maximum

Daily Standup

- Purpose:
  - To synchronize on how much work has been accomplished and how much remains
  - To inspect overall progress toward the Iteration goals.
  - To improve communication, identify impediments, and promote quick decision-making.
  - To improve knowledge sharing

Daily Standup

- Process:
  - The team meets in person
    - for distributed teams a videoconference can be used
  - A facilitator leads the discussion
  - Each team member answers 3 questions:
    - What did you do yesterday?
    - What will you do today?
    - Are there any impediments in your way?
  - The team sets a goal for what it wants to accomplish in the next 24 hours
  - The task board is updated in real time
  - Issues/topics requiring additional discussion are handled separately after the meeting.

Iteration Review

- is held on the last day of the iteration as a closure event
- the main purpose of IR is to obtain feedback from project stakeholders on the stories built during the iteration
- it gives everyone a shared understanding of the current state of the software
- provides a learning experience for both team members and their business counterparts
- timeboxed - 1 - 1.5 hours

Iteration Review

- Process:
  1. Invite a set of stakeholders from outside the team to participate (in person or via videoconference)
  2. Provide a simple test script which exercises the new stories
  3. Allocate time in the review for the stakeholders to use the software individually, with team members observing and helping
  4. Capture all feedback in real time
  5. Review and summarize feedback as a group

3. could be replaced with "*A team member runs through the test script, and asks stakeholders for additional testing alternatives*"

Retrospective

- team discussion about
  - what is working
  - what is not working
  - what can be improved
- there is always room for improvement :)
- retrospective is held at the end of an iteration
- timeboxed (usually 30 minutes per iteration week)
- good facilitation is needed

Retrospective

- Process:
  - Set the stage
  - Gather data
  - Generate insights
  - Decide what to do
  - Close retrospective

Retrospective

- Set the stage
  - comfortable and safe environment for the team
  - introduce the objective of the meeting
  - share initial context
  - ensure that basic ground rules are followed:
    - team is conversing (not arguing),
    - understanding (not attacking/defending) etc.

"Regardless of what we discover today, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand."

Norm Kerth
Retrospective Prime Directive

Retrospective

- Gather data
  - Create a shared picture of what happened during the iteration.
  - Use some brainstorming approach
  - Ideas can be written on a whiteboard, cards, or post-it notes for a co-located group. (Trello, Google docs, LeanKit for virtual teams)

Retrospective

Gather data approaches:

- What went well?
- What went wrong?
- What could be improved?

Retrospective

Gather data approaches:

- Enjoyable
- Frustrating
- Puzzling

Retrospective

Gather data approaches:

- Same
- More
- Less

Retrospective

Gather data approaches:

- Liked – What did we like?
- Learned – What did we learn?
- Lacked – What did we lack?
- Long For – What do we long for as an improvement?

Retrospective

- Generate insights
  - Reading cards/ideas anonymously (encourages people who are reluctant to say what they really think to participate)
  - Group similar items together
  - Discuss the items that seem the most important
  - Possibly do root-cause analysis on one or more ideas (using *5 whys / fishbone etc. techniques*)

Retrospective

- Decide what to do
  - Find 1-2 good enough ideas for improvement
  - Focus on the things which are in the power of the team to change!
  - Identify small experiments to be performed over subsequent iterations.
  - Capture summary and action items with target dates and responsible team members
  - Display the action items them in a place visible to the team

Retrospective

- Close retrospective
  - Summarize the collective view of the team
  - Reflect on what helped during the Retrospective process itself