

Exploring Artifacts



ROAD TO MASTERY
VIRTUAL EDITION
FOR GUIDES



Exploring Artifacts

LET'S RAID SOME TOMBS

Exploring Artifacts

Let's go canyoning, caving, and tomb raiding. Let's hunt for valuable artifacts and find lost treasures. This adventure is not without risk! There will be unknown unknowns. Mysteries. Traps. Trolls?

I hope you brought your flashlights and spare batteries.

This adventure is about the three artifacts in Scrum: the Product Backlog, Sprint Backlog, and Increment. We may refer to participants as explorers, canyoneers, or tomb raiders. During this adventure, our canyoneers are on their own without a guide. However, they are accompanied by a curious bat who habitually drops helpful messages...

- Make work visible;
- Managing the artifacts in practice;
- Busting some ancient myths;
- Bring your understanding and practice concerning artifacts up to date;
- Techniques for effective Product Backlog Management;
- Techniques for effective Stakeholder Management;
- Applying Kanban as a strategy to visualize workflow;
- Determining what constitutes an increment of value and what doesn't;
- Co-creating a Definition of Done.



Illumination

When two individuals share truth, illumination happens. Like two flames joining, they produce a brighter light. Through this, they can accomplish something greater than themselves. Their fire produces inspiration, opportunity, energy, and momentum.

Fire can spread. When these individuals make their work visible, they spread their light. They illuminate their ambitions. This way, providence happens. Brightness reveals opportunity. It shows the way which lends power to their endeavor.

Artifacts in Scrum are there to spread the illumination of truth. They exist to provide a real-time picture of our true actions. Energy, like fire, can spread. It can also die out. People, too, can burn out. So, keep a sustainable fire alight.

The energy is kept alive by fueling it with truth through transparency, inspection, and adaptation. Without it, there is darkness and obscurity.



Bat Signals

Say it with symbols. Pictures can be better than words. Let's create some Bat Signals. This play is similar to the drawing game 'Pictionary' where a participant draws a card with a word on it and then draws a picture. Other participants then have to guess from the picture what is written on the card.

This play is also helpful for practicing drawing and visualization. Anyone can draw. Consider drawing icons from elementary shapes like:

1. Straight line
2. Squiggly line
3. Curved line
4. Circle
5. Oval
6. Square
7. Rectangle
8. Diamond
9. Triangle
10. Spiral

The guide (bat) prepares a set of cards (hidden from participants)

1. Transparency
2. Product Owner
3. Sprint Backlog
4. Increment
5. Inspection
6. "Done"
7. Stakeholder

30 seconds
(per turn)

Taking turns, a volunteer picks a card and looks at it. They then draw a picture/icon representing what's written on the card while other participants guess and shout out what they believe is drawn.



\circlearrowuparrow Ψ
 Ψ \circlearrowleft
 \circlearrowleft \circlearrowuparrow
 Ω \rightarrow
 \circlearrowuparrow Ψ

Solve the Riddle

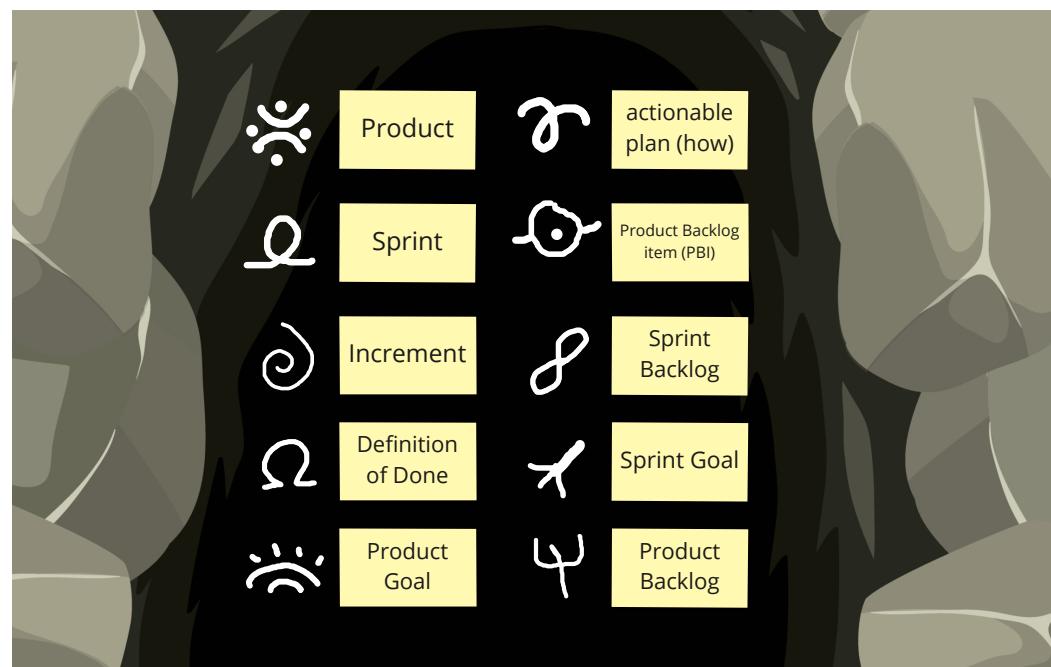
Speak friend and enter. Only those worthy may enter these caves, tombs, and canyons. This adventure will test both the guide and the traveler. Our explorers must work their way through with as little guidance as possible. That includes you, the guide, but also the Scrum Guide! Can they do without? Can you, the guide, stay in the shadows?

You may travel along as a *bat*. You may only speak to the explorers during the entire adventure if you know no other way of responding to their needs. As a guide, sorry, bat, you may use body language and write bat signals, but only when necessary. Virtually, you may mute yourself, but taping your mouth shut may send a clearer visual message.

“Solve the Riddle” can work well as a Fast-Pass. It gives our explorers something to figure out as they enter the (virtual) room.

10 minutes

Can the explorers solve this puzzle without guidance? (this includes not consulting the Scrum Guide). As a bat, you may send a signal when they have solved the





SCRUM'S ARTIFACTS REPRESENT

OR.

THEY ARE DESIGNED TO MAXIMIZE

OF KEY INFORMATION.

THUS, EVERYONE

THEM HAS THE SAME BASIS FOR



value

adaptation

work

inspecting

requirements

transparency

Pass the Guards

Ugh, once again the way is blocked. These guards won't let them pass without another test. It's a "Fill in the Blanks". The guards are friendly enough to give the explorers the answers. But unawares to the explorers, one is a trap! You want to warn them, but, well, you're a bat. Will they fall for it? What's more, the answers must fit the correct slot. Any mistake and...

2,5 minutes

Tick, tock. The explorers only get 2,5 minutes to pass the guards.

The bat knows the correct answer reads:

"Scrum's artifacts represent *work* or *value*. They are designed to maximize *transparency* of key information. Thus, everyone *inspecting* them has the same basis for *adaptation*."

