Iteration Planning

5 Levels of Planning
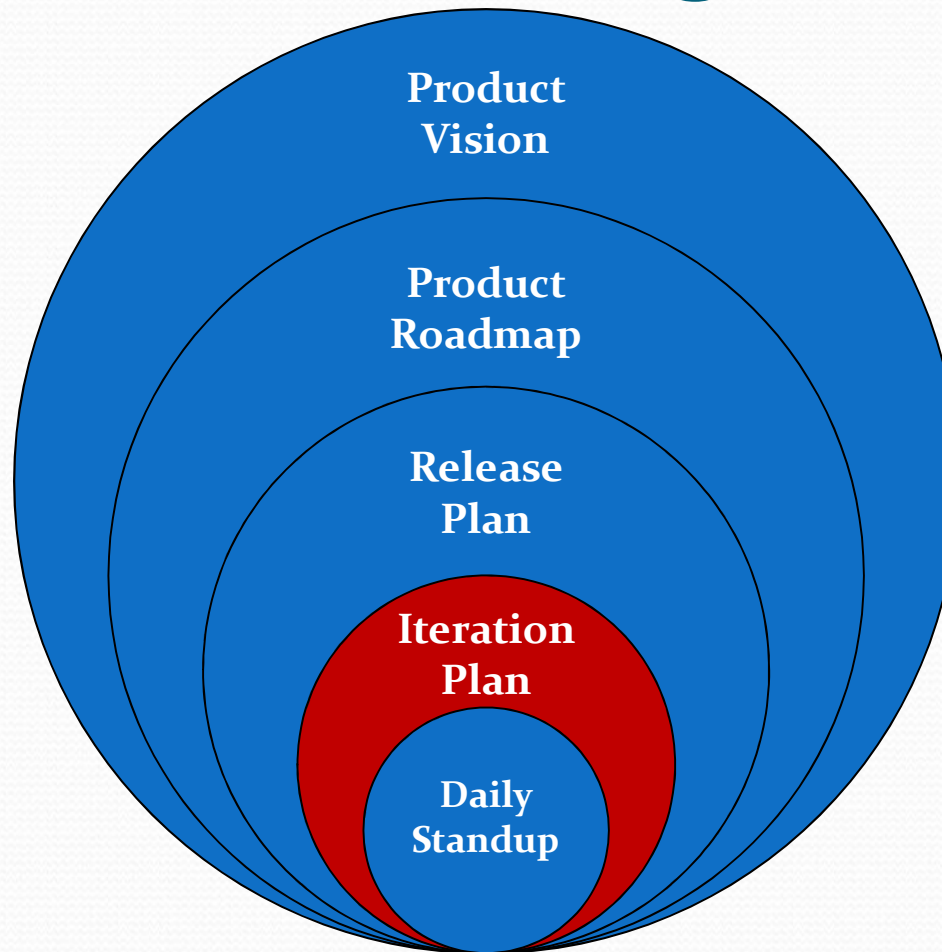
Iteration Planning

- Define scope as a team
- Define a clear understanding of "done"
- Plan just enough that you can commit

Roles

- Product Owner
- Scrum Master
- Team Member

Product Owner

- Prioritizes the backlog
- Communicates what is important … and what is not
- Is a proxy for the customer and other stakeholders

Scrum Master

- Responsible for the process
- Facilitates the meeting

Planigle

Team Member

- Asks questions
- Collaborates with others
- Signs up for work

The Backlog

- A ranked list of stories
- What is a story?
  - A scenario that we must do work to implement which results in business value
  - Typically in the form of: "As a <type of user>, I want <feature> so that <business value>"
  - Good stories meet the INVEST criteria

Before you Start

- Well Groomed Product Backlog
  - Prioritized
  - Estimated
- Iteration Theme/Goal

Stories	Schedule	People						Refresh	Log out

| 1.0 ▼ | Iteration 9 ▼ | All Owners ▼ | Not Done ▼ | Add Existing | Create | Update | Delete | View Surveys |

Velocity Used: ▶ 3.5 of 3.7 (95%) - Planigle

Name	Owner	Effort	Status	Public	Priority	User Priority				
User accepts license agreement	Walter Bodwell	1.5	In Progress ▼	true	1	4				
User maps iterations to release	Walter Bodwell	2	Created ▼	true	2	6.333				

Estimated   Prioritized

Planigle

Exercise 1

- Create a prioritized backlog
  - As a <user> I want <feature> so that <business value>
  - Estimate relative size
- At least enough for one iteration
- Choose any domain you like
- We'll use the results in a future exercise

What's your goal for the iteration?

Planigle

A Typical Iteration Planning Session

- Discuss Logistics
- Review Iteration Goals
- Understand the Stories
- Task Out the Stories
- Commit

Typical Duration: 3-4 hours

Attendees:
- Product owner
- Scrum master
- Delivery team

Materials:
- Stories (cards or online)
- Task planning material (cards, whiteboard, online)
- Planning/estimation materials (e.g. planning poker cards)

Planigle

Discuss Logistics

- Review Historical Velocity
- Review Team Availability
  - Holidays / Vacations
  - Meetings
  - L3 Support, outside commitment, etc
- Review the Definition of Done
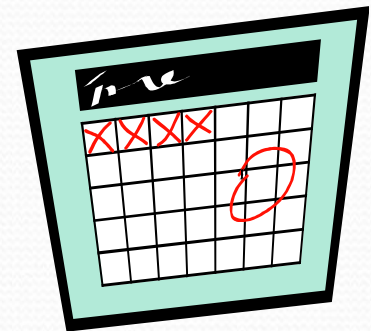
Definition of Done

- You need to define for your environment

- Definition will evolve over time

- Example:
  - Unit tests written and passed
  - Acceptance tests automated and passed
  - User facing documentation written
  - Checked in to the build
  - No defects introduced

DONE!

