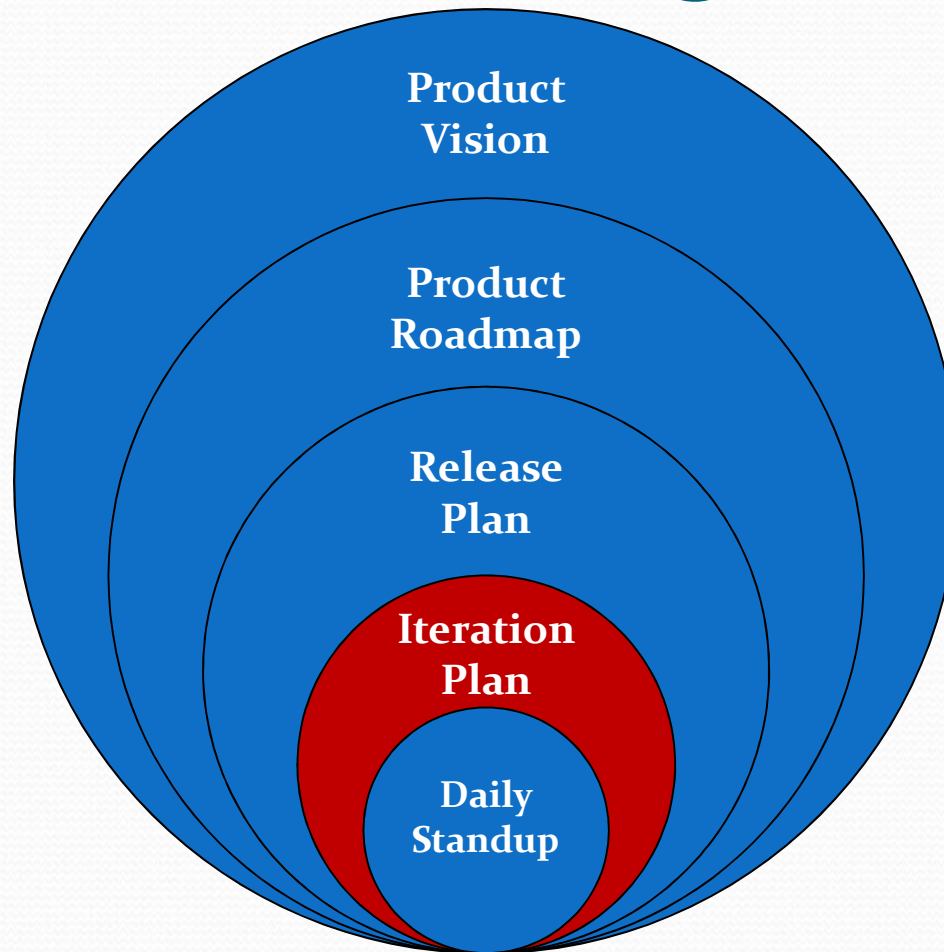


# Iteration Planning



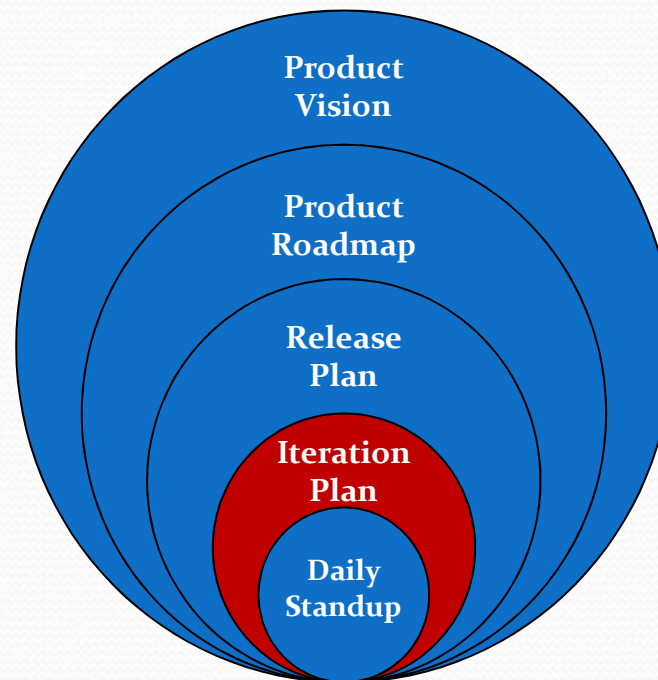
# 5 Levels of Planning

Adapted from "5 Levels of Agile Planning" by Hubert Smits



# Iteration Plan

- Define scope as a team
- Define a clear understanding of “done”
- Plan just enough to 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 agile meetings
- Helps to remove road blocks