Requirements is the trap. The placement must be correct.

If the explorers pick either the trap (which gets triggered instantly) or fail to correctly fill in the blanks within the 2,5 minutes, one explorer must share a personal story about a past failure. You, the bat, may send a bat signal to send this message to the explorers.



Each artifact contains a **commitment** to ensure it provides information that enhances transparency and focus against which progress can be measured:

- For the Product Backlog it is the ...
- For the Sprint Backlog it is the ...
- For the Increment it is the ...

Product Goal

Sprint Goal

Definition of Done

Supporting Each Other

There is one more commitment in Scrum.
The Scrum Team commits to



It's completely up to the canyoneers.
There is no right or wrong.

Together write down at least 5 things to make **visible** in Scrum.
For each of these, write down 4 possible tools, plays, or techniques to do so.



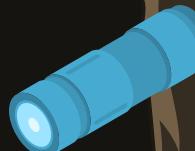
efficacy of current management



environment



work



The (emergent) process



work techniques

Made visible in/through

Ordering and Refinement of the Product Backlog items

Definition of Done

Increment

Sprint Backlog

Improvement Plan

Accountability Autonomy Matrix

Radar

Sprint Backlog

Kanban Board

Sprint Backlog

Increment

Magic Wand

Definition of Done

Definition of Done

Under the Rug

Impact Craters

Compass

Sizing Canvas

Value Stream Map

Increment

- *What are good practices for Product Backlog Management?*

Ordering

Sizing

Refinement

*It's completely up to the canyoneers.
There is no right or wrong.*



- *What strategies or techniques help make visible the progress towards the Sprint Goal in the Sprint Backlog?*

Kanban

Radar

Burndown

Daily Scrum

Impediment
Wall

Timeline
(Map)



Checkpoint 1

Hanging from the ceiling offers us a fresh perspective on our canyoneers.

How are they managing themselves? And are you, their guide, really staying in the back of the cave? The canyoneers passed some tricky tests. They learn about the value of light in dark places. Are they doing an excellent job at making their progress visible?



5 minutes

1. What have the previous activities brought to light?
2. Did you stumble upon any traps? If so, which?
3. Write down three advantages of making work visible.

Now our canyoneers are deeper down, they'll encounter some ancient scrolls and artifacts. Can they tell what's on it?



**LOOKS ANCIENT
TO ME!**

WHAT'S WRONG?

The Product Backlog is a prioritized requirements document.

*When multiple Scrum Teams work on a single product,
they each have their own Team Product Backlog.*

*The Product Owner is the sole person managing the Product Backlog,
including its content, budget, delivery dates, assignees and estimations.*

*The Product Backlog consists exhaustive functional
and technical instructions.*

*The Product Backlog must contain
all stakeholder requirements.*

Ancient Artifact: Product Backlog

The canyoneers encounter a very ancient and incorrect definition of a Product Backlog. Is this really what people used to think in this organization cave?

15 minutes

In triads or quads, our canyoneers figure out what's wrong with these statements. Can they rewrite it? (The Scrum Guide may be consulted).

Let's see if the canyoneers can find and correct the anti-patterns. You might share some conclusions as a guide once the canyoneers return from their break-out caves.

- The Product Backlog is ordered, not prioritized. Priority may be a factor to consider when ordering Product Backlog items.
- The Product Backlog is not a requirements document. It contains items the Scrum Team believes are needed for the Product and to achieve a Product Goal.
- Although the Product Owner is accountable for ensuring that the Product Backlog is transparent, the Product Backlog may be managed by the entire Scrum Team. Items in the Product Backlog remain the accountability of the entire Scrum Team and should not be individually assigned. It's the developers that may size the work needed for an item. Only they can determine if it fits in a Sprint. Although possible, establishing delivery dates in a Product Backlog may work against empiricism, agility, predictability, and quality.
- Product Backlog items that can be “Done” by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. By no means does this need to be exhaustive, instructive, or fully specified. The Scrum Team may come to this understanding by working through the complex problem toward a solution.
- Although the Product Backlog is a single source of work undertaken by the Scrum Team, it does not need to contain everything asked for by stakeholders. A transparent Product Backlog is short and focused on a Product Goal. New needs may emerge into the Product Backlog in a timely manner.



In... or Out?

The Product Backlog contains what is needed to improve the Product. What that may be and how that is defined will vary with the domain of work. It's up to the Scrum Team to determine what is needed for them.

As a bat (guide), prepare a list of things one might find in a Product Backlog. Whether these should or shouldn't be there will be up for our canyoneers to contemplate. Do not reveal these to the canyoneers just yet.

3 minutes

Invite the canyoneers to individually write down as many things they list that one may find on/in a Product Backlog. Whether it's a good or bad thing (depending on context) is irrelevant now.

3 minutes

Pair up. The canyoneers reveal their lists and consolidate them into a single list by removing duplicates and grouping similar items.

3 minutes

Merge pairs (to form quads). Once again, consolidate the lists.

3 minutes

All together, create one master list. (The bat may add some, too, after the canyoneers are done). Now ask the group to dot-vote items they believe are generally considered NOT good to have in a Product Backlog.

5 minutes

Going through the items, from most dots to least, participants share why they believe they shouldn't be part of a Product Backlog. There are no right or wrong suggestions.

Context matters. It's ultimately up to the Scrum Team to decide what goes in or out. That may depend on situational context. Ultimately, it's whatever is needed for the developers to have enough transparency for an item to be deemed ready for selection in a Sprint Planning event. Throughout the Sprint, more will be learned. Refinement is an ongoing activity. Often, a good way to understand the problem, is to work through it. Remember, simplicity is the art of maximizing the work NOT done.

Product Backlog

Our Product Goal is to achieve...
[this outcome] for [these people]

Order	Value	Validation	Size
#1	We [the Scrum Team] believe that [doing this], for [these people], will achieve [these outcomes].	We will know [outcome A] is achieved when we see [this behavior/response]. We will know [outcome B] is achieved when we see [this behavior/response].	XS
#2	We [the Scrum Team] believe that [doing this], for [these people], will achieve [these outcomes].	We will know [outcome A] is achieved when we see [this behavior/response].	XS
#3	We [the Scrum Team] believe that [doing this], for [these people], will achieve [these outcomes].	We will know [outcome A] is achieved when we see [this behavior/response].	S
#4	We [the Scrum Team] believe that [doing this], for [these people], will achieve [these outcomes].	We will know [outcome A] is achieved when we see [this behavior/response]. We will know [outcome B] is achieved when we see [this behavior/response].	M
#5	We [the Scrum Team] believe that [doing this], for [these people], will achieve [these outcomes].	We will know [outcome A] is achieved when we see [this behavior/response]. We will know [outcome B] is achieved when we see [this behavior/response].	M

Product Backlog Template

To the left, our canyoneers can see an example (for training purposes) of what a Product Backlog may be constructed of. Product Backlogs may be managed in various ways. Its items may be described and visualized using various formats given what makes the most sense for that specific item. Applying a technique, format, or template just for consistency may not always be the right thing to do.