Review Iteration Goal(s)

- Product Owner
  - Explain the Goal (theme)
  - Make priority adjustments based on feedback from delivery team

- Team Members
  - ASK QUESTIONS
  - Understand the Goal, not just the desired features

Understand the Story

- Product Owner
  - Explain the Story
  - Explain the "Why" ("as a <role> I <what> so that **<WHY>**")
  - Break down as needed
  - Elaborate on acceptance criteria/tests
  - Make priority adjustments based on feedback from team

- Team Members
  - Understand the story
  - Understand and question the acceptance criteria (how will you build a test for each? What about...)
  - Validate the size/implementability

Planigle

Acceptance Criteria

- What is required for the success of this story?
- Typically determined at iteration planning jointly between product owner, dev, QA, writers, etc.

Task out the Story

- Define tasks

- Estimate the work involved

- Validate capacity again

The Product Owner can help
in avoiding less valuable work

Planigle

Hold Off On Names

- Keeps everyone focused on all the tasks, not just theirs
- Encourages team commitment
- Within the iteration, encourages focus on priorities
- And teamwork

Repeat

- Until the team cannot take on more
- Split stories as necessary
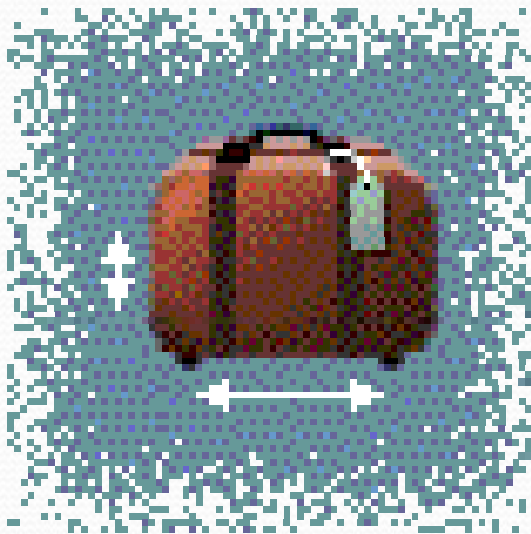
Splitting a Story

- The closer to the present a story is, the smaller it will become
- Those for this iteration need to fit within the iteration
- When splitting a story, each "slice" should add incremental user value

Planigle

Commit

- Everyone agrees the iteration is doable
- Use disagreement and uneasiness in team members to drive out hidden risks, tasks, and issues
- Drive agreement with a fist of five
    - Absolutely, no question
    - I think this is good and will make it happen
    - I can support this
    - I'm uneasy about this and think we need to talk about it more
    - Let's continue discussing this idea in the parking lot

Planigle

Effective Meetings

- Everyone should be focused on the task at hand
  - No working on laptops
- Every minute should be valuable
- If not, figure out how to make it so

Tools

Exercise 2

- Do iteration planning
- Go through stories in priority order
- Create acceptance criteria
- Task out
- Stop when you can't do more
- Commit

Do you believe in your result?

Estimating

- Identify a medium sized story that is well understood; call it a 5
- Now estimate other stories relative to that
- Is it about the same, ½ as difficult, twice as difficult?
- Use Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21
- If bigger than that or if too hard to estimate, split the story
- Tackle as a team; Planning poker can help (www.planningpoker.com)

Why Story Points?

- Time estimates
  - Vary by person
  - Encourage padding
  - Tend to grow stale
- Story points
  - More consistent from person to person
  - Not a commitment to time frame
  - Don't change as much
  - Easier to estimate relative size

Planigle

Velocity

- Now that stories have sizes, you can track how many points you typically get done in an iteration
- You can now use this to predict future completion rates
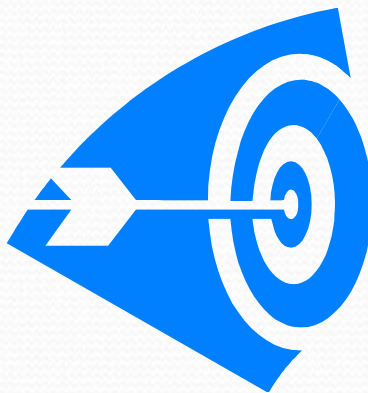
Release Planning Deliverables

- Plan for each Iteration
- Assumptions
- Dependencies
- Risks

- Are things synched up across teams?
- Are you attacking the most important stories?
- Does the team believe in the results?

Coordinating Teams

- Simplest if one team has the skills to take on an item by themselves

- If not, try to minimize the gap
  - Within the same iteration is ideal

- Touch base before and after iteration planning

- Daily scrum of scrum meetings can help

Kanban

- Instead of planning it all up front, you can pull things in as you go

- Keep iterations (Scrumban) or not (pure Kanban)

- Advantages
  - More flexibility (great for start ups and support)

- Disadvantages
  - Less predictability
  - Harder to coordinate

Planigle

Questions?

Walter Bodwell
Planigle
wbodwell@planigle.com
Twitter: @wbodwell
www.planigle.com
www.walterbodwell.com

Planigle