# Team Member

- Signs up for work
- Asks questions
- Collaborates with others
- Communicates progress / blocking issues
- Makes it happen



# Before you Start



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

Name	Owner	Size	To Do	Status	Rank
User removes book from shopping cart	Ed Owner	1		Not Started	1
User chooses shipping option	Ed Owner	1		Not Started	2
User views order status	Ed Owner	5		Not Started	3

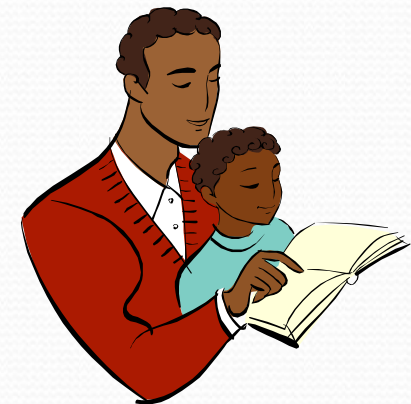
Estimated

Prioritized



# 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



# Exercise: Create a Backlog

- Goal: To create a backlog for a web site that sells books (competitor: Amazon)
- Roles: Product Owner, Stakeholders
- Assumptions:
  - Your team is focused on the web application
  - You have just enough in place to show a hello world screen
  - You have the ability to check in code, do a build and deploy
- Deliverables:
  - Prioritized list of things to do
  - Finer grained at the top (doable in a couple of weeks)
  - Larger grained at the bottom

# Sample Solution

- Find book by title / author
- Buy book via PayPal
- Show picture / details on book
- Show top selling books
- Show book reviews
- Remember user info
- Show user reviews
- Show other books by author
- Show related books

# Story Points

- Identify a medium sized story (that you would take on in an iteration) that is well understood; call it a 5
- Now estimate other stories relative to that
- Is it about the same,  $\frac{1}{2}$  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

# 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



# A Sizing Session

- Who
  - Product Owner
  - Scrum Master
  - Team implementing the story
- How
  - Take highest priority story
  - Product owner explains the story
  - Team asks questions
  - All team members vote on size at once
  - High and low explain why
  - Revote until consensus

# Epics

- Some stories are large
- They're too big for a team to take on in an iteration
- Stories far down the backlog can be left at this level



# Splitting a Story

- When splitting a story, each “slice” should add incremental user value
- Reprioritize and resize after splitting





# Splitting Example

- Buy a Book
  - As a book purchaser, I want to buy a book so that I can enjoy reading it

Might become

- View List of Books
  - As a book purchaser, I want to see a list of books that I can purchase so that I can make my purchasing decision
- Buy Book w/ Credit Card
  - As a book purchaser, I want to purchase a selected book via credit card so that I can enjoy reading it
- See Book Details
  - As a book purchaser, I want to see details about a book so that I can determine if I want to buy it
- See Other User Comments
  - As a book purchaser, I want to see comments from other users so that I can better determine if I want to buy it

# What About Risk?

- If multiple approaches and each has the same cost
  - No discussion necessary to size
- If multiple approaches and each has a different cost
  - Discuss enough to decide which is most likely
  - Use that for sizing
  - Resize if assumptions change
- Dependencies by themselves should not affect size



# Defects



- The most important thing in an iteration is anything that would prevent you from shipping
- Defects can be represented as stories or as tasks on the stories that they impact
- The goal is to keep up with defects as you go and to not allow them to build up
- Don't give points for defects; this keeps your velocity honest

# A Typical Iteration Planning Session

- Discuss logistics
- Review iteration goals
- For each story (in Priority Order):
  - Understand it
  - Task it out
- Stop when “full” and commit

Typical Duration: 1-4 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)

# Discuss Logistics

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



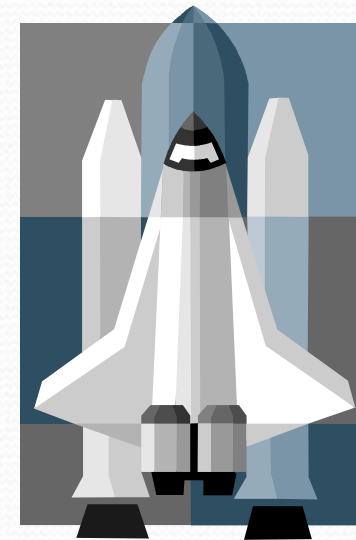
# Definition of Done

- 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



# Staying Releasable

- Goal: Could release after any iteration
- Reality: Ability to do this will evolve over time
- Staying releasable gives you the ability to more easily change direction / take on new things
- It also tends to improve quality
- And predictability



# Review Iteration Goal(s)

- At a high level, what are we trying to accomplish this iteration
- Examples:
  - Improve reporting
  - Improve performance
  - Get ready for beta





# Understand the Story

- Discuss the story
- Discuss why it is important
- Elaborate on acceptance criteria/tests
- Make priority adjustments
- Break down as needed



Do we need to answer this in order to commit?

# Acceptance Criteria

- What is required for the success of this story?
- Typically determined / refined at iteration planning jointly between product owner, dev, QA, writers, etc.
- Examples
  - Must be able to add a new user given a login, name and email address
  - Must generate random password
  - Must send password to email address
  - Must be able to log in with new login / password



# Task out the Story

- Define tasks
- Estimate the work involved
- Double check ability to commit






The Product Owner can help  
in avoiding less valuable work

# Tasks

- What do we need to do to accomplish this story?
  - Defined by the team in iteration planning
  - Refined throughout the iteration
  - Keep to a day or less
- 
- Examples
    - Implement add user screen
    - Send email with credentials
    - Test ability to add a user and log in



# Which Is It?

- Is it a goal (something worth achieving by itself)?  Story
- Is it a requirement that must be met?  Acceptance Criteria
- Is it something that you do in order to accomplish your goal?  Task

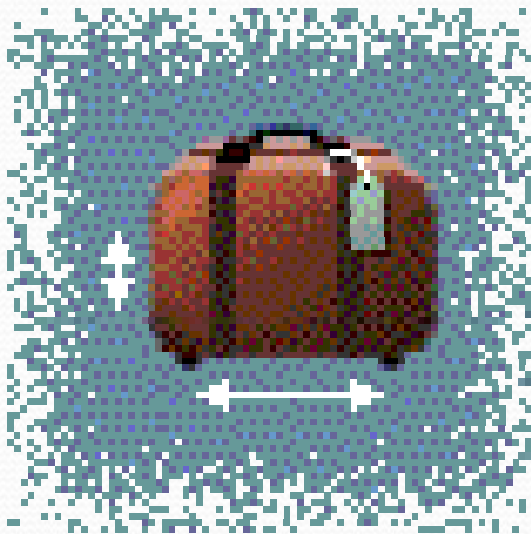
# Hold Off On Names

- 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



# Commit

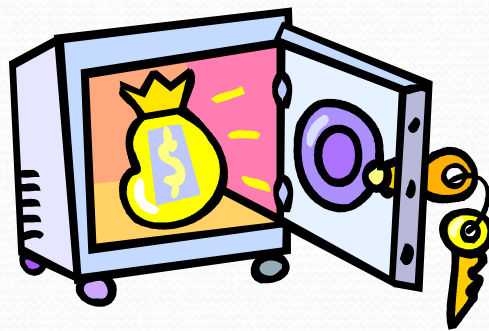


- Everyone agrees the iteration is doable
- Use disagreement and uneasiness in team members to drive out hidden risks, tasks, and issues
- Drive agreement with a fist of five
  -  Absolutely, no question
  -  I think this is good and will make it happen
  -  I can support this
  -  I'm uneasy about this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



# 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

Stories												
Stories	Reports	Schedule	People	Changes						Refresh	Log out	Planigle
1.1	Iteration 4	Team A	All Owners	All Statuses								
Number of Stories: 5		Velocity Allocation: 24 of 19.33 (124%) - Team A		Utilization: 23 of 19 (121%) - Team A								
Name	Owner	Size	Time	Status	Public	Rank	User Rank					
- User searches for books by author, title or ISBN number	Sue Tester	8	7	In Progress	true	1	1					
- Search by title showing just titles	Bob Developer	2		In Progress								
- Add more details to results	Bob Developer	2		Not Started								
- Add search by author or ISBN	Bob Developer	1		Not Started								
- Test search	Sue Tester	2		Not Started								
+ User views detailed information on a book	Sue Tester	5	5	In Progress	true	2	4					
+ Administrator adds new books to site	Sue Tester	5	5	Not Started	true	6	5					
+ Administrator deletes book	Sue Tester	3	3	Not Started	true	7	6					
+ Administrator edits existing book info	Sue Tester	3	3	Not Started	true	8	7					

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	



Sprint 3 for MultiVue Install		IT Team Tasks		days remaining				
Task description	Commit	Status	9/27	9/28	9/29	9/30	10/1	
<b>Requirements Component</b>			32	0	0	0	0	
Project Requirements Gathering	Campbell	Not started						
Formal Requirements Documentation	Campbell	Not started						
<b>MultiVue Configuration Component</b>			256	0	0	0	0	
Append Additional Demographics	Campbell	Not started						
<b>SAP database Component</b>								
Design SAP Database	Campbell	Not started						
Creation of the SAP Database	Campbell	Not started						
Create stored procedures on SAP database	Campbell	Not started						
<b>SAP Code Component</b>								
Creation of SAP .NET Component	Jan	Not started						
Creation of SAP Web Application	Jan	Not started						
<b>SAP Security</b>								
Creation of Security Administration Site	Campbell	Not started						
Secure Messaging	Campbell	Not started						
Security Integration	Jan	Not started						
SAP system testing	Campbell	Not started						
SAP System Verification	Campbell	Not started						
<b>SAP Hardening</b>								
Bug Fixing / Cosmetic Changes	Mark	Not started						
Install in Live Environment	Campbell	Not started						
<b>BizTalk 2004 Component</b>								
Extend ePEX-3 Adaptor	Campbell	Not started						
Extend Sw ift A daptor	Mark	Not started						
Extend Upstream Schemas	Mark	Not started						
Create AIC schema	Campbell	Not started						
Create Mappings	Campbell	Not started						
Create SAPAIC	Mark	Not started						

# Exercise: Iteration Planning

- Goal: Commit what the team can accomplish in the next iteration
- Roles: Product Owner, Scrum Master, Team Members
- Assumptions
  - Make assumptions about your team size and velocity
- Deliverables:
  - Stories
  - Acceptance Criteria
  - Tasks

Do you believe in your result?

# Sample Solution

- Find book by title / author – 5
  - Acceptance Criteria
    - Home page has fields to search by title and/or author
    - On submit, user shown books that match the criteria
    - If no matches, say “No matching books were found”
  - Tasks
    - Create template HTML for site pages - 2
    - Create search page - 4
    - Verify search works - 2

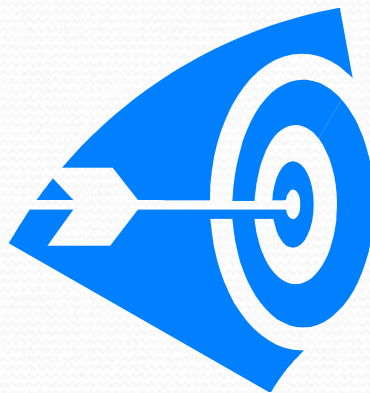
# Speeding It Up

- Before planning meeting:
  - Stories sized
  - First cut at acceptance criteria
  - First cut at tasks
  - Dependencies understood
- During the meeting:
  - Be wary of tools
  - Do we need to go into this now?



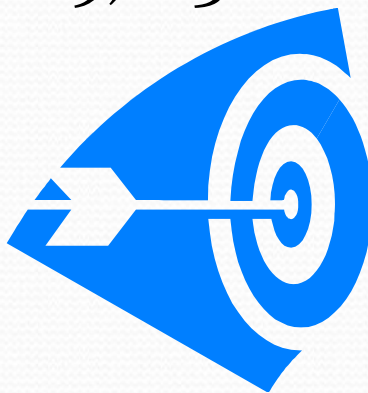
# Velocity

- Now that stories have sizes, you can track how many points you typically get done in an iteration
- Only count points for stories that get accepted in the iteration
- You can now use this to predict future completion rates



# Velocity Example

- Iteration 1: Took on 25 points, got 15 accepted
- Iteration 2: Took on 22 points, got 25 accepted
- Iteration 3: Took on 22 points, got 19 accepted
  
- Velocity =  $(15 + 25 + 19) / 3 =$  around 20 points



# Story Points Across Teams

- To get teams in the same ballpark, pick a baseline story
  - Each team should understand the complexity
  - Choose a medium size story
  - Call it a '5'
  - All other stories are relative to it
- Don't compare velocity
  - Used by a team to evaluate itself
  - If others use it for evaluation, it will be gamed and become useless



# 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 or 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

# Resources

Walter Bodwell  
Planigle

[wbodwell@planigle.com](mailto:wbodwell@planigle.com)

Twitter: @wbodwell

[www.planigle.com](http://www.planigle.com)

[www.walterbodwell.com](http://www.walterbodwell.com)

[www.agileaustin.org](http://www.agileaustin.org)