4 minutes

Backlog Management techniques they believe are reflected in this template:

1. Prioritization
2. Ordering
3. User Stories
4. Sizing
5. Hypothesis Statements
6. Value Pyramid
7. Story Mapping
8. WSJF (Weighted Shortest Job First)

(Answer: 2, 4, 5)

Order or Priority

4 minutes

Pair up and answer: What's the difference between order and priority? What factors other than priority may weigh in when ordering a Product Backlog?

Ten items may be of urgent priority, but only one can be ordered first. Ordering means that the Product Owner must make decisions.

Product
Goal

PBI

The Spotlight

Keeping the Product Backlog short, clear, and focused is generally a good idea. The Product Goal helps achieve that. It serves as a spotlight. If (new) needs are not in the Spotlight, they may not (yet) be needed in the Product Backlog. Refine just-in-time.

“The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define “what” will fulfill the Product Goal.”
– The Scrum Guide.

There may only be one active Spotlight, as the Scrum Team must fulfill (or abandon) a Product Goal before taking on the next. This Spotlight can also be a visual way to show stakeholders what the Scrum Teams focus on. It helps to clearly illustrate the decisions a Product Owner makes regarding what goes in or out of the Product Backlog and how the items (PBIs) are ordered.

4 minutes

Once again, our canyoneers pair up. Together, list other examples of when using a Spotlight can be a helpful visual play/aide.

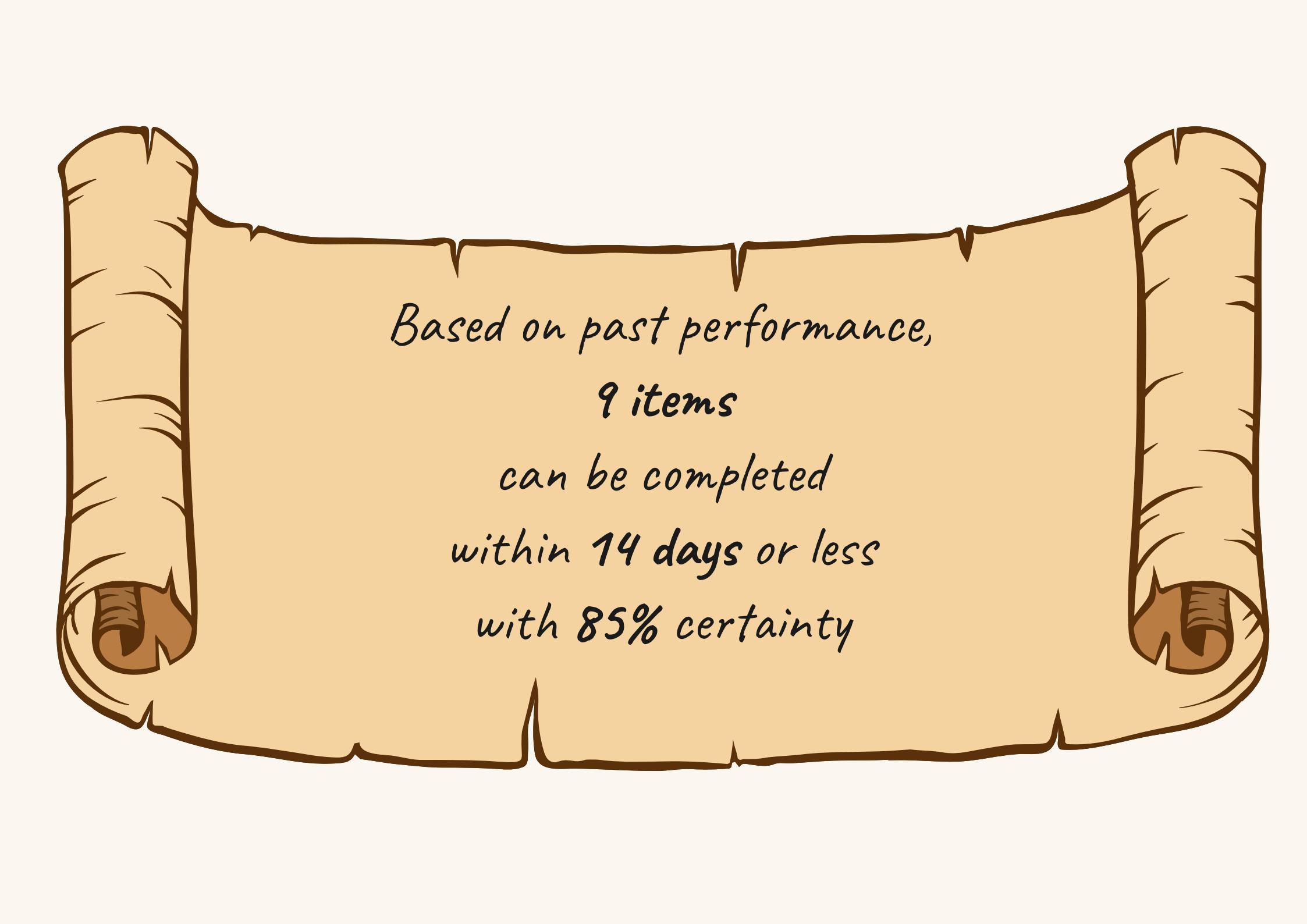




COMING THROUGH!

OUT THE WAY!

MY TURN!



Based on past performance,
9 items
can be completed
within 14 days or less
with 85% certainty

Probabilistic Forecasting

Here's Tom with the weather!

"Only what has already happened may be used for forward-looking decision making." – The Scrum Guide.

A Probabilistic Forecast is a calculation about predicting the completion of items that includes a probability. Probability is not a certainty. It is based on historical data (past performance). It's evidence. There are various possible calculations and ways of visualizing and communicating the forecast. It calculates what's completed in similar timeframes and runs simulations (such as Monte Carlo).

The purpose of using probabilistic forecasting is NOT to increase a team's predictability! It is to increase transparency and encourage empirical product planning. In Scrum, there needs to be a (healthy) degree of unpredictability and inconsistency to allow for creative, adaptive, complex problem-solving. So beware! a Probabilistic Forecast may make managers uneasy. A Probabilistic Forecast may be juxtaposed with a *wishful thinking* roadmap. It might reveal an expectation gap enabling the organization to align and adapt its expectations timely.

4 minutes

Invite participants to pair up and list what historical data and past performance are available to them that can be used for forward-looking decision making.

4 minutes

Merge pairs into quads. Consolidate the lists, group similar items, and remove duplicates.

4 minutes

All together. Consolidate the lists, group similar items, and remove duplicates.

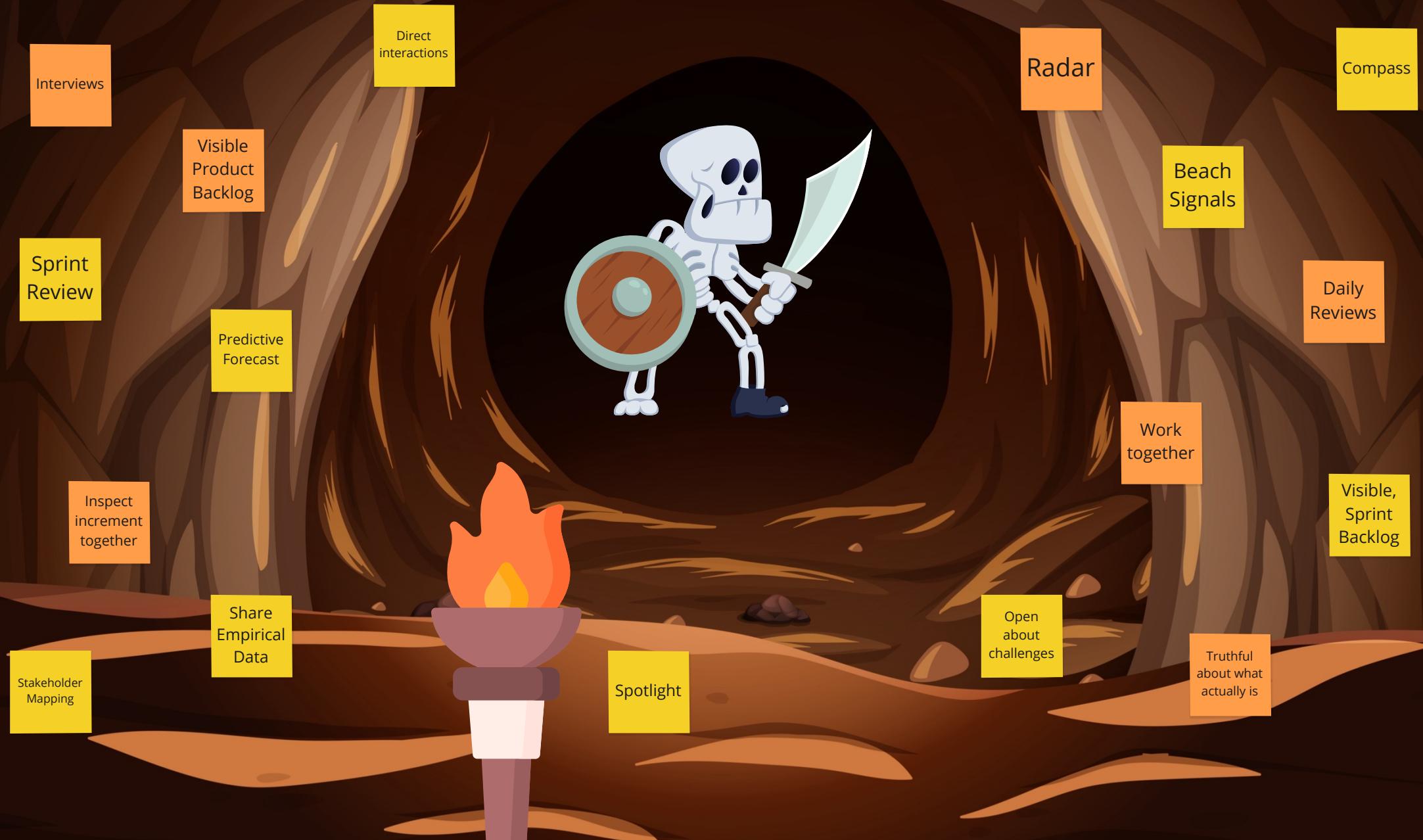
4 minutes

Ask the canyoneers to write down the numbers of the items in the list that can impact the accuracy of a forecast (either positive or negative) of when items may be delivered.

1. Re-ordering items in the Product Backlog;
2. Refinement of Product Backlog items;
3. Size of Product Backlog items;
4. Switching work between Product Backlog items;
5. Number of Product Backlog items in progress;
6. Number of Product Backlog items in the Product Backlog;
7. Shared understanding of items in the Product Backlog.
8. Balance between output (completed items) and input (new items).

(answer: all)

How can you keep stakeholders from being 'in the dark'?



Stakeholder Management

“Stakeholder: a person external to the Scrum Team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the Product Owner and actively engaged with the Scrum Team at Sprint Review.” — Scrum.org Scrum Glossary

Please don't assume, but research your stakeholders and how to best involve them. The Scrum Team should be in the lead and not be led by its stakeholders. That requires proactive engagement from Scrum Team members.

1 minute

Ask the group to collectively brainstorm various types of stakeholders.

As a guide, you may share these examples (if not yet shared by the group) after one minute has passed.

1. Users (consider User Panels or Focus Groups)
2. Customers
3. Subject Matter Experts (SME) outside the Scrum Team
4. Sponsors / Investors
5. Executives
6. Scrum Team members have a stake in the Product too.

The network of stakeholders can be complex. There may be many types of Stakeholders.

5 minutes

Pair up. Invite canyoneers to list various techniques that keep stakeholders from being in the dark. In what ways can expectations and needs be managed in ways that are true to empiricism? Think of plays and techniques that you learned about in the Road to Mastery.

3 minutes

Merge pairs into quads. Consolidate the lists. Group similar items and remove duplicates.

6 minutes

All together, consolidate the lists. Group similar items and remove duplicates. Review the list together.



Stretch Break: Dancing in the Dark

Play a (short) dance song for one minute.
Everyone may turn off their camera.

Now dance like no one is watching!
until the minute runs out.

All stretch breaks are invitations.
Participants may pass.

Participants may perform alternative stretches too.

Checkpoint 2



Being in the dark can be scary. That's why people may be anxious and cautious. The brighter the light, the more we can be at ease. We may open up. When we can trust our footing, we may dare to move faster.

The Product Backlog is a means to cast light onto what otherwise would be a dark and obscure future.

5 minutes

1. What have the previous activities brought to light?
2. Which Product Backlog management technique will you try first?
3. Which Stakeholder management technique will you try first?

Let's continue our stroll, but let's avoid trolls. We're getting closer to finding hidden treasures. Our canyoneers encounter another mysterious backlog artifact: "The Sprint Backlog".



WHAT'S WRONG?

The Sprint Backlog is a complete plan
of how all the work will be completed during the Sprint.

The progress of this plan is reported in the Daily Scrum.

As new work is required, the Product Owner adds it to the Sprint Backlog
so the Development Team timely responds to emerging requirements.

Only the Product Owner can change the Sprint Backlog during a Sprint.

The Scrum Master makes sure the Developers
keep the Sprint Backlog up to date.

Ancient Artifact: Sprint Backlog

The canyoneers encountered yet another ancient and incorrect artifact. It's almost unrecognizable.

15 minutes

In triads or quads, our canyoneers figure out what's wrong with these statements. Can they rewrite it? (The Scrum Guide may be consulted).

Let's see if the canyoneers can find and correct the anti-patterns. You might share some conclusions as a guide once the canyoneers return from their break-out caves.

- The Sprint Backlog contains not a complete plan, but an emergent plan. It is updated throughout the Sprint as more is learned. This plan is geared towards a Sprint Goal.
- The purpose of the Daily Scrum is not to report progress. It is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary.
- The whole Scrum Team stays focused on the Sprint Goal. Developers may collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.
- It is not the Product Owner, but the Developers who do the work, who may adapt the Sprint Backlog as more is learned.
- Developers are accountable for managing the Sprint Backlog so that it remains a highly-visible real-time picture of the work undertaken. The Scrum Master may coach them.

THE SPRINT BACKLOG IS A

BY AND FOR THE DEVELOPERS

IT IS A HIGHLY

THAT THE

PICTURE OF

PLAN TO ACCOMPLISH DURING THE SPRINT

IN ORDER TO ACHIEVE THE

THE SPRINT BACKLOG IS UPDATED THROUGHOUT THE SPRINT AS MORE IS

valuable

real-time

sprint goal

product goal

Developers

Scrum Team

learned

delivered

Planned

visible

the work

plan

performance

Unlock the Shrine

We're getting closer to unlocking value. There are no guards here, but our canyoneers shouldn't let their guard down. They need to find the right match for the missing words. There are more options than slots. That means some must be trapped and should remain untouched.

2,5 minutes

Tick, tock. The explorers only get 2,5 minutes to unlock the shrine.

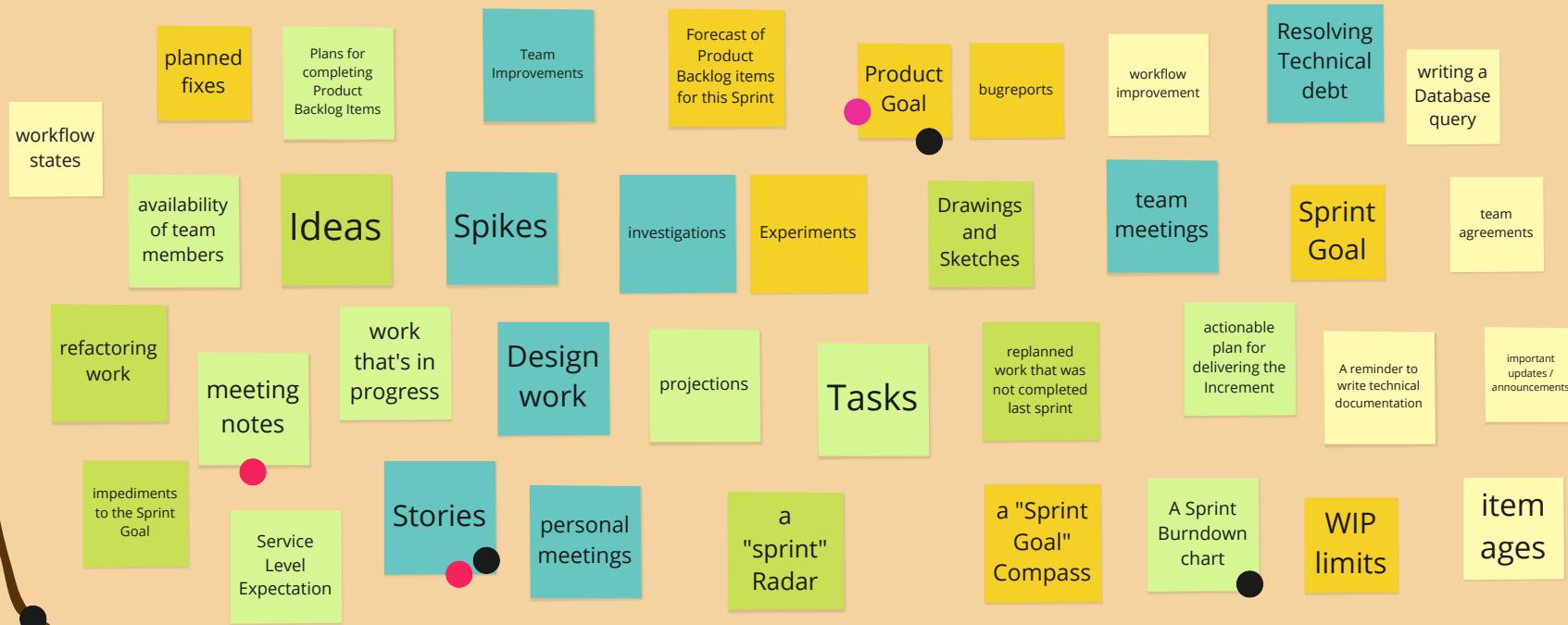
The bat knows the correct answer reads:

The Sprint Backlog is a *plan* by and for the Developers. It is a highly *visible, real-time* picture of *the work* that the *Developers* plan to accomplish during the Sprint to achieve *the Sprint Goal*. The Sprint Backlog is updated throughout the Sprint as more is *learned*.

valuable, product goal, Scrum Team, delivered, planned, and performance are all traps. The placement must be correct.

If the explorers pick a trap (which gets triggered instantly) or fail to fill in the blanks within the 2,5 minutes correctly, one explorer must share a personal story about a past failure. You, the bat, may send a bat signal to send this message to the explorers.

Sprint Backlog



In... or Out?

A Sprint Backlog may contain anything developers plan to do to achieve the Sprint Goal and create value. The granularity of the work should not be so granular that it gets in the way of the work, nor should it be so broadly defined that the work and progress become obscure.

As a bat (guide), prepare a list of things one might find in a Sprint Backlog. Whether these should or shouldn't be there (which depends on context) will be up for our cave-dwellers to contemplate. Do not reveal these just yet.

3 minutes

Invite the canyoneers to individually write down as many things they list that one may find on/in a Sprint Backlog. Whether it's a good or bad thing (depending on context) is irrelevant now.

3 minutes

Pair up. The canyoneers reveal their lists and consolidate them into a single list by removing duplicates and grouping similar items.

3 minutes

Merge pairs (to form quads). Once again, consolidate the lists.

3 minutes

All together, create one master list. (The bat may add some after the canyoneers are done) Now ask the group to dot-vote items they believe are generally considered NOT good to have in a Sprint Backlog.

5 minutes

Going through the items, from most dots to least, participants share why they believe they shouldn't be part of a Sprint Backlog. There are no right or wrong suggestions.

Context matters. It's ultimately up to the Developers to decide what goes in or out. That may depend on situational context.



Visualizing Workflow 1/2

Let's make work visible. A Sprint Backlog is sometimes also called a Scrumboard. Many Scrum Teams apply Kanban to visualize their work(flow). There are numerous ways to visualize the workflow. Let's start with a basic grid example to outline a workflow and how to read it.

3 minutes

Present the canyoneers with the grid to the left but without the corresponding definitions in the sticky notes. Can the canyoneers identify the elements of this grid?

3 minutes

Reveal three sticky notes with the definitions. Can the canyoneers map these to the correct boxes? The bat may correct the placement when applicable.

3 minutes

The group collectively brainstorm examples of workflow states.

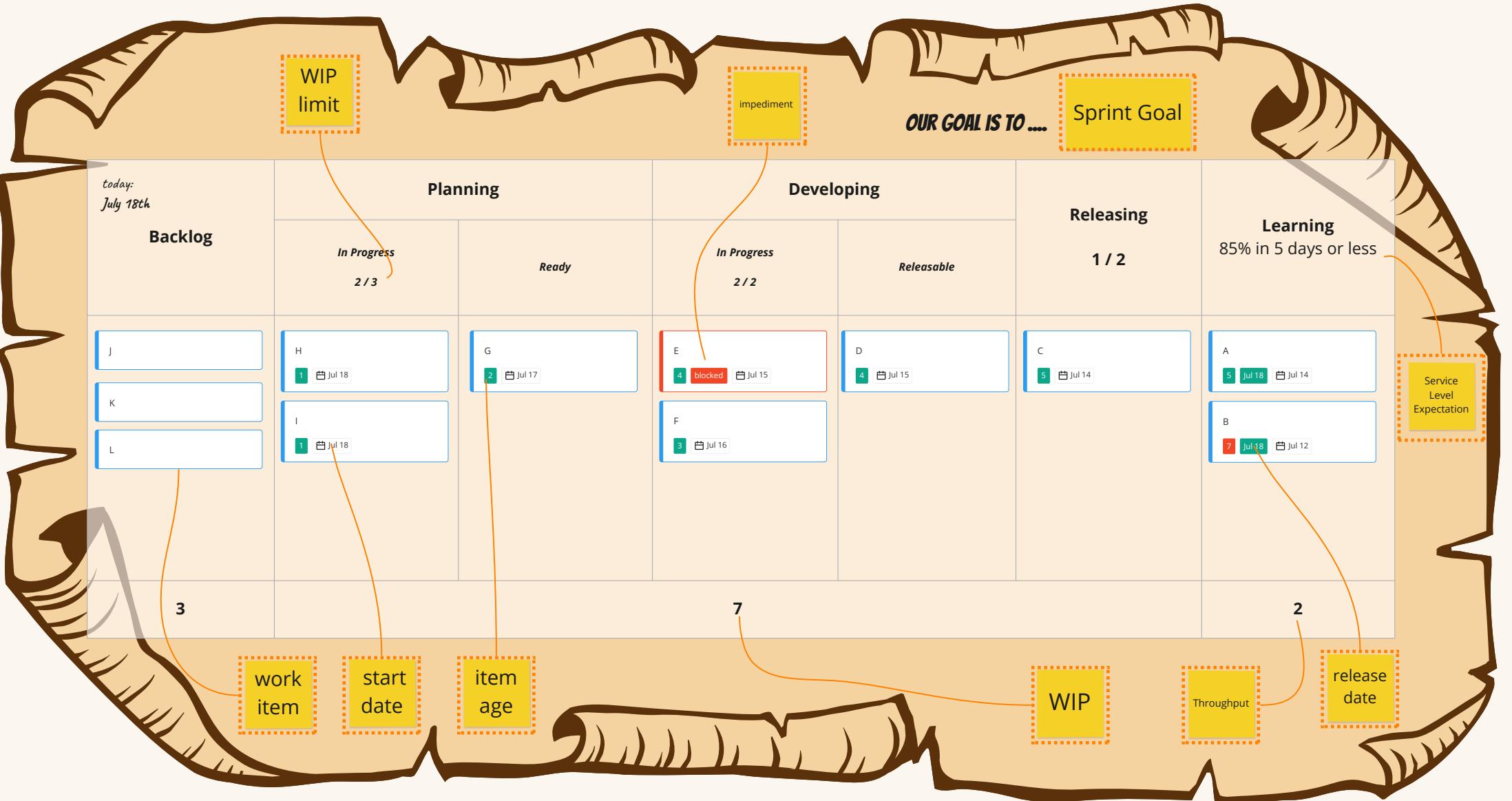
3 minutes

The group collectively brainstorm examples of item states.

3 minutes

The group collectively brainstorm examples of uses of (swim)lanes.

In Scrum, Developers work on their outline. They might use this grid or try something entirely different. It's generally a good idea to start with a simple workflow outline and revisit it frequently. The Sprint Retrospective provides regular opportunities, but the workflow may be adapted at any time if any aspect is deemed unacceptable by the developers. To gain shared ownership and accountability of the workflow, Scrum developers must be in control. A Scrum Master can guide them and show them various ways and techniques. The proof is ultimately in the pudding. Try it, understand it, inspect it, adapt it.



Visualizing Workflow 2/2

Let's add a workflow and work items to our grid, including some visual aids and indicators for progress. Remember that this is just a board for training purposes. Explain to the participants that we are not discussing the applicability or relevance of this workflow.

10 minutes

Prepare boards for triads or quads. Invite the canyoneers to map the names of the various visual workflow elements to the corresponding boxes. They may discuss what they already know about it. All together, review the various boards. Observe if the canyoneers can self-correct any misplacements. If any remain incorrectly placed, the bat may move/switch them.

Workflow Glossary:

1. WIP stands for "Work(items) in Progress". It relates to the number of items in a certain workflow state, or across the entire workflow.
2. A WIP limit is the maximum number of work items allowed in that particular state, or across the entire workflow.
3. Work can be either blocked or impeded. It is generally a good idea to add a visible indicator on the item in the workflow state in which it is. It is generally not a good idea to have 'blocked' as a workflow state as it obscures where the item is in the workflow it got blocked.
4. Throughput: the number of items that reached the end of the workflow over a given amount of time (for example a Sprint).
5. Service Level Expectation is a forecast of how long a given item should take to flow from start to finish within the Scrum Team's Workflow. Ideally, this is based on historical data of past performance to reflect probability.
6. Item Age shows the number of days an item has entered the workflow. The count starts with "1" the moment it enters the workflow. The count stops when it has reached the end of the workflow.
7. Release date (may in other workflows also be called Completion Date or Done Date) defines the date an item reached its final state in the workflow. In this example, it is when the item is released, and learning starts.
8. A Work Item must represent individual units of value (stakeholder value, knowledge value, process improvement value). These are ideally not individual actions or tasks specific to a single competence. These are never individually assigned, and the whole Scrum Team remains accountable. These are generally Product Backlog items or a breakdown thereof.

Invite participants to read the glossary individually. Then check for understanding using thumbs up or down.

1. WIP limit means a team may not take on fewer work items than indicated. (false, they may not take on more)
2. It is generally not a good idea to have 'blocked' as a workflow state. (true, use a visual indicator on the work item instead)
3. Service Level expectation is a deadline within which an item must be completed. (false, it is a forecast within which 85% of items are expected to be completed.)
4. Item Age shows how long an item is or was moving through the workflow. (true)
5. A Release date is a target date for when an item must be completed. (false, it is the date the item reached the end of the

TERRIBLE TROLLS!



The Three Terrible Trolls

Oh no! We've disturbed three Cavetrolls from their slumbers! They're in a terrible mood. They disrupt anything that's in their path!

1 minute

Quickly, break the group into three, each engaging one of the three trolls:

1. The Codebase Cruncher
2. The Workflow Waster
3. The Product Puncher

3 minutes

Within the breakout, the groups brainstorm ways their troll can do damage. They may use sticky notes in the color of their troll and stick these around their troll.

1. The Codebase Cruncher does damage to codebases.
2. The Workflow Waster disrupts the entire workflow.
3. The Product Puncher knocks out all sorts of value from the Product.

6 minutes

Let's brainstorm quick ways to eliminate your troll and prevent or counter its damage. Use sticky notes in a different color than their troll.

5 minutes

Review the trolls, how they caused damage, and how the teams countered them.

Definition of Workflow (DoW)

There is always a workflow, whether defined, understood, or agreed upon. Scrum Team members must have a shared understanding of the workflow to improve it. At the very least, Scrum requires a Definition of Done.



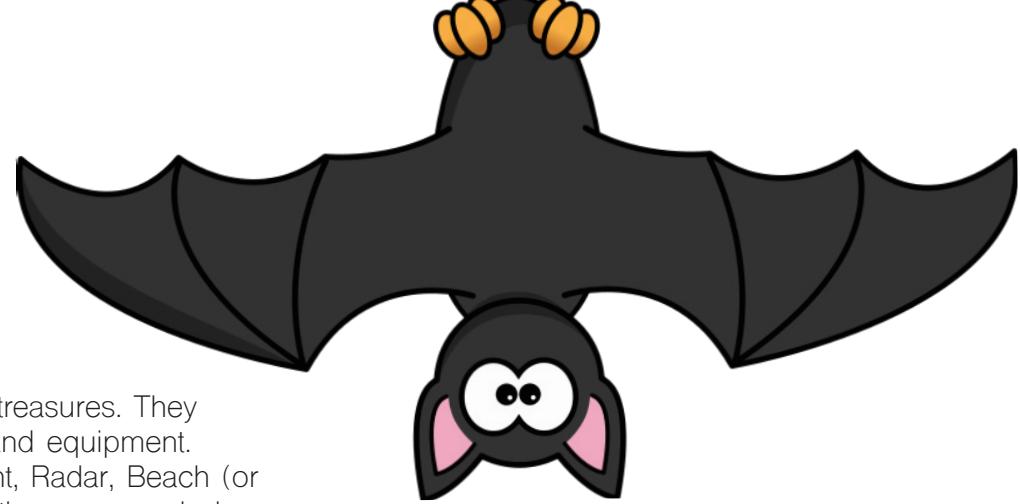
3 minutes

1 minute
per pair

Pair up. Each pair is assigned a different state of the workflow: Planning, Planning Ready, Developing, Releasable, Releasing, Learning. Each pair defines a short definition for that state. This is just to practice. The definitions may vary depending on context.

Each pair takes turns sharing their definition with the whole group.

Checkpoint 3



Our canyoneers are true professionals in marking their way toward valuable treasures. They know the importance of making work visible and have the right know-how and equipment. Other than visualizing the workflow, they may try adding a Compass, Spotlight, Radar, Beach (or Bat) signals, Tiny Monsters, "Postcards, Tacos, Unexpected Encounters, or other ways and plays to their Sprint Backlog.

What other techniques have proven helpful?

5 minutes

1. What are your team's current strategies for making work visible?
2. Can you show an example to the rest of the group that you like?

The canyoneers are close to discovering the final artifact: The Increment. They've come all this way and survived (so far). The best is yet to come. They better not be getting claustrophobic.



AN INCREMENT IS A CONCRETE STEPPING STONE TOWARD THE

Sprint Goal

EACH INCREMENT IS ADDITIVE TO ALL

features

AND THOROUGHLY

checked
for
mistakes

IN ORDER TO PROVIDE VALUE, THE INCREMENT MUST BE

approved by
the Product
Owner

WORK CANNOT BE CONSIDERED PART OF AN INCREMENT UNLESS IT MEETS THE

acceptance
criteria

INCREMENT MAY BE CREATED WITHIN A SPRINT.

only
one

AN INCREMENT

may
not

BE DELIVERED PRIOR TO THE END OF THE SPRINT.



Correct the Blanks

Oh, dear! it appears someone has been here before but not made it through alive. It appears all these options to fill in the blanks were traps. These tomb architects made it hard for us to reach the most precious of artifacts: the Increment.

5 minutes

Tick, tock. Without consulting the Scrum Guide, can the ‘tomb raiders’ rewrite to correct the statements in under 5 minutes?

The bat knows the correct answer reads:

- An increment is a concrete stepping stone toward the *Product Goal*.
- Each increment is additive to all *prior increments* and thoroughly *verified*.
- To provide value, the increment must be *usable*.
- Work cannot be considered part of an increment unless it meets the *Definition of Done*.
- *More than one* increment (or multiple increments) may be created within a Sprint.
- An increment *may* be delivered before the end of the Sprint





Architectural update



built-in
testscript



a Product
Backlog item
that meets the
Definition of
Done



A script that
collects
product
telemetry

Which of these qualify as an increment of value?



an
interactive
wireframe



Bugfix



An update
to the visual
design
document



a new
technical
design
version

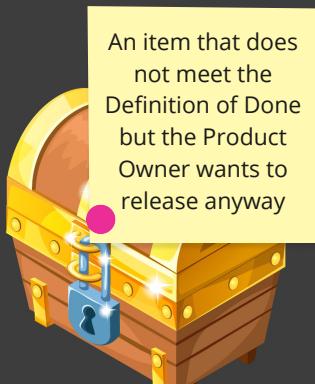
Place a dot!



A PowerPoint
presentation to
stakeholders with
screenshots of
what the Scrum
Team worked on.



a
security
patch



An item that does
not meet the
Definition of Done
but the Product
Owner wants to
release anyway



Workflow
improvement



there is no more work left to be done for us to be able to learn from it

we know what we need to learn from it



The Holy Grail

“Done”. This is what Scrum revolves around. It is a bar all Scrum Teams must pass each Sprint. But are we ever quite “done”? We make things better over time. We continuously improve. Product development is complex; we hardly ever get it right the first time.

“Great things are not done by impulse, but by a series of small things brought together.” — Vincent Van Gogh

To me “done” shouldn’t equal “final”, but “ready to learn from it”.

Adjusting the “ Definition of Done” can significantly impact the product. The “Done” state refers to the entire product: the sum of all increments. The whole product may need to be reworked to reflect this new state of Done.

15 minutes

1. Pass a (virtual) cup and invite each tomb raider to write down one or various statements that they believe defines “Done”.
2. They may then find a partner who is also ready. Together they may consolidate their answers to a single Definition of Done in a single cup.
3. They may then find another pair and once again consolidate their cups.
4. Finally, the whole group may consolidate their definition of “Done” in a single cup: the Holy Grail.

The bat knows that for something to qualify as “Done” in Scrum, the work done toward a Product Backlog item:

- Is potentially useful;
- Is in a useable condition;
- Is potentially releasable;
- Is inspectable;
- Is thoroughly tested;
- Supports empiricism at the end of the Sprint;
- Is not in conflict with product or organizational standards;

The bat may share these with the tomb raiders at the end of the play. Nothing leaves the kitchen that the cooks themselves are not satisfied with. Serve small plates frequently at exceptional quality.



An illustration of three climbers in a dark, rocky cave. One climber in orange gear stands on the left, reaching up towards a red vertical line. Two other climbers, one in a yellow shirt and another in a blue shirt, stand on a ledge further up the wall, looking at the line. The background consists of large, textured rock formations.

*What are **good approaches** to when
Developers **cannot yet achieve**
a "Done" increment within a Sprint?*

Almost Done...

Almost done is not done. There is no such thing as x% done. It's done, or it isn't. If your Scrum Team can create and deliver at least one valuable "done" increment within a Sprint without compromising quality or introducing technical debt, consider yourself successful at Scrum. When your team can make it out safely with those treasures, it's cause for celebration.

Scrum artifacts must be in a transparent state.
They must frequently be inspected and adapted.



Travel Journal

Exploring Artifacts

What did you discover when exploring these Scrum Artifacts together?

What scared/scares you?

What will change in practice through this training?



Appendix: 4C Map

C1 Connections - C2 Concepts - C3 Concrete Practice - C4 Conclusions

The R2M-VE applies a 4C baseline from Training From the Back of the Room, Virtual Edition (TBR-VE) by Sharon Bowman.

1. Exploring Artifacts

Learning Outcomes:

1. Draw/visualize Scrum Artifacts;
2. Map essential definitions related to Scrum Artifacts;
3. Share an understanding on the definition and purpose of Scrum Artifacts;
4. Defining the commitment for each Scrum Artifact;
5. List way to make work visible;
6. Share and list good Product Backlog Management practices ;
7. Share and list good Sprint Backlog Management practices.

Plays:

1. Tao Mediation (C1)
2. Bat Signals (C1)
3. Solve the Riddle (C1)
4. Pass the Guards (C2)
5. Fill in the blanks (C2)
6. 5x5 (C3)
7. Brainstorm (C3)
8. Checkpoint 1 (C4)



2. Product Backlog

Learning Outcomes:

1. Correct wrong and outdated statement about the Product Backlog;
2. Categorize what could/should be part of a Product Backlog;
3. Analyzing and identifying techniques used in a Product Backlog template/example;
4. Creating a spotlight for your Product Backlog to keep it clean and focussed towards a Product Goal;
5. Identifying Product Backlog management variables that impact predictability;
6. Applying Probabilistic Forecasting as a technique for empirical product planning;
7. List stakeholder management techniques to keep your stakeholders from being in the dark.

Plays:

1. Ancient Artifact (C1/C2)
2. In or Out (C2)
3. Product Backlog Template (C3)
4. Spotlight (C3)
5. Probabilistic Forecasting (C2)
6. Stakeholder Management (C3)
7. Stretch Break (C4)
8. Checkpoint 2 (C4)

3. Sprint Backlog

Learning Outcomes:

1. Correct wrong and outdated statement about the Sprint Backlog;
2. Sharing an understanding on the definition and purpose of the Sprint Backlog;
3. Categorize what could/should be part of a Sprint Backlog;
4. Applying Kanban as a strategy for visualizing workflow;
5. Co-creating a Kanban Board;
6. Defeating terrible trolls that mess up the product, workflow and codebase;
7. Co-creating an example of a Definition of Workflow;
8. Exchanging examples of workflow visualization techniques applied by participants in their practice.

Plays:

1. Ancient Artifact (C1/C2)
2. Unlock the Shrine (C2)
3. In or Out (C2)
4. Visualizing Workflow (C3)
5. Terrible Trolls (C3)
6. Definition of Workflow (C2/C3)
7. Checkpoint 3 (C4)

4. Increment

Learning Outcomes:

1. Correct wrong and outdated statement about the Increment;
2. Determining what constitutes an increment of value and what doesn't;
3. Co-creating a 'holy grail' Definition of Done;
4. Strategizing good approaches for when developers cannot yet achieve a "Done" increment within a Sprint;
5. Capturing learnings in the Travel Journal.

Plays:

6. Correct the Blanks (C2)
7. Treasure Chests (C2)
8. Holy Grail (C3)
9. Almost Done (C4)
10. Travel Journal (C4)



Call to Adventure

Create a Spotlight

Be "the Bat" in
the back of the Room

Forecast Predictively

Defeat the Terrible Trolls

In... or Out?!

Clean those Backlogs!

Find the Holy Grail

Kanbanize your Sprint Backlog



The Road to Mastery (R2M) is a Scrum Learning Journey containing 12 adventures (modules) from Basecamp to Bootcamp.

1. Basecamp
2. Agile Backpacking
3. The Game of Scrum
4. Living the Scrum Values
5. Mountaineering Scrum together
6. Kayaking the Value Stream
7. Surviving Self-Management
8. Deep Diving Developers
9. Exploring Artifacts
10. Smooth Sailing the Events
11. Bootcamp
12. Coaching Dojo

Acknowledgement

The Road to Mastery (R2M) is developed by Sjoerd Nijland.
The R2M is published at Serious Scrum; AGNC.

It builds on and is inspired by on the works of:

- Ken Schwaber and Jeff Sutherland: the Scrum Guide;
- Sharon Bowman: Training from the Back of the Room (TBR);
- Evelien Roos: Training from the Back of the Room Virtual Edition (TBR-VE);
- Project Zero: a research center at the Harvard Graduate School of Education;
- Henri Lipmanowicz and Keith McCandless: Liberating Structures;
- Scrum Patterns: Jeff Sutherland, James Coplien and others;
- Freepik for vector illustrations;
- And others attributed in the various plays.

For feedback and questions, please contact:
sjoerd.nyland@gmail.com

Comment on attribution:

"I aim to diligently attribute anyone who deserves credit or reference and refer to those sources of inspiration from which the Road to Mastery emerged. Please reach out with suggestions and comments on where this can and should be improved". - Sjoerd Nijland.



Serious Scrum R2M

Exploring Artifacts

SJOERD NIJLAND

[WWW.ROAD2MASTERY.COM](http://www.road2mastery.com)