This page intentionally left blank.

MoSCoW

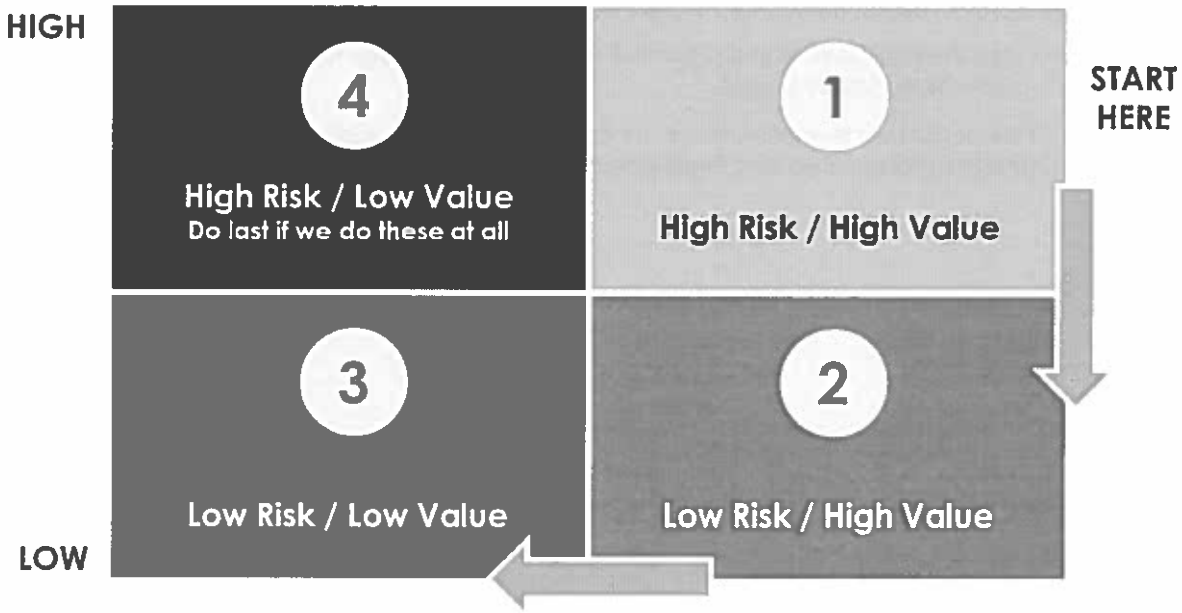
M	*MUST* have	The definition of MUST is that the feature needs to be included for the product to work.
S	*SHOULD* have	SHOULD have the feature if at all possible, or the more of these features the better but the product will still work without them.
C	*COULD* have	COULD have the feature if it does not affect anything else and there is time permitting.
W	*WOULD* have	Will not have the feature this time but WOULD like to include it in the future (these will go back to the Backlog or stay in the Backlog for another Release).

Source: Dynamic System Development Method

- The MoSCoW technique for ordering the Product Backlog comes from an early Agile approach, Dynamic System Development Method (DSDM). MoSCoW has been widely adopted in the Agile community.
- In order to plan a Release, the Product Owner must prioritize items on the Backlog. The goal is to ensure that the highest priority items, or the "MUST haves" and the "SHOULD haves" make it into the Release. Lower valued items, or "COULD haves" could make it in the Release if there is time. "WOULD haves" are items that do not make the cut, but they remain on the Backlog for a future Release.

Ordering Based on Risk & Value

Consider risk and value as you order.



Source: Cohn, Mike. *Agile Estimating & Planning*

- It's important to consider risk and value as you order the Backlog.
- Start by looking for Stories that are high risk and high value and assign them a 1. Addressing high risk/high value items early increases the chance of working through the complexities involved with high risk items and still meeting the Release date.
- After high risk/high value items we focus on those high value items that are low risk since these are the easy wins.
- Finally, if there is time, we address low risk/low value items.
- Items that are high risk and low value may not be addressed because they are not necessarily a good investment or good use of time.

Exercise: Order the Product Backlog

Instructions:

- In your Scrum Teams, order your Product Backlog:
  - Use the MoSCoW technique or the Risk-Value technique to rank the list from highest to lowest.
- If your Product Owner is in attendance, he or she should drive this since the Backlog is one of their primary responsibilities and feeds directly into Release Planning.

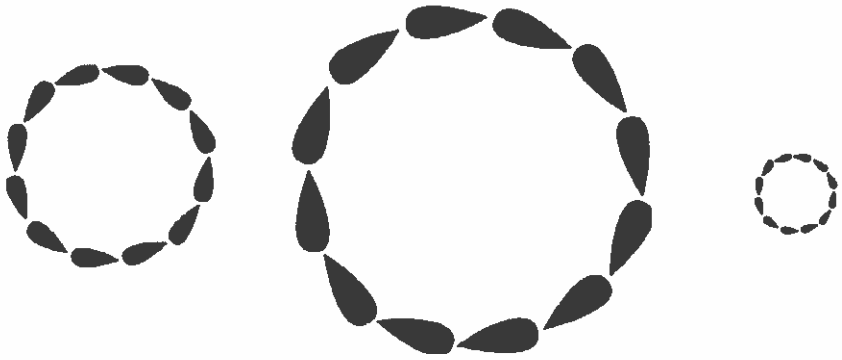
Assessment: Manage the Product Backlog

Questions:

1. True or False? The Product Owner orders the Product Backlog based on business value.
2. True of False? MoSCoW is one method for ordering the Product Backlog.

Estimating

Relative Estimation



Which circle is the biggest?

- Which circle on this slide is the biggest?
- How did you reach that conclusion? If you compared the three, that is exactly the premise behind relative estimating.
- An estimate is really a best guess. Traditional estimates have been in ideal hours or days and rarely tend to be accurate. Therefore, most Agile approaches use some form of relative estimation.
- Relative estimates are not time-based (days, hours, weeks); instead, they are based on the effort, risk, and complexity of each item as compared to others.
    - We compare the item to several others to determine its size in relation to the other items.
- Traditional estimates have been in ideal hours or ideal days and rarely tend to be accurate.
- The increased amount of time spent on estimating in hours or days does not statistically produce a more accurate estimate.

Accuracy or Consistency?

To establish consistency the Development Team must:

- Use a consistent approach to estimation.
- Establish a baseline.
- Remain stable.

- With relative estimation, consistency is more important than accuracy. Consistency can lead to predictability, which is what we encourage Teams to strive for with agility.
- If the Development Team can consistently complete a specific amount of work in a given Sprint, we can predict when we can deliver future Backlog Items or Releases.
- This requires the Development Team to:
  - Use a consistent approach to estimation.
  - Establish a baseline.
  - Remain stable. If an organization constantly swaps out Development Team members, the baseline changes and the opportunity for consistency is lost.
- On the next few slides, we will take a look at some common estimation techniques.
- As a self-organized, empowered Scrum Team, you should agree with each other on the best method that will work for the problem you are trying to solve, for your Team, and for your organization.

Discussion: Estimation

Questions:

- Why do we estimate?
- What has been your experience with estimating?
- Who does the estimation in your organization?
- And what can be done to improve estimating in your organization?

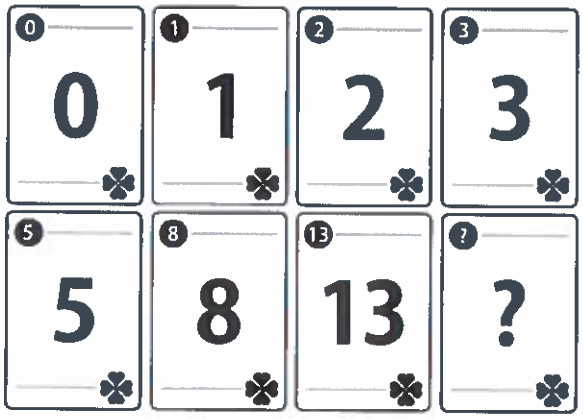
Story Points

Story Points are an arbitrary value assigned to User Stories by the Development Team to represent the effort required to complete the Story.

Source: *Agile Estimating & Planning* by Mike Cohn

- Story Points are an arbitrary value assigned to User Stories by the Development Team to represent the effort required to complete a User Story. It tells the Team how hard the Story is based on complexity, unknowns, or how much of it there is and relative to other Stories:
  - In most cases, the Story Point range is a specific series of numbers or words (2, 3, 5, 8, 13) or X-Small, Small, Medium, Large, X-Large.
  - The Team starts by identifying a baseline Story they all relate to.
  - Then, they size the rest of the Stories by comparing them to the baseline Story.
- If your Scrum Team is interested in adopting the use of Story Points, it is important to note that there is no conversion between Story Points and time.
- Story Points do not equal hours, days, or weeks. They are intentionally nebulous so the Development Teams can focus on the complexity of effort for a Story relative to other Stories.

Planning Poker®



- Planning Poker® is a consensus-based estimating technique. It is a fun and effective card-based approach.
- The Scrum Master facilitates.
- The Product Owner describes the Story or feature and answers questions the Dev Team may have in order to size the Backlog Item.
- Size is measured in Story Points. The Team uses the cards to vote on the size.
- The Development Team has the final say on all estimates.

Planning Poker is a registered trademark of Mountain Goat Software, LLC

- Planning Poker® is a registered trademark of Mountain Goat Software, LLC.
- This is a consensus-based estimating technique, similar to what is described in the Project Management Institute's Project Management Body of Knowledge when they refer to "Wideband Delphi."
- It can be useful, however, as it invites participation from the WHOLE Development Team.
- Those traditionally focused on development may not agree on an effort. Discussing viewpoints with each other can lead to consensus. When those who are thinking about Acceptance Criteria and testing weigh in, another level of complexity can be exposed if particular data is needed to test, or a particular environment, etc.

How to Play Planning Poker®

Steps:

Step 1	Each estimator has a deck of Planning Poker® cards.
Step 2	To establish a baseline, the Development Team identifies a relatively small Story and assigns it a value of 2.
Step 3	The Product Owner describes a Story and allows for brief discussion.
Step 4	Each estimator selects a numbered card from the deck for their estimate and places the card face (number) down.
Step 5	All cards are shown at the same time once the moderator (Scrum Master) gives the cue.
Step 6	The Development Team discusses differences between the estimates.
Step 7	Estimators re-estimate to reach convergence; the Development Team plays three rounds of Planning Poker® to reach consensus.

- These are the instructions for how to play Planning Poker®.
- Your instructor will provide you each with a hand of the Planning Poker® cards.
- For virtual or distributed Development Teams, you may also play "virtually" by using a session at planningpoker.com.
- Many smart phones also have free applications for download that are referred to as Planning Poker®, Agile Poker, or Agile Estimating cards.
- Most Development Teams decide on a best method for reaching convergence. One approach is to limit the number of rounds of voting.
- Some Development Teams take an average, or a mean number that they see in the group, or err on the side of taking the highest as the outlier in case risks are realized or until they mature in their process.

Class Exercise: Dog Points

Your instructor will provide direction for this exercise.

Exercise: Estimate the Product Backlog with Planning Poker

Instructions:

- In your Scrum Teams, use your Planning Poker® cards to estimate the Stories in your Product Backlog. Start at the top and record the size on each Story card or sticky note:
    - Remember the team roles and responsibilities!
- Estimate as much of the Backlog as you can, at least 3-4 items.
- In the last five minutes of the exercise:
    - Take any remaining unsized items and assign random Story Points to them. It's OK, this is just a simulation!
    - Once all items have a size, add up all the Story Points in the whole Backlog. Write this number on a sticky or card note by itself.

This page intentionally left blank.

Velocity

- Velocity is the number of Story Points a Development Team completes in a Sprint.
- At the end of each Sprint, the Story Points of all completed and accepted Stories are added up.
- A Development Team's average velocity is a running number that stabilizes over time to become fairly consistent, and becomes a number a Team can use to help with Sprint Planning.

Sprint 1 – 5+3+5+8+8+1+8=38
Sprint 2 – 5+8+3+5+8+2+3=34
Sprint 3 – 5+8+8+3+5+8+8=45
Average running velocity – (38+34+45)/3=39
This Development Team should forecast Sprints and Releases using approximately 39 Points per Sprint, plus or minus some uncertainty buffer.

- Velocity is the number of Story Points a Development Team completes in a Sprint. It's main purpose is to predict Release dates, but it can also help to create Sprint forecasts.
- You need a minimum of three to five Sprints to determine velocity you can use for planning:
  - At the end of each Sprint, add up the Story Points of all completed and accepted Stories.
  - Add the Story Point total for all the Sprints together.
  - Divide the total by the number of Sprints. This gives you the average running velocity.
  - Use the average running velocity plus or minus some uncertainty buffer to forecast future Sprints and Releases.
- In the example on the slide, the average running velocity is based on three Sprints. The Development Team should forecast Sprints and Releases using approximately 39 points per Sprint, plus or minus their uncertainty buffer.
- Each Team's velocity will be entirely different because each Team will have a different baseline of Story Pointing:
  - One Team might have a velocity of 20 while another Team's velocity is 100.
  - Both Teams are delivering high-quality Stories for their respective Sprints and working the same hours.

Exercise: Forecast a "Scope Fixed Release" Using Velocity

Scenario: Your boss comes to you and says, "How long will it take for you to deliver all of the items in the current Product Backlog?" She says she wants that number in weeks.

Instructions:

1. Your instructor will give you an estimated velocity to forecast with. Assume 2-week Sprints.
2. Number of Sprints = total Story Points in Backlog/forecasted velocity. Then convert this into weeks.
3. Now add and subtract 20% to give a range to buffer for some uncertainty.
4. Be prepared to tell your boss the estimate of how many weeks it will take to deliver this scope.

"T-Shirt Sizing" Estimation

- This technique helps Development Teams group Stories together for comparison and then group similarly sized Stories.
- To avoid confusion between the numbers in Points and the temptation to correlate to hours, days, or weeks, this technique uses "t-shirt sizes": small, medium, large, and so on.

- Another technique that has been gaining popularity in the Agile community is an Affinity Based Estimation technique using "t-shirt" sizes.
- This helps Development Teams break away from ideal hours or days and truly focus on how complex something is compared to something else.
- This technique involves comparing Stories to each other but then grouping "like" Stories together under the buckets of extra small, small, medium, large, and extra large.
- This technique can be relatively silent if you have the Story cards written or printed out.
- Development Teams can discuss after the "shuffling" has stopped and everyone has made their way through the pile of Story cards.

Assessment: Estimating

Questions:

1. True or False? Relative estimates are time-based.
2. True or False? When you use Story Pointing to estimate, you assign points to a baseline Story and then size the rest of the Stories by comparing them to the baseline.
3. True or False? A Development Team's average velocity is a running number that stabilizes over time and can be used to help with Sprint planning.

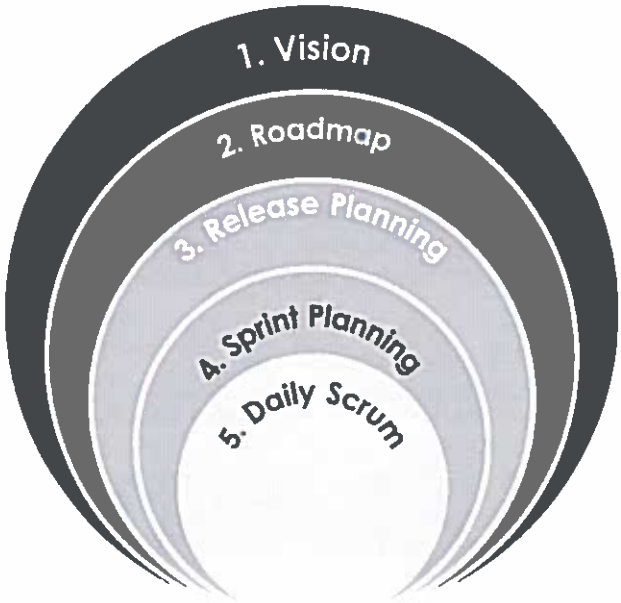
TEKsystems
Our people make IT possible.

Agile Planning: Release Planning

Agile Planning – Release Planning

Release Planning:

- Is deciding what to build and in what order.
- Involves planning multiple Sprints or Iterations to predict when a Release (or Releases) might be delivered.
- Occurs as often as needed to build the foundation on which to deliver the Vision.



- The Product Backlog with our requirements and Stories is an input to holding Release Planning.
- Release Planning gives direction to the project. It helps us decide what to build and in what order and answers the questions:
  - How many Sprints will it take us to deliver this set of features.
  - How much of this scope can we get done by a specific date?
  - How many people or Teams do we need for the project?
- The Release Planning session gives us the opportunity to see how many features or which of the features on our customer's list can be delivered by a particular date given a particular budget and/or Development Team. If all of the features are required, the Release Planning session can be used to determine the date when all of those features can be delivered with the given Scrum Team.
- Release Planning can occur at any time that a new project is kicking off.
- Some organizations standardize on their Releases to production (e.g., once per quarter) in which case Release Planning can fall in sync with that schedule.
- Release Planning is a significant session. Many who do quarterly Releases invest a full 6-7 hour Release Planning session to frame up the Release. It can take less time for more frequent Releases with a smaller feature list or Product Backlog.

This page intentionally left blank.

Release Planning Readiness

Key inputs to Release Planning are:

- Vision and Roadmap.
- Product Backlog ordered by business value.
- Scrum Team working agreements on Estimation approach.
- Velocity, if the Scrum Team has worked together on a previous Agile project.
- If the above items are not available, the Scrum Team is not ready for Release Planning.
- Scrum Masters should coach the Product Owner and Scrum Team to prepare for this meeting.

- The Product Owner facilitates the Release Planning session. Release Planning cannot take place if the Product Owner is not ready and has no ordered Product Backlog.
- The Scrum Master may coach the Product Owner on getting ready for this session or co-facilitate to keep it moving.
- Key Inputs to Release Planning are:
  - Vision and Roadmap.
  - Product Backlog ordered by business value.
  - Scrum Team working agreements on estimation approach.
  - Velocity, if the Team has worked together on a previous Agile project.
- If these items are not available, the Scrum Team is not ready for Release Planning.

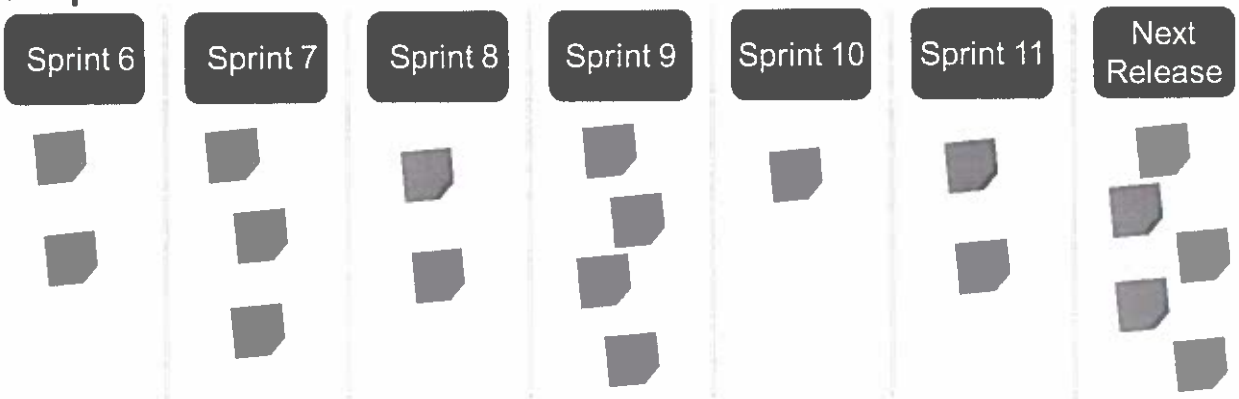
Release Planning Meeting

To hold Release Planning:

- The Product Owner describes and facilitates discussion on each Story or Product Backlog Item.
- The Development Team has a timebox for discussion – typically five minutes per Story.
- At the end of the Discussion Timebox, the Development Team estimates.
- The Scrum Master Timeboxes everything to keep the meeting on track.

- If all the inputs are available, we are ready to hold Release Planning.
- This is not a tasking session, nor is it a detailed design session.
- It is a high-level planning session in which we are discussing just enough detail to provide rough order-of-magnitude sizing.
- The Product Owner typically reads or explains a Product Backlog Item or User Story.
- The Development Team is given a timebox for discussion—typically about 5 minutes per Story.
- When the time is up, the Development Team estimates the work based on complexity and effort relative to similar work:
    - They should have agreement on what their "baseline" is.
    - What's the definition of "Small," for example, or "5 points"?
    - If they have Stories identified to continue to compare back to, this process will be much more efficient.
- Scrum Masters help keep this whole thing moving and ensure that all of the players are prepared.
- There is no prescription for who captures information in this session but it does need to be captured. Acceptance Criteria that come out of discussion, the resulting estimate in Points or t-shirt size, etc.

Sample Release Plan



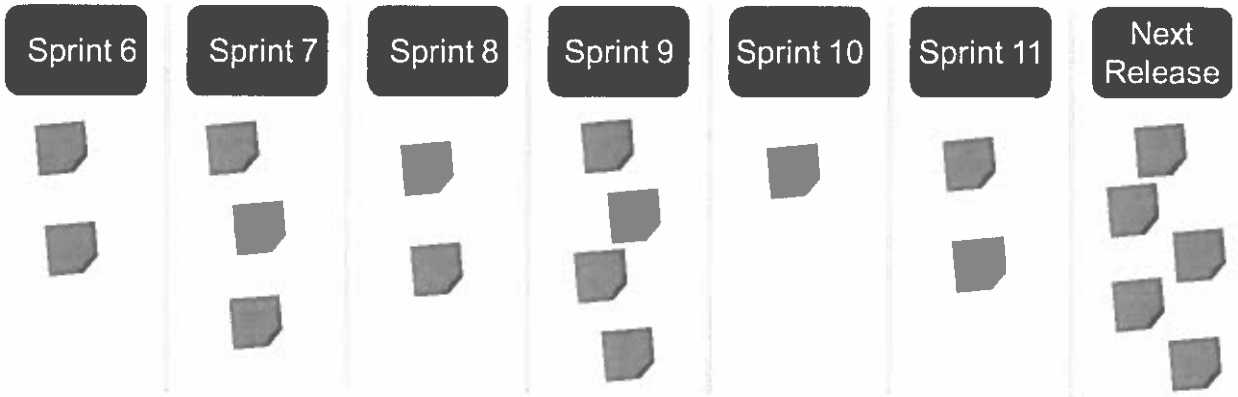
- A Release Planning session is not a large effort and does not need to be a large effort.
- Shows order of Stories, approximately where they will fall in Sprints.
- Allows Team to focus on highest ordered items, try to get them completed by end of Sprint, then shift to the next in order of the Product Backlog.
- Always subject to change based on reordering, results/actual progress, changing biz conditions.
- Updated after each Sprint to reflect reality as it emerges, as described above.

- This is an example of a Release Plan.
- It is not a large plan such as a Microsoft Project plan.
- It's enough to give visibility to the order of the Stories and approximately where they fall in the Releases' Sprints.
- We don't want to waste time on items in the Sprint that begins on July 5 now in May because something in our business may change and the Product Owner may take that item off the plate and replace it with something else.
- This type of planning allows the Development Team to focus on what is highest priority, and to get it done by the end of the Sprint, and then shift focus to what is next in priority.
- The Product Owner always needs to be refining the Backlog and staying ahead of the Development Team, so that when it is the next Sprint Planning time, he or she can ask the Development Team to work on only the next highest priority items and not waste time/cycles.

Exercise: Release Planning Meeting ("Date Fixed Release")

Instructions:

- In your Scrum Teams, hold a Release Planning session. Assume that the Release period will be six 2-week Sprints (3 months), and culminate in a single Release at the end of the 3 months.
- On the wall or a table, lay out seven sticky notes (or cards) horizontally, one for each Sprint ("Sprint 1," "Sprint 2," etc.), and then the seventh one will say "Next Release" (meaning that Story will not likely make the Release).
- Using your forecasted velocity per Sprint, place Stories into Sprints (columns) by placing the Story under the Sprint it will go in based on your forecast.
- Did you have enough Stories to fill the Release period? Did you have too few?



This page intentionally left blank.

Assessment: Agile Planning: Release Planning

Questions:

Product Owner

1. True or False? The Scrum Master describes and facilitates discussion on each Story or Product Backlog Item during Release Planning.
2. Which of these are key inputs to Release Planning? Select all that apply.
   A. Vision and Roadmap
   B. Product Backlog
   C. Scrum Team working agreement on estimation approach
   D. Velocity if the Team has worked together

The Sprint

Daniel Pink
Drive book

Practology - Why people do what they do.

Sprint Planning Readiness

Key inputs to Sprint Planning are:

- The Release Plan.
- The ordered, estimated Product Backlog.
- The Scrum Team's upcoming capacity for the current Sprint, including any known time off, appointments, etc.
- The Scrum Team's forecasted velocity, if they have been together for a previous Sprint or Sprints.
- The company calendar, including any holidays, town hall meetings, etc.

- In order to prepare for Sprint Planning, we need the Release Plan and to ensure that the ordered Product Backlog has the items at the top in a ready state to work on.
- The Product Owner should have included the necessary detail and be prepared to discuss with the Development Team at Sprint Planning.
- Any known company holidays or meetings should be brought to planning, as well as any Development Team member vacations, appointments, etc.
- If a Development Team has been together for several Sprints and knows their velocity, this is a key planning metric that we will use as input into planning.

Capacity vs. Velocity

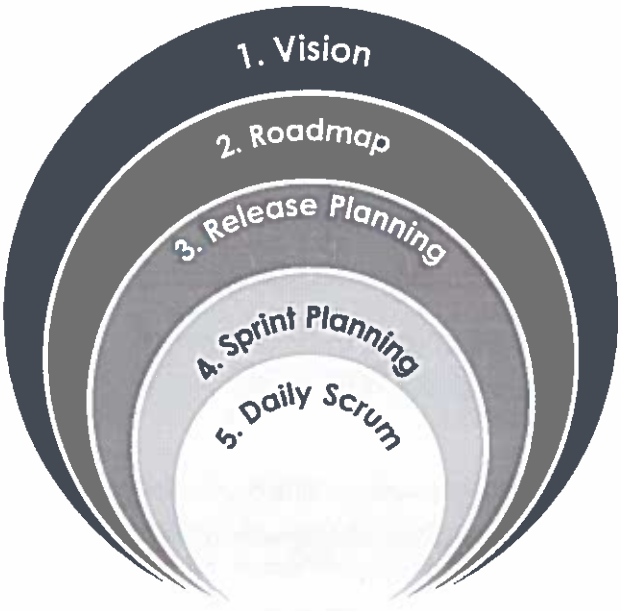
- Capacity is *how much time the Development Team has available* to work during the Sprint, including the known holidays, time off, etc.
- Velocity is the *amount of Product Backlog that the Development Team is historically capable of completing in a Sprint,* based on empirical data—it is a planning metric.
- It is important for the Sprint length to remain consistent for the Development Team to establish a cadence and forecast velocity.
- When Sprint lengths or Development Team staff change throughout the Release, this heavily disrupts the usefulness of velocity to use as input for Sprint forecasting/planning—remember that and adjust velocity forecasts as needed.

- Capacity is how many hours during the Sprint the Development Team is available to work.
- This involves looking ahead at the next 2 weeks, for example, if we are doing 2-week Sprints, and taking into account those company holidays, vacations, appointments, etc.
- Some Development Teams assume that there are 6 productive hours in a day as a guideline, not 8, accounting for any meetings, answering emails, etc.
- Others assume 7.
- It will vary from company to company.
- Capacity is NOT the same as velocity.
- Velocity is the amount of Story Points or work that the Development Team is historically capable of producing in a Sprint.
- This number is a "rolling average."
- For example, if our velocity in Story Points for Sprint 1 was 60, Sprint 2 was 73, and Sprint 3 was 68, our average velocity going into Sprint 4 will be 67.
- In other words, the Development Team should take on no more than 67 Story Points worth of work for Sprint 4.

Agile Planning – Sprint Planning

Sprint Planning

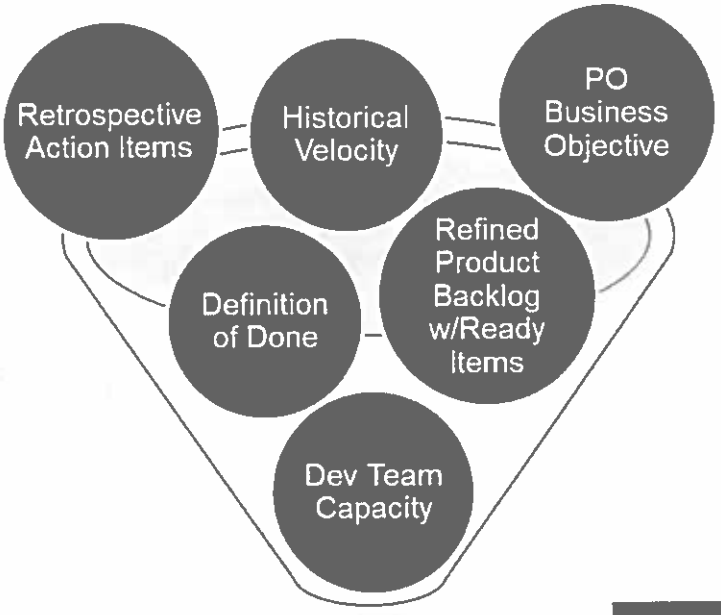
- Also known as an Iteration in some approaches, it occurs every 1-4 weeks depending on the Team or organization.
- A Sprint is the Timebox in which the Team completes a working product increment.
- Scrum Teams maintain a "cadence" to determine a consistent pace and baseline so they can estimate how much work can be completed per Sprint.
- This allows the Scrum Team to consistently estimate how much work can be completed.



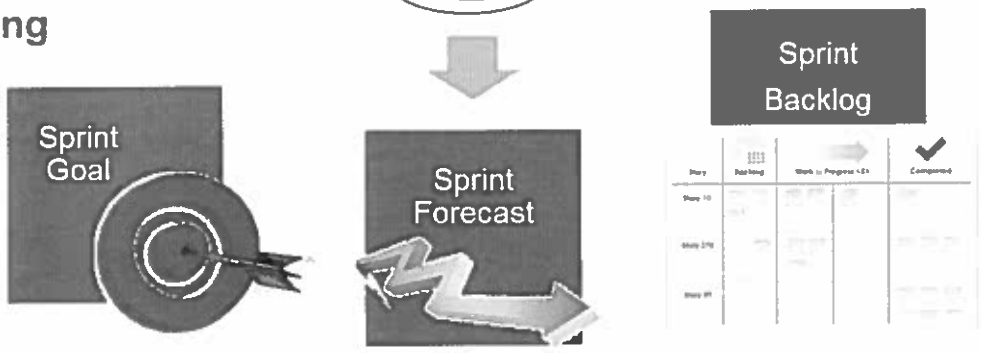
- Let's level-set where we are at with Agile Planning. We've got a product or project Vision and a high-level Roadmap.
- We've simulated a small slice of Release Planning so that we understand what items will be at the top of the Backlog.
- What's highest on the list will get addressed in the first Sprint Planning session.
- We're now ready to take a look at Sprint Planning.
- If your Team has been together working with Agile for awhile, you may have already adopted a rhythm or cadence.
- It is important to keep this Sprint timebox (1 week, 2 weeks, etc.) consistent.
- We would not do a 1-week Sprint and then say "extend" the Sprint to 8 days next time or change that Timebox.
- It takes away any baseline and any opportunity for the Team to be consistent and become predictable in what can be achieved in a Sprint.

The Sprint Planning Event

Inputs to Sprint Planning



Outputs from Sprint Planning

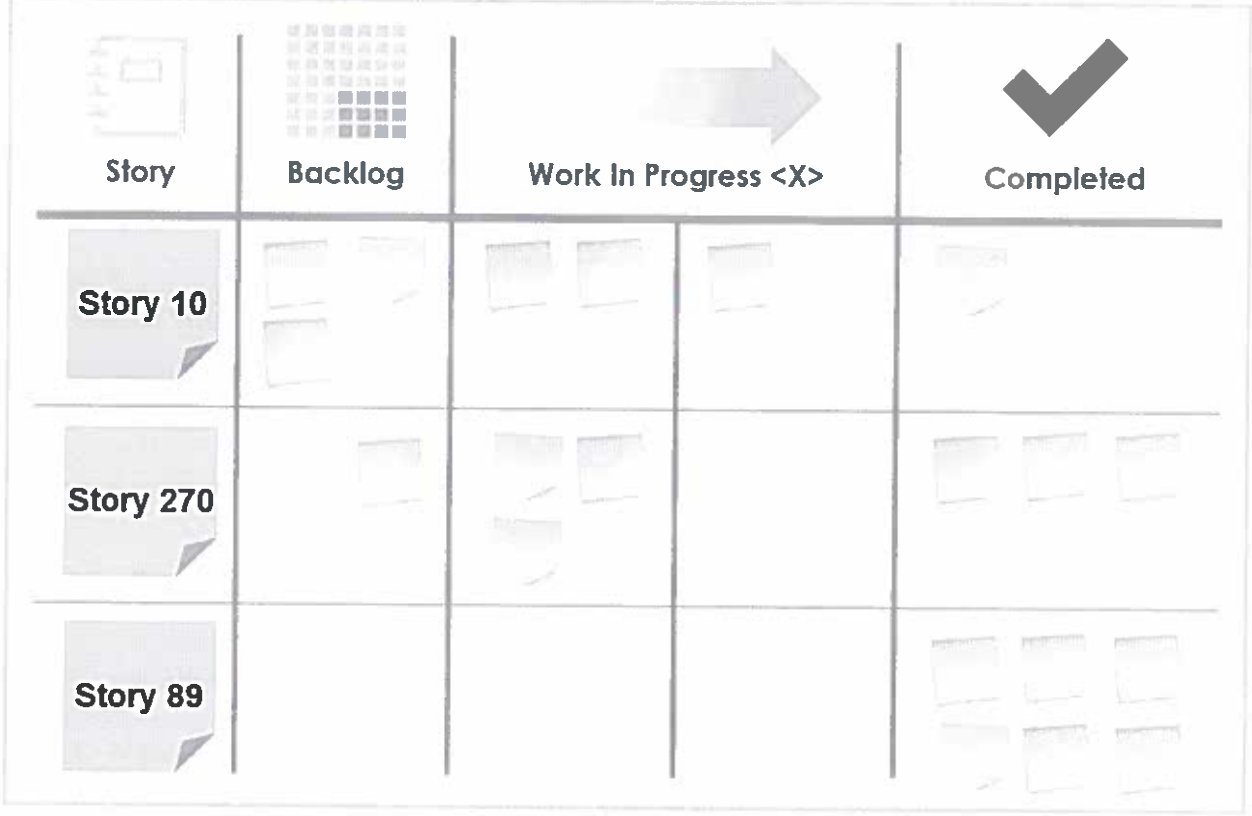


- The objective is for the Dev Team to realistically forecast the Stories and tasks that will get them to completion within the Timebox (Sprint).
- The Product Owner is present to answer questions about User Stories or Product Backlog Items.
- The Scrum Master facilitates as wanted or as needed, to keep the event moving.
- Often, the Dev Team decomposes (breaks down) the Product Backlog Items into tasks that they will complete during the Sprint, and estimates those tasks. The tasks are often broken down to units of one day or less.
- Sprint Planning is a working session for the Development Team, the Scrum Master, and the Product Owner.
- It is NOT a management or an executive forum.
- The Product Owner is there to clarify Backlog order and User Story requirement details.
- The Stories should have been previously sized in Release Planning or Product Backlog Refinement.

Sprint Planning Output: The Sprint Backlog

The Sprint Backlog:

- Is a set of Product Backlog Items selected for the Sprint, plus a plan for delivering the product increment, and realizing the Sprint Goal.
- Often expressed as a set of Stories that are broken down into tasks, as shown below.
- Is a forecast of what the Dev Team thinks they can complete.



- The Sprint Backlog is a set of Product Backlog Items selected for the Sprint, plus a plan for delivering the product increment and realizing the Sprint Goal.
- The Sprint Backlog is a forecast by the Dev Team about what functionality will be in the next increment, and the work needed to deliver that functionality.
- Defines the work the Dev Team will perform (in tasks) to turn the Product Backlog Item into a "Done" increment.
- Should be visible and include all work for the Sprint, with enough detail for work in progress to be understood at the Daily Scrum.
- Some Teams break Stories into tasks one at a time, while other Teams will work in small groups to break Stories down in parallel—to speed up Sprint Planning.

- Some Development Teams will be tempted to go over their velocity if they see that hours are available in their capacity—this should be avoided, and room for uncertainty should be built in.
- If they overload their plate, they take away the flexibility that the framework provides for them to inspect and adapt.

Exercise: Sprint Planning

Instructions:

- In your Scrum Teams, simulate Sprint Planning. Note that your Sprints will be made up of three simulated days. Each day is 10 minutes long.
- Discuss velocity and capacity, and review the Product Backlog.
- Break the first Item on the Backlog down into tasks, adding it to the Sprint Backlog. Repeat as you have time.
- Product Owners answer questions, Scrum Masters keep the process moving, Dev Team members break Stories down into tasks and estimate tasks.

During the Sprint

Product Owner

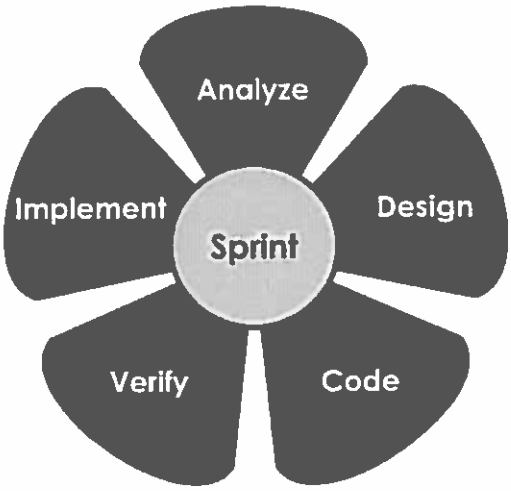
- Clarifies requirements.
- Gives feedback (over the shoulder check):
  - Accepts/rejects.
  - Recommends tweaks.
  - Product Backlog Refinement.

Dev Team

- Swarms/pairs on items.
- Presents completed PBIs to Product Owner.
- Updates Sprint Backlog.
- Updates Sprint Burndown.
- Product Backlog Refinement.

Scrum Master

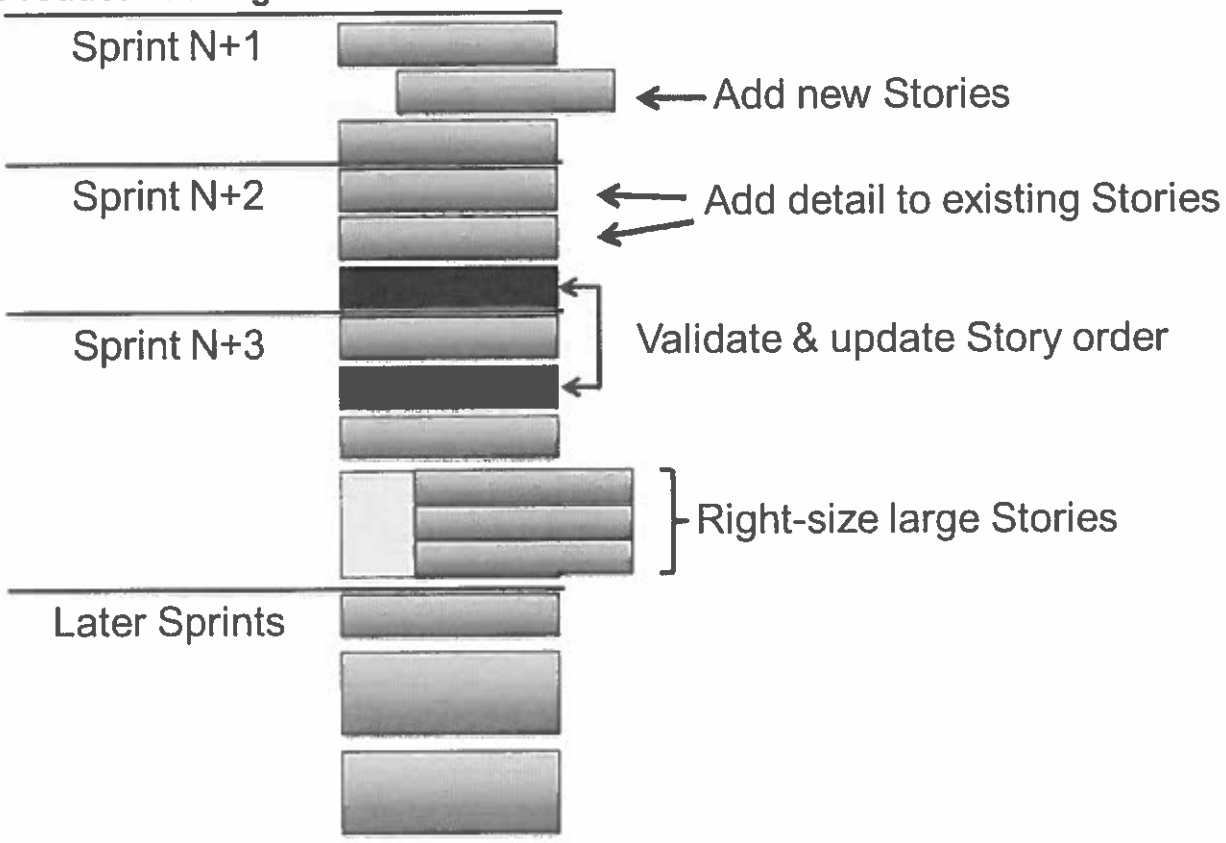
- Removes impediments.
- Teaches/coaches Team.
- Teaches/coaches org.



- Before we simulate the Sprint, let's take a look at what typically happens during a Sprint.
- During the Sprint, the Dev Team is working on Story tasks, swarming and/or pairing as necessary.
- The Dev Team seeks clarification from the Product Owner as needed.
- The Dev Team collaborates, communicates, updates the Sprint Backlog (Scrum Board) and burndown chart, and inspects and adapts at the daily Scrum.
- During the Sprint, as soon as the Dev Team is finished with a Backlog Item, they will present it to the Product Owner for an "over the shoulder check" to accept, reject, or suggest small tweaks.
- During the Sprint, the Scrum Master removes impediments that the Dev Team cannot remove, and coaches and teaches the Team and the organization on how to do Scrum well and how to benefit the most from Scrum's strategic benefits.
- When people are new to adopting Agile, one thing that they tend to gravitate toward is the meetings identified, because it's an area of comfort from a traditional process.
- There is a common occurrence for any of these meetings to run far beyond their intended length because Development Team members, Product Owners, and maybe even some new Scrum Masters are afraid it is one of the only times they will talk so all details must be fleshed out to resemble traditional project requirement documents, design documents, etc. This is not the case.
- Scrum prescribes Timeboxes for each of the Scrum events to discourage Teams from spending too much time in these events.

- Everyone is encouraged to collaborate all the time during the Sprint.
- This is why the emphasis is on co-location.
- If co-location is not possible, what technology is available to promote collaboration?

Product Backlog Refinement

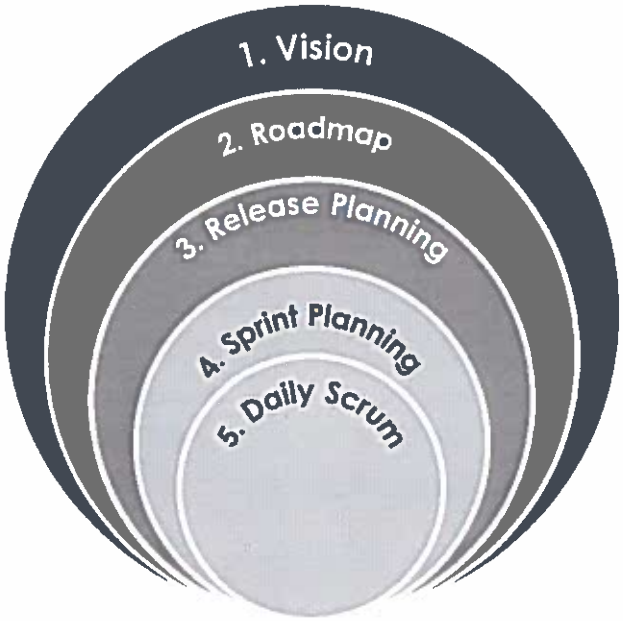


- The intention of Product Backlog Refinement is:
  - To identify and create new Stories.
  - To add detail to existing Stories.
  - To split large Stories into right-sized Stories.
  - To validate and update ordering.
  - To prepare for Sprint Planning.
  - To prepare for Releases and Release Planning.
- Backlog Refinement is continuous.
- Refinement is about the Stories. Stakeholders help the Product Owner create the Vision, which is then translated into Stories. Lead Technical Team members (architects) and Lead Test Team members sometimes are helpful in breaking down Stories and identifying gaps and dependencies for which the Product Owners might need help identifying.
- It is good practice to gather external inputs regularly.

The Daily Scrum Event

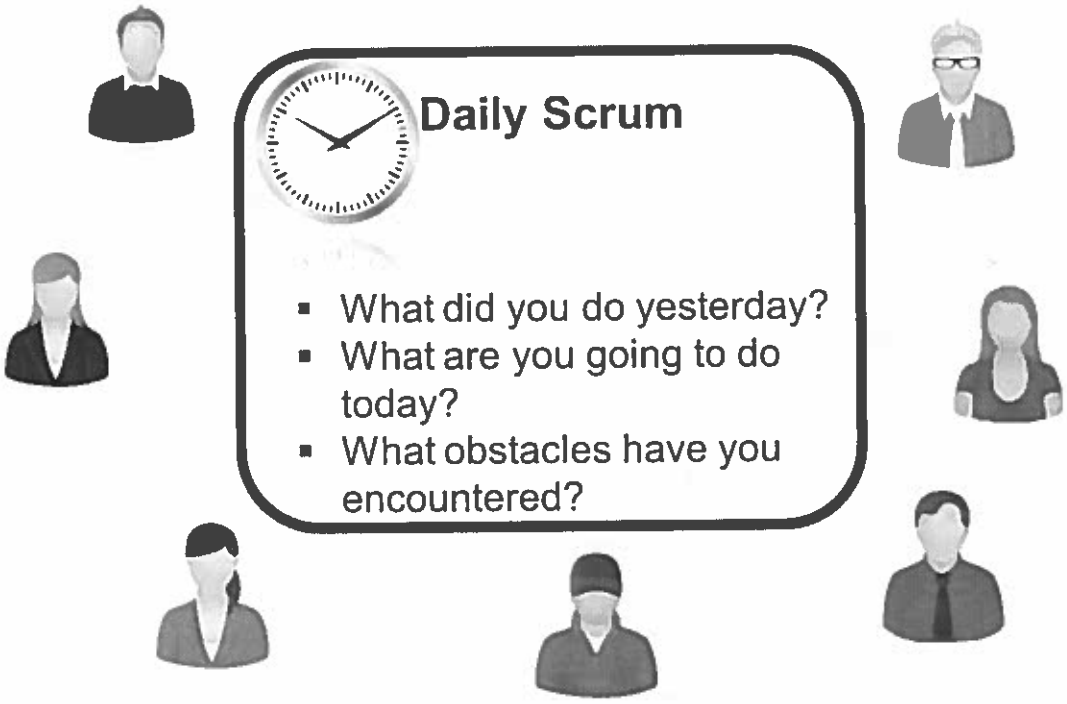
Daily Scrum:

- Daily Planning occurs at the Daily Scrum.
- **This is not a status meeting**, it is an inspect and adapt mechanism for the Dev Team.
- The Dev Team talks to each other about:
  - What they have completed.
  - What they are planning for that day.
  - Any challenges or impediments keeping them from achieving the Sprint Goal.
- This daily synchronization allows the Dev Team to make real-time course corrections to meet the Sprint Goal.



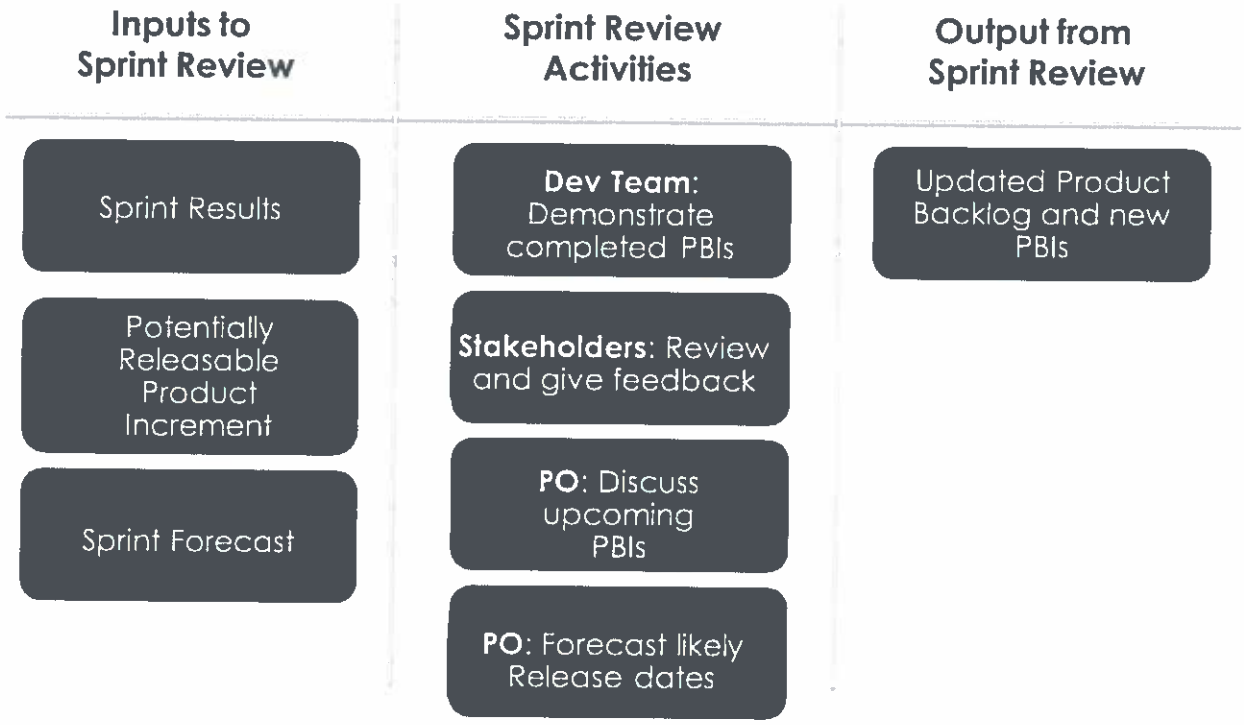
- The final layer of planning in Agile is at the daily level.
- In Scrum, we refer to this daily 'huddle" as the Daily Scrum.
- Other methods will refer to it as Daily Stand Up.
- Yes, participants are expected to stand up.
- The reason that we stand at this daily meeting is so that the 15-minute Timebox is respected.
- This is not a status meeting to the Scrum Master, the Product Owner, or management and it should not turn into one.
- The Development Team talks to each other.
- Again, this is not an executive or management forum but an opportunity for Development Team members to sync with each other on: What they accomplished yesterday, what they will focus on today, and what is blocking them or impeding them from getting to "done" or making progress.
- If a Development Team member raises an issue that someone else can help with, the conversation does not derail into the details about this.
- The Development Team member who can help may say, "Let's chat after the Scrum, I can help you with that."
- If the Development Team needs time with the Product Owner they may raise that but they don't have that detailed conversation at Stand Up; they ask if the Product Owner can come to the Team room or meet with them right after.

Daily Scrum



- This is an inspect and adapt mechanism for the Dev Team.
- This is NOT an executive forum or a meeting to provide status to a Project Manager or the Scrum Master.
- It is ideal for the Dev Team to meet all together, physically or visually (via video conference).
- The Scrum Master and Product Owner are completely optional at this event, though they often will attend as silent observers.
- Sometimes the Scrum Master and Product Owner will communicate with the Dev Team just after the Daily Scrum (sometimes known as the "After Meeting" or the "After Party").
- Some Development Teams meet around their Sprint Backlog/task board, and this is fine so long as they don't spend most of their Daily Scrum "updating" the Sprint Backlog (turns the Daily Scrum into a waterfall status meeting).
- The Daily Scrum is held at the same time and place every day, to reduce complexity.
- If someone is always late or always misses it, the Development Team should have an honest and open conversation about this and talk about what would be the best time for everyone to make it.
- This is a forum for the Development Team to be accountable to themselves and to each other.
- Ideally, this is held by the Development Team's task board so that Team members can move their own sticky notes, update them, speak to them, etc.

Sprint Review

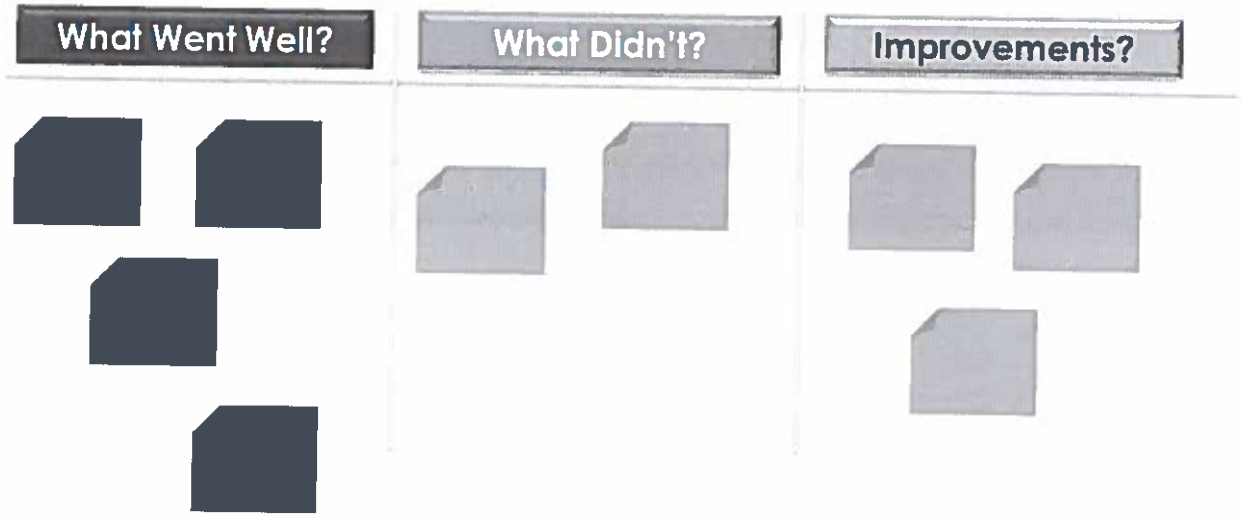


- The Dev Team demonstrates (only) completed Product Backlog Items during the Sprint Review to the stakeholders.
- Any incomplete items go back into the Product Backlog, where the PO can choose to carry them over or reorder them as needed.
- Feedback is captured in new Product Backlog Items or User Stories, and added to the Product Backlog as necessary.
- The Review should not be a surprise to anyone; preparing for this demonstration of a working product increment should be taken into account during the Sprint.
- The Sprint Review is not the first time we ask the Product Owner for feedback or have demonstrated the working product. That should occur as needed throughout the Sprint.
- The Product Owner should be the lead facilitator in the discussion with the key stakeholders in Sprint Reviews, discussing the present state of the Product Backlog and what will be worked on in the coming Sprints.
- The Product Owner discusses the progress toward milestones/Releases, and forecasts likely completion/Release dates, in a communication for key stakeholders. A lot of Teams forget to do this.
- The Scrum Master has no role other than to teach and coach how to hold a Sprint Review.
- It is important to understand that this does not imply this is the first and only time the Product Owner would be seeing the completed Product Backlog Items.
- He or she should have been involved in the process throughout and already had an opportunity to provide feedback so there is still time for the Team to respond to this feedback within the Sprint.

- The Product Owner invites the key stakeholders and customers.
- The Review is not a PowerPoint presentation but is a demo of working software. It also does not need to be lengthy.
- Any feedback received at the review in the way of new functionality, features, etc., will be captured by the Product Owner and ordered in the Product Backlog along with everything else.
- There should not be an assumption that feedback items will automatically be worked on at the next Sprint.
- The process is to put those ideas on the Product Backlog and evaluate at or by Sprint Planning.
- This is why it's important that the Product Owner be involved prior to the final review.

Sprint Retrospective

An inspect and adapt mechanism at the end of each Sprint that allows the **Scrum Team** to improve their effectiveness.



- What went well?
  - Celebrate good Team behavior. Recognize simple but effective things.
  - Were there any experiments to improve that went well?
  - Focus on the Scrum Team and how they work together to deliver value.
- What didn't go well?
  - Identify impediments to the Scrum Team's performance. If not removable by Team, then have the Scrum Master attack these.
  - Were there any experiments to improve that didn't go so well?
  - Try to focus on those items within the Scrum Team's control, and turn over the rest for the Scrum Master to do as part of their "Impediment Removal" responsibility.
- What will we improve?
  - Focus on practical changes, and be realistic in choosing the 1-2 improvement items the Scrum Team can actually change in the upcoming Sprint.
  - Beyond the 1-2 items, keeping a "Retrospective Backlog" or "Improvement Backlog" of good ideas and suggestions allows the Scrum Team to revisit those items for future Sprints.
  - The Scrum Team decides which items they will dedicate themselves to trying for the next Sprint. Some people call these "improvement experiments."
  - Sprint Retrospectives are worthless if nothing changes, so it's important for the Scrum Team to plan time into the coming Sprints to work on improvement items. Consider this as part of the capacity discussion in Sprint Planning.
  - The Sprint Retrospective should not just be about "who said what," "meeting minutes," or "good, bad, and ugly." The focus should be on choosing improvements to make.

  - Many Scrum Teams don't spend the time needed in Retrospectives to get to "root causes." Instead, they try to solve "surface" problems that often don't yield much improvement at all. Get to the root!

- The Scrum Master optionally facilitates this session.
- It occurs after the Sprint Review, but before the next Sprint Planning session, so the Scrum Team can inspect and adapt as necessary and "course correct" in real time.
- The Scrum Team makes improvements *within* the Scrum Framework. The Team is not allowed to make changes to the Scrum Framework itself.
- Where the Scrum Framework provides flexibility (estimating, forecasting, Product Backlog management), Teams are encouraged to experiment, inspect, and adapt their techniques, while respecting the Scrum Framework principles and practices.
- Esther Derby and Diana Larsen identified an approach for facilitating this session in "Agile Retrospectives":
  - Set the Stage.
  - Gather Data.
  - Generate Insights.
  - Decide What to Do.
  - Close the Retrospective.

Assessment: The Sprint

Questions:

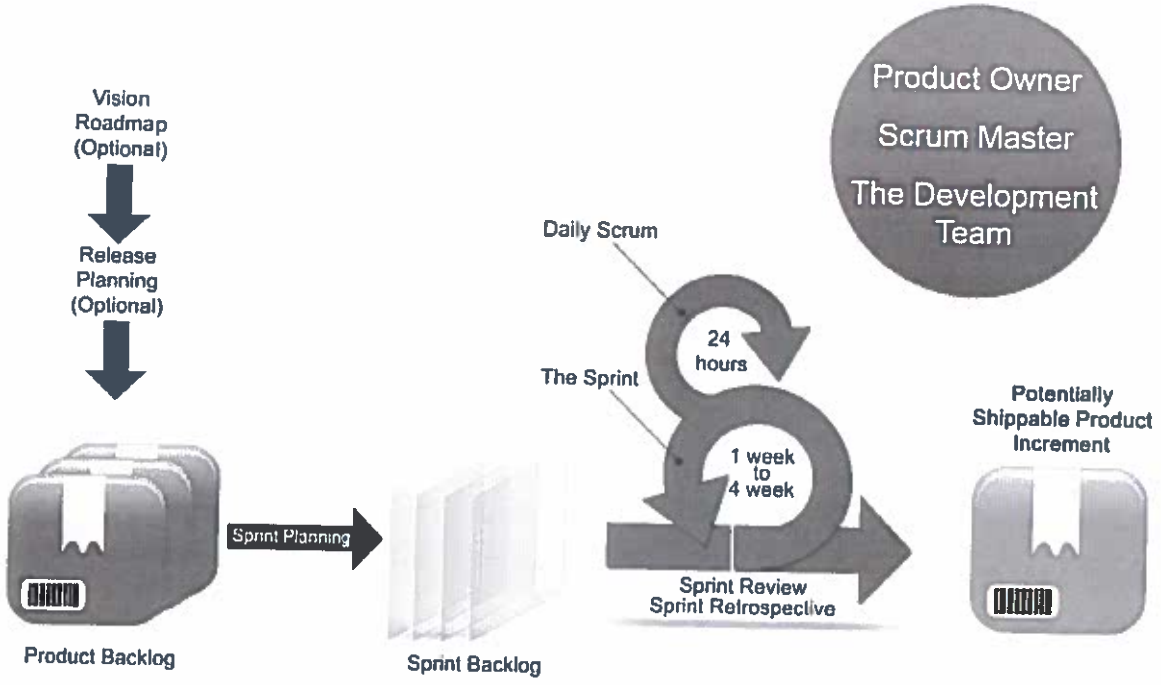
1. True or False? Outputs from the Sprint Planning event are the Sprint Goal, the Sprint Forecast, and the Sprint Backlog.
2. Capacity is how much time the Development Team has available to work during the Sprint, including known holidays, time off, etc.
   A. Velocity
   B. Capacity
3. True or False? The Sprint Review is the inspect and adapt mechanism at the end of each Sprint that allows the Scrum Team to improve their effectiveness.
4. During the Daily Scrum, the Development Team answers which questions?
   A. What did you do yesterday?
   B. What are you going to do today?
   C. What help do you need?
   D. Did you solve any problems?
   E. What obstacles have you encountered?

Retrospective

TEKsystems
Our people make IT possible.

Practice Simulations

Scrum Flow



- This view provides a holistic view of the extended Scrum process.
- This may seem like a lot to understand all at one time, but this view is intended to provide the "big picture" only.
- We will be covering each of these areas in detail in this course.
- Please note that the objective of each Sprint is to deliver "working software"—even if it is an increment.
- Release Planning is an optional event and is not considered one of the "core" events in Scrum.
- Core Scrum events include: Sprint Planning, Daily Scrum, Sprint Review, and the Sprint Retrospective.
- A Potentially Releasable Product Increment that functions is still considered valuable—even if we save those increments to be released to production together after a certain number of Sprints (a.k.a., Release).