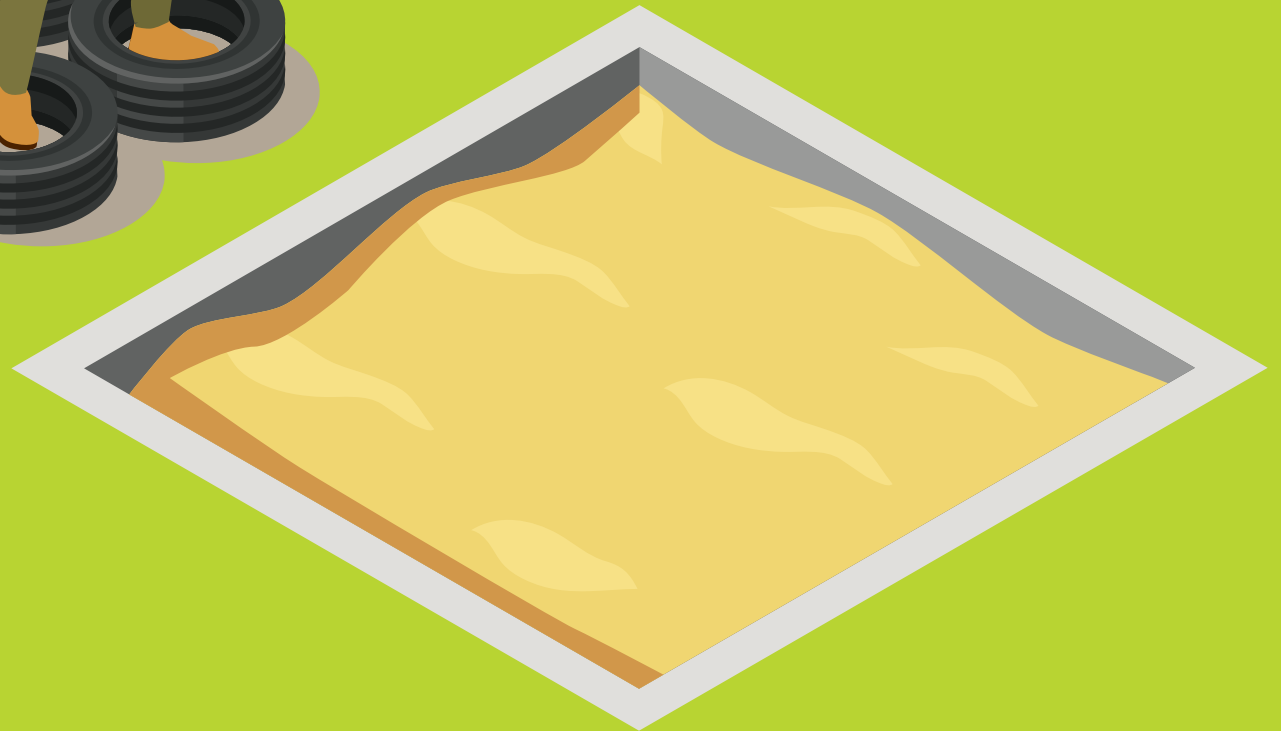


# Bootcamp!



**ROAD TO MASTERY**  
**VIRTUAL EDITION**  
**FOR GUIDES**





# Bootcamp!

**DRAG EACH OTHER THROUGH DIFFICULT  
CHALLENGES AND MESSY SITUATIONS**

# Bootcamp!

Scrum is like dragging an organization through its own dirt. The field is rarely in perfect condition, requiring Scrum Masters to coach circumstances where Scrum is poorly understood or practiced.

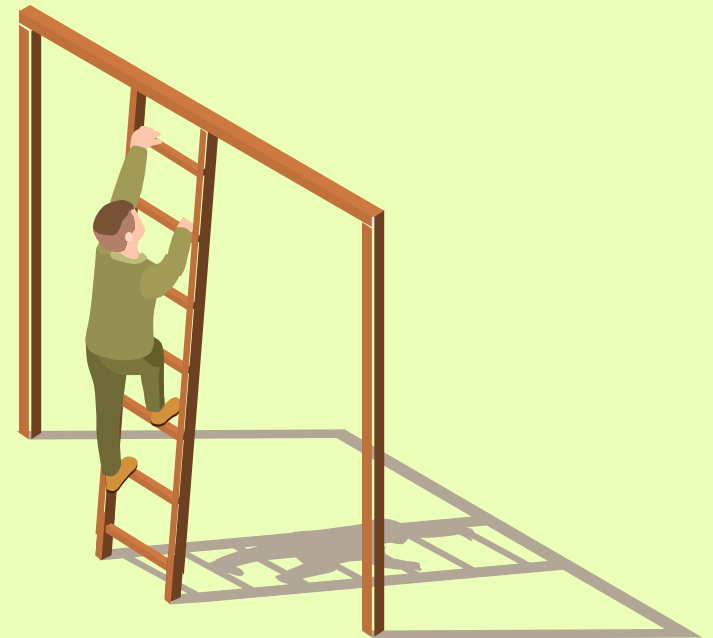
The Bootcamp is where you put yourself to the test. Here we connect your learning to behavior.

In this adventure, your travelgroup will work through challenging Scrum scenarios. You will need the courage to work through them. And making it to the end of Bootcamp requires determination and commitment.

I will refer to participants of this Bootcamp as “Rookies”, and by the end, they will be “Veterans”.

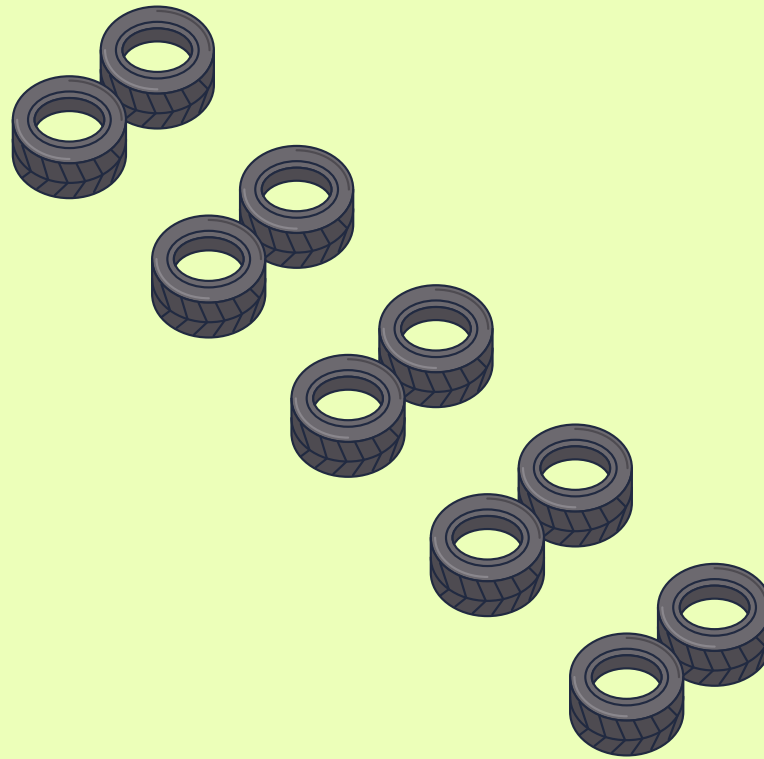
During Bootcamp, we will banish uncertainty, affirm strength, hold resolve, and drag each other through. Those will be our rookies’ best assets.

Make sure to introduce mind/body breaks between the challenges.



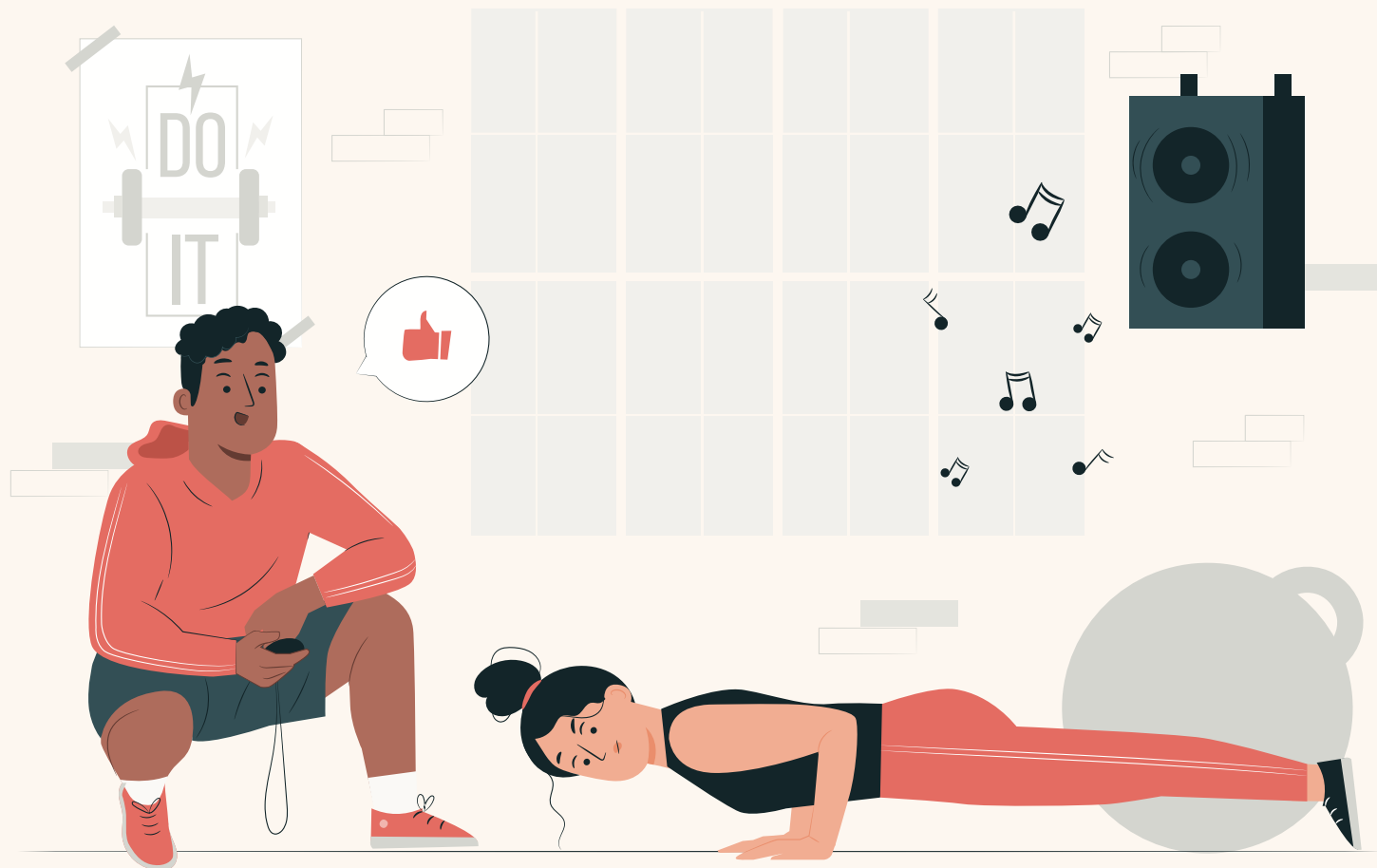
# Pair Up!

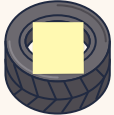

Our Rookies will pair up for the duration of the Bootcamp. They will autonomously go through various assignments. After each assignment, they check in with the trainer to review the assignment and be introduced to the next one.




# Warming Up


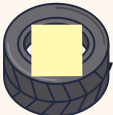

Invite each pair to fill in the blanks to complete the worksheet. Once the worksheet is completed, they may consult the Scrum Guide to verify and self-correct their worksheet. Once a pair has completed the assignment, they can report to the trainer to review it and receive the next.



The Scrum Team commits to achieving its  and to  each other.

Their primary  is on the work of the Sprint  
to make the best possible progress toward these goals.

The Scrum Team and its stakeholders are  about the work and the 

Scrum Team members  each other to be   people,

and are respected as such by the people with whom they work.

The Scrum Team members have the  to do  , to work on   
the empirical Scrum pillars of transparency, inspection, and adaptation come to life building .



## **TWO MINUTE CHALLENGE!**

*With your partner,*

*List as many characteristics of the Scrum Team you can!*





# Stick 'n High!

Once the two minutes are up, invite each pair to walk up to a wall, door, or large window. Each pair takes turns writing down one sticky note (per person) with one characteristic not already on the wall. Invite them to stick the note as high as they can.

As a trainer, you may keep the following list and cross out any of that closely resembles the answers provided by the rookies:

- Cross-functional
- Self-managing
- Cohesive
- Small (Typically ten or fewer)
- No sub-teams or hierarchies
- Focussed
- Committed
- Courageous
- Open
- Respected/Respectful
- Accountable for creating value

Continue for a maximum of three rounds per pair.

Review the list with the group. How many did you manage to cross out?



# Must, May, Could, Should

The following assignment tests how well your rookies know the difference between what rules must, may, could, or should be applied.


It's good to remind the participants that the rules in Scrum should guide the interactions between individuals, not stop them from thinking about what they believe to be the right thing to do, given the nature of the circumstance.




The Scrum Guide guides us on what rules must be followed for Scrum's pillars to be upheld. Empiricism and Trust are fundamental.

As a trainer, you can provide the following cheat sheet to rookies:


- **Should**, **May**, and **Must** imply extrinsic empowerment;
- **Could** implies intrinsic empowerment (ability);
- Just because you *could*, doesn't mean you *should*; It depends.
- Just because you *may*, doesn't mean you *can* or *must*; It depends.
- Just because you *should*, doesn't mean you *can* or *would*; It depends.
- Just because you *must*, doesn't mean you *can* or *will*. It depends.


Invite each pair to fill in the blanks to complete the worksheets. Once the worksheets are completed, they may consult the Scrum Guide to verify and self-correct their worksheet. Once a pair has completed the assignment, they can report to the trainer to review it and receive the next.

The Scrum artifacts and the progress toward agreed goals  be inspected frequently and diligently to detect potentially undesirable variances or problems.


When a Sprint's horizon is too long the Sprint Goal  become invalid, complexity  rise, and risk  increase.


If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced  be adjusted.

A product  be a service, a physical product, or something more abstract.


If Scrum Teams become too large, they  consider reorganizing into multiple cohesive Scrum Teams

The Product Goal is the long-term objective for the Scrum Team. They  fulfill (or abandon) one objective before taking on the next.

In order to provide value, the Increment  be usable.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams  follow it as a minimum

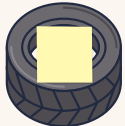
The Product Owner  delegate the responsibility to others

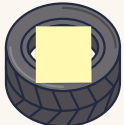
The Product Owner  represent the needs of many stakeholders in the Product Backlog.


The Sprint Backlog is updated throughout the Sprint as more is learned. It  have enough detail that they can inspect their progress in the Daily Scrum.




Each Sprint  be considered a short project.

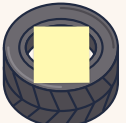
The Scrum Team  also invite other people to attend Sprint Planning to provide advice.

During the Sprint Review, the Product Backlog  also be adjusted to meet new opportunities.

During the Sprint Retrospective, the most impactful improvements are addressed as soon as possible. They  even be added to the **Sprint** Backlog for the next Sprint.

The Developers who will be doing the work are responsible for the sizing. The Product Owner  influence the Developers by helping them understand and select trade-offs.

Multiple Increments  be created within a Sprint. An Increment  be delivered to stakeholders prior to the end of the Sprint.

For Product Owners to succeed, the entire organization  respect their decisions.

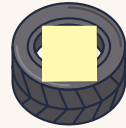


The Sprint Review is a working session and the Scrum Team



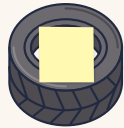
avoid limiting it to a presentation.

The emergent process and work



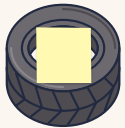
be visible to those performing the work as well as those receiving the work.

The Sprint Goal



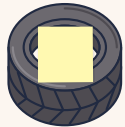
be finalized prior to the end of Sprint Planning.

Scope



be clarified and renegotiated with the Product Owner as more is learned.

A Sprint



be cancelled if the Sprint Goal becomes obsolete.

During the Sprint Planning, the Product Owner proposes how the product



increase its value and utility in the current Sprint





## **TWO MINUTE CHALLENGE!**

*The rules of Scrum guide relationships and interactions*

*With your partner,*

*List as many rules of the Sprint you can.*

*During the Sprint...*



# Scatter and Chatter

Once the two minutes are up, invite each pair to write down the rules on sticky notes and distribute them throughout the room. They can scatter them on the room's tables, walls, windows, doors, floors, ceilings, plants, and other attributes.

Once all pairs have scattered their rules, invite them to locate and group duplicate and similar rules together. They may communicate to coordinate the grouping.

Scrum is designed to be lightweight in its prescriptiveness. Invite the group to remove sticky notes with rules they believe they can do without to still be effective in Scrum. Each individual may remove as many as they like without agreement from others.

As a trainer, you may hunt for rules that aren't prescribed in the Scrum Guide.




# Mapping Competencies

The Scrum Master is there to enable the team self-manage its work. The worst way they could do this is by making decisions or managing the work for them. That robs the team of its ability to self-manage.

The Scrum Master is accountable for coaching the team and those external to the team in environments in which Scrum isn't fully understood and adopted. The team will experience situations where Scrum's values, routine, interactions, and responsibilities are challenged.

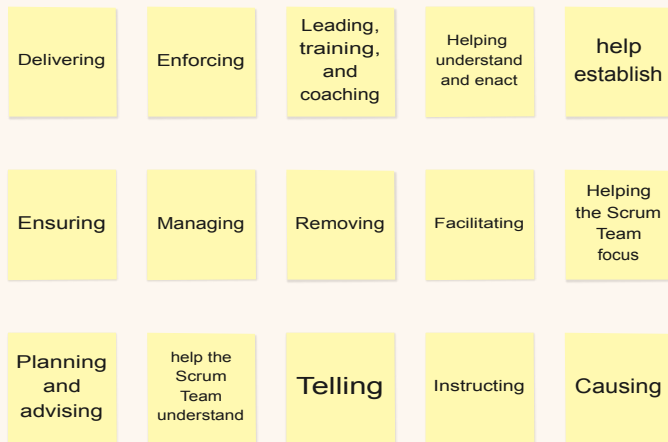
Let's see how well our rookies understand which competencies map to which service.

Invite the rookies to pair up with their partners and complete the worksheet to the right.  
Can they map the sticky notes to the blanks? 

Once they complete the worksheet, they may verify their answers with the Scrum Guide. They may self-correct their worksheet.

Once a pair has completed their assignment, they can report to the trainer to review the assignment, write down takeaways, and receive the next.





## Services to the Product Owner

- techniques for effective Product Goal definition and Product Backlog management;
- the need for clear and concise Product Backlog items;
- empirical product planning for a complex environment; and,
- stakeholder collaboration as requested or needed.

## Services to the Scrum Team

- the team members in self-management and cross-functionality;
- on creating high-value Increments that meet the Definition of Done;
- the removal of impediments to the Scrum Team's progress; and,
- that all Scrum events take place and are positive, productive, and kept within the timebox.

## Services to the wider organization

- the organization in its Scrum adoption;
- Scrum implementations within the organization;
- an empirical approach for complex work; and,
- barriers between stakeholders and Scrum Teams.

# Heavy Lifting

Introduce a 15-minute minimum mind/body break before starting this challenge.

Now that our rookies have demonstrated their understanding of Scrum values, rules, and accountabilities, they can put it to use and become veterans. You'll provide them with challenges that can weigh heavy on them!

1. Provide each pair with a hand-out with each a different scenario (without the commentary).
2. Ask them to break it down and formulate a response in five minutes.
3. Make your round to all pairs. They can share their response with you (without the other pairs listening in).
4. Once a pair shares their response with you, give them another hand-out with the commentary and ask them to review it.
5. Once you have made your round with all pairs, the pairs may exchange their scenario (without the commentary) with another pair.
6. Once more, each pair breaks it down and formulates a response in five minutes.
7. Once the five minutes are up, invite each pair to share their response with the pair that handed them the scenario. Exchange the commentary.

*Let your mind do  
the heavy work!*



What do other developers need?

Does "self-managing" apply to a **single** developer or the **whole** team?

A Developer, Bart, argues he is **self-managing** and indicates **he** no longer wants to participate in the Daily Scrum; he **collaborates** with other Developers **throughout the day**.  
He feels it has become a **needless ritual**.

What's **enabling** Bart to do this?

What about this **event** makes it feel like a needless "ritual" to Bart?  
How would Bart define the **purpose** of the Daily Scrum?

The Daily Scrum is not the only time Developers can adjust their plan. They often meet throughout the day for more detailed discussions, **inspecting** and **adapting** the the Sprint Backlog. Collaboration is not the only purpose of the Daily Scrum.

The Daily Scrum should be a **valuable** and **meaningful** event that produces an **actionable, visible** plan for the day towards the Sprint Goal.



It's good that Meera asks and not assigns.  
But is it a decision for a **single** Developer to make?

If Developers start new work,  
what are they **not** finishing?

The Product Owner, Meera, asks a Developer, Raj, to start work on a new Product Backlog item late in the Sprint. It is **not** related to the Sprint Goal. Raj says he can start work on this. Raj thinks they are close to meeting the Sprint Goal, but may drop an improvement goal. Raj asks you what to do.

Raj thinks the Sprint Goal is close to being met, but it is **not yet** met. Also, what does it mean for Raj to be committed to the goals of the team? What should Raj be **focussing** on foremost?

We welcome changes, even late in development. Yet, this request does not contribute to the Sprint Goal. Is this **distracting** Developers from achieving their Sprint Goal?

Given these considerations, Raj might **consult** other developers to agree what is the right thing to do.



The Definition of Done exists to create **transparency**, providing everyone a **shared understanding** of completed work as part of the Increment. As it stands, is this state transparent?

The Developers disagree on whether a specific Product Backlog item meets the **Definition of Done**. Specifically, Prabhat and Jess are concerned with **how Oskar** wrote a part of the code.

Oskar agrees they should **refactor**, but that it is already **releasable** and **usable**.

Now What?



The Developers are owners of the codebase. Irrespective of who was responsible for writing it, they are all accountable for it.

A Scrum Master may encourage Developers to refactor the code **together**, so that everyone can learn from it and be confident with how the Product Backlog item meets the Definition of Done.

Whether it is releasable and usable is something they should **collectively** understand and support.

The purpose of the Sprint Review is to inspect the **outcome** of the Sprint and determine future adaptations. What has changed? What to do next?

Empiricism asserts that knowledge comes from experience and making decisions based on what is **observed**. The Scrum Team learns about the needs incrementally **through** delivery.

During a **Sprint Review**, the Stakeholders share **new needs** with the Scrum Team. A Developer, Rio, suggests that **re-work** is wasteful and that they should have shared those needs during Product Backlog refinement sessions.

Not re-working the product (as more is learned) is also wasteful. There will always be guesswork and therefore re-work. Consider the waste of developing something that didn't turn out to be needed.

During ongoing refinement, the Scrum Team may look into crystal balls, but during a Sprint Review they inspect the **real** Increment - which is the **primary** measure of progress!

The **actual** needs emerge through actual **use**. Therefore, it is a good idea to share **emerging** needs directly while **inspecting** the latest increment and **reviewing** the outcomes of the Sprint together.



People organize their schedules based on what they value. How valuable are the **outcomes** of the Sprint to Roger if he is not there to review them?

Kate is part of the Scrum Team and is a **required** participant in the Sprint Retrospective. She should be reminded of her **accountability** as a professional team member.

Roger is a key Stakeholder to the Scrum Team. Roger and Kate, the Product Owner, meet frequently. Roger never attends the Sprint Review. He explains he has a busy **schedule** and that his sessions with Kate are enough for him to remain informed about the ongoing development **status**. During the Sprint Retrospective, the Developers express that Kate frequently interrupts their focus by asking them to do ad-hoc changes resulting from her **meetings** with Roger. Kate is **not present** because she is busy making a **Sprint Report** for Roger.

Kate is acting as a proxy between Stakeholders and Developers. This is an **arduous** way of ensuring their needs are transparent, visible and understood by all. Furthermore, it disrupts focus.

Roger may not know that the Sprint Review accomplishes far more than providing status. Has Roger considered that by participating in the Sprint Review would render his frequent meetings with Kate and reports redundant, thus freeing up their busy schedules?



Where is the line between strict and too strict? The Definition of Done is a standard that must be followed as a **minimum** as defined by the wider (development) organization. It defines what is **essential** to a **compliant** and **quality** product.

Fortunately, the Developers are **open** about the challenge. With **courage**, they can work through it. The gap between current practices and target practices is a measure of competence and competitive risk. This starts with knowing what **is** possible and what can be **made** possible.

Developers agree that creating a working Increment within a Sprint is **impossible**. Fatima explains that the Definition of Done set by the development organization is **too strict**. She says it imposes several **limitations** on their workflow. Fatima also shares they are **inexperienced** in certain practices.



What practices the Scrum Team applies should be up to them. **Imposing** practices by masquerading them as a Definition of Done may impede the team's ability to self-manage. In that case, it may hinder them. But that might not be the case here. Those limitations may be **quality standards**. Developers may require more experience to pass the bar. How can this experience best be gained?

The Definition of Done (DoD) is a formal description of the **state** of the Increment, not an imposed workflow. The Scrum Team **internally** decides when, what and how to get there. It may be that a specific workflow is required to achieve that the "Done" state of the Increment. What, other than their experience, is specifically limiting Developers?



Developers are accountable for creating a plan for the Sprint, instilling quality by adhering to a Definition of Done; adapting their plan each day toward the Sprint Goal, and holding each other accountable as professionals.

A great way to review the effectiveness of a Scrum Team is to **review** the outcome of the Sprint together with stakeholders regularly. Scrum provides just such an opportunity: The Sprint Review. Scrum provides **Sprint Retrospectives**, where the entire Scrum Team inspects how individuals perform and interact.

The HR manager, Marion, instructs Henry, the Scrum Master, to **review** the **performance** of developers. She believes Henry is their **manager**. She also assumes him to be **closest** to their day-to-day work.  
What could Henry do?

There may be aspects to a Developer's work and capabilities that may not be suitable for a Scrum Master to comment on. Who, besides Henry, are close to the day-to-day work?

The Scrum Master is **not hierarchical** to developers as there may not be a hierarchy within a Scrum Team. The Scrum Master can comment on how individual developers live up to their accountability and can hold them accountable as professionals.



This does not necessarily indicate a problem. Product Backlog items usually acquire **transparency** through refining and sizing activities. It is the process of coming to a **shared understanding** of the nature of the challenge.

Brian may influence developers, but will ultimately need to **respect** their ability to size the work. They have the skills and are closest to it.

The Product Owner, Brian, **doesn't understand** the size given by the Developers for a Product Backlog item. Brian is **surprised** to learn that the item is deemed **very complex**. Brian refers to a **similar** Product Backlog item, which was **not complex** at all.

Sharing similarities and differences can benefit transparency. The nature of complexity is that it contains **unknowable** elements. Product Development can be a real jungle. Developers can share the nature of the complexities with the Product Owner.

It's great that Brian can refer to earlier work. What has happened in the past can provide useful information for new developments. This way, the Developers can also learn more about Brian's expectations.



It is the Scrum Master's **accountability** to ensure that all Scrum events **take place** and are positive, **productive**, and kept within the timebox, not that of the Product Owner.



Sprint Retrospectives are **optimally** held at the same time and place to reduce complexity. Sometimes, there are good reasons to prepone or postpone it.

The CEO announces a company event. The event is **scheduled** to occur at the same time as the Sprint Retrospective.

The Product Owner, Roy, **instructs** Katherine, the Scrum Master, to **cancel** rather than pre- or postpone. Roy argues: "We have so many anyway. I do not want to **hamper** our productivity even more".

Canceling a Sprint Retrospective eliminates an opportunity to plan ways to increase quality and effectiveness. It would result in **a lost opportunity** to inspect and adapt. Scrum events are designed to provoke change so the team can quickly adapt to newly discovered opportunities.

The Sprint Retrospective exists to plan ways to increase quality and effectiveness, not to hamper it. What if the Scrum Team does not improve its effectiveness? What is **really** hampering productivity?

Understandably, the Product Owner is concerned that the company event reduces the Scrum Team's capacity to deliver value. But that is a company wide decision. This should not be deflected to the Sprint Retrospective. The Scrum Team can be transparent on the impact of the company event to Stakeholders.

It is likely that Pierre learned something new about the item. Other Developers may also know something about the item Pierre does not. Who should Pierre talk to?

Pierre isn't individually responsible for that item. Pierre can be encouraged to be **open** about the work and the challenges with his team.


During a Sprint, a Developer, Pierre, believes an item in the forecast is **much more complex** in size than estimated. Pierre **independently** swaps it for an item he thinks he can **deliver** that Sprint.

Pierre discovers more work is required for that item. It does not lower its value. It might still be the most important thing to work through. Is this item needed to achieve the **Sprint Goal**? Development in Scrum requires the **courage** to work through tough problems as they emerge.



Is there a better way for Tina to see a highly **visible, real-time** picture of the progress toward the Sprint Goal? By inspecting the Sprint Backlog, Tina should see the progress. Tina can review Increments which are a **primary** measure of progress. What else is Tina missing here?

The **purpose** of the Daily Scrum is for Developers to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work. It is **not** to provide a status update to the Product Owner.

An illustration of a man (Fred) in a black tank top and blue shorts, performing a squat with a barbell. A yellow line connects him to the central text box.

The Product Owner, Tina, **insists** on participating in the **Daily Scrum**. She tells the Scrum Master, Fred, that she wants to be informed about the **status** of development. She also tells **Fred** that the Developers are not **mature** enough to plan the work for the day themselves.

We **respect** each other for being **capable** individuals. We are all adults and mature in our work. By parenting the Daily Scrum and insisting on a status update, Tina may be impeding the Developer's ability to **self-manage** within that 15-minute timebox.

Fred is ultimately **accountable** that the Daily Scrum is positive and productive for Developers, not Tina. Fred can demonstrate to Tina that her participation, although well intended, is not helpful. Fred can show her **better ways** of staying informed.

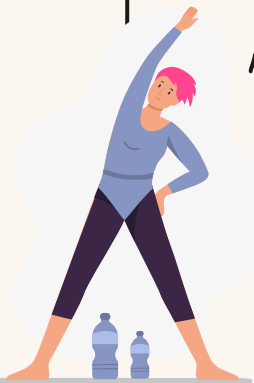
Inspection without adaptation is considered pointless. The defect renders the Product in an **'un'done'** state. The adjustment must be made as soon as possible to minimize further deviation.

The Developers may not be responsible for its creation, but they are now **accountable** for instilling quality by adhering to a Definition of Done. Whatever the code's origins, the Developers are collective owners of it. It's not just Mary's call.

A Developer, Ali, **inspects** a defect in the Product that is affecting some users. The defect originates in a part of a **legacy codebase** that the Developers poorly understand. It is a source of frustration for users and developers alike. Other Developers are **reluctant** to work on it, and even the Product Owner, Mary, tells Ali not to **waste** time on it. This **upsets** Ali.

**Technical Debt** is typically unpredictable overhead often caused by less-than-ideal decisions, contributing to the total cost of ownership. It's a source of waste. It badly affects customer experience, and it dilutes value.

It takes **courage** for Ali to be **open** about the challenge. He clearly wants to work through this thorny problem. Ali cares. He does the right thing by holding his peers equally accountable.



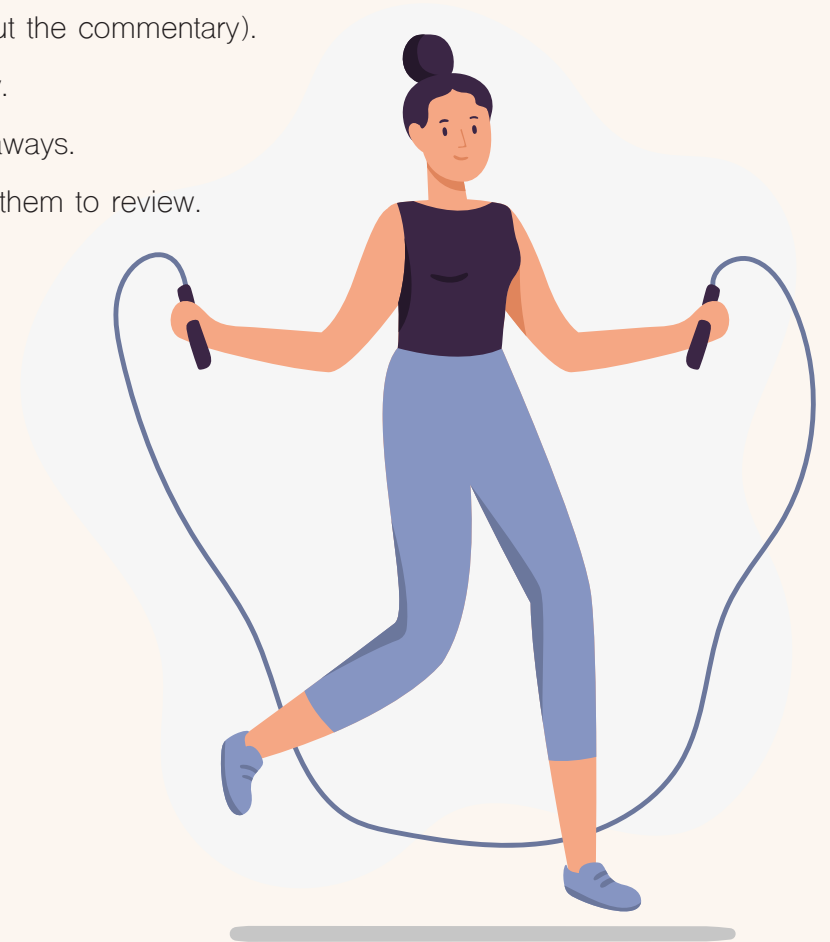


# Jump!

Introduce a 15-minute minimum mind/body break before starting this challenge.

What if something gets in the way? Can our veterans jump through these scenarios?

1. Provide each individual with a hand-out with each the same scenario (without the commentary).
2. Invite each individual to write down their answer in under five minutes silently.
3. Invite individuals to pair up and compare their answers and write down takeaways.
4. Provide each pair with a second hand-out with the trainer's commentary for them to review.
5. Repeat these steps with the second and third scenarios.





# What if...



*You notice a Developer, Lucas, remained silent throughout the entire Sprint Planning and did not collaborate.*

*Lucas may be attending the Sprint Planning but is not participating. The possible reasons for this are not provided in this context.*

*Is Lucas familiar with the purpose of the Sprint Planning and what is expected from participants? There may be a lack of psychological safety. Is it just Lucas? It is tempting to make assumptions, but it is better to share your **observations** with Lucas than your interpretations.*

*How is the Sprint Planning facilitated? Does it provide an opportunity for everyone to participate? Various facilitation techniques (or plays) can invite active collaboration between all participants. Consider a 1-2-4-all, for example.*

*Scrum Team members can be encouraged to be open about the work and the challenges. They may be encouraged to do the right thing. What does Lucas want and need?*

## What if...



*Just prior to the Sprint Review the Developers share with you that they are unable to present any working, usable increment during the Sprint Review.*

*This may be an opportunity for a Scrum Master to address why Scrum requires at least one "Done" Increment every Sprint. The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. A Product Backlog item must meet the Definition of Done to be released or even presented at the Sprint Review. The Scrum Team must be open about its challenges.*

*The Scrum Team has many opportunities to visualize progress towards their goal and identify impediments during the Sprint. What is getting in the way of the Scrum Team to deliver not even one "done" increment? Is the team working as a cohesive and focused unit? What's the right thing to do? How can the team work together and demonstrate courage to work through challenging problems?*

# What if...



*A Developer tells you that "Scrum just doesn't work here"*

*Scrum foments change which is difficult and fraught with conflict. It's a journey that may take many years of sustained effort. The focus in Scrum practice is changing from old habits to better ways. Scrum itself is not an actor that works. It's practiced and played.*

*What exactly isn't working? Doesn't Scrum provide opportunities to inspect and adapt? Through practice, one can determine if its philosophy, theory, and structure help to achieve goals and create value.*

*The Scrum framework is also purposefully incomplete. It's not prescriptive in determining what works and what doesn't. Scrum relies on empowered self-managing teams, who are accountable for creating valuable, useful Increments every Sprint. To do so, they must have all the necessary skills and means. What is getting in the way of this? What would the Developer need to gain confidence?*

## What if...



*A director of the organisation, Maria, comes to you the Scrum Master, asking for advice on how to organize Scrum Teams for developing a new product. There are 15 developers available.*

Start with a small unit or two to develop the Product incrementally, focusing on one objective (Product Goal) at a time. This way, the teams are small enough to remain nimble but large enough with all the skills and focus needed to create value each Sprint.

Ideally, the formation is self-managed by the (Product) development organization. I suggest inviting the 15 available developers to self-manage the formation. To do so, the Product Owner communicates the Product Goal and the items on the Product Backlog needed to achieve it. They also must collectively formulate the Definition of Done for the Product. Ultimately each team requires all the necessary capabilities to meet the Definition of Done and deliver working increment(s) each Sprint.

## What if...



*Developers often collaborate with stakeholders throughout the day. Sometimes these collaborations are helping the Scrum Team achieve its goal. Other times they are distracting.*

The Scrum Team focuses on achieving a Sprint Goal. This can help developers decide what interactions with stakeholders are helpful towards achieving it and which are not. When developers are open about the work and the challenges, the Scrum Master learns what interactions with stakeholders are helpful and which are not. Scrum provides regular opportunities for Stakeholders to collaborate with the Scrum Team, such as the Sprint Review. During the Sprint, the Product Backlog may be refined as needed, and stakeholders may be invited. The Daily Scrum is an opportunity for Developers to align what interactions with Stakeholders will be helpful to them that day and which will not. The Sprint Retrospective also provides an opportunity for the Scrum Team to plan ways how developers ideally interact with Stakeholders during the Sprint while retaining their focus.

# What if...



*You are asked to explain what Technical Debt is to someone unfamiliar with the term. What risk does it introduce and how can Scrum help to address it?*

Technical Debt is the accumulation of quality deficiencies in a Product. These deficiencies are often obscure. Obscurity ultimately results in poor decision-making as there is insufficient understanding.

Technical Debt results in extra work that accumulates when code is not written or designed in an optimal way. This happens when developers don't apply recommended professional practices and make suboptimal decisions, such as shortcuts and workarounds. This can happen when the team is under pressure to deliver quickly, or when they don't have all the information they need to make the best decisions. It generally results in fragile, rigid, obscure, and immobile parts of systems or products. This may also be the result of insufficient refactoring, upkeep, and untimely adaptations to ever-changing technology and market conditions. It introduces risk because rigidity impedes the ability to timely adapt and deliver value. It generally results in unstable systems causing unexpected problems and disruptions, badly affecting the customer experience.

What may help to address this, is to encourage the Scrum Team to collectively assume accountability over the Technical Debt. The Scrum Team may adopt a pattern that addresses anything outside acceptable limits together. The Definition of Done should result in a shared understanding of the Product's state. Furthermore, they may use visualization techniques to make the impact of Technical Debt visible to the wider organization.

## What if...



*A Scrum Team needs to determine what Sprint Length is suitable for them. Which factors would they need to take into account?*

The Sprint Length should be short enough to enable timely inspection and adaptation of progress toward a Product Goal. When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can generate more learning cycles and limit the risk of cost and effort to a smaller time frame. I may learn from the Product Owner and stakeholders how volatile demand and market may be, and I may learn from Developers the nature of the complexities involved with turning ideas into value. The Definition of Done would be the starting point as it indicates how stringent the quality standards are. We may learn from past performance in both demand and development.

# What if...



*You are asked to explain what Technical Debt is to someone unfamiliar with the term. What risk does it introduce and how can Scrum help to address it?*

Technical Debt is the accumulation of quality deficiencies in a Product. These deficiencies are often obscure. Obscurity ultimately results in poor decision-making as there is insufficient understanding.

Technical Debt results in extra work that accumulates when code is not written or designed in an optimal way. This happens when developers don't apply recommended professional practices and make suboptimal decisions, such as shortcuts and workarounds. This can happen when the team is under pressure to deliver quickly, or when they don't have all the information they need to make the best decisions. It generally results in fragile, rigid, obscure, and immobile parts of systems or products. This may also be the result of insufficient refactoring, upkeep, and untimely adaptations to ever-changing technology and market conditions. It introduces risk because rigidity impedes the ability to timely adapt and deliver value. It generally results in unstable systems causing unexpected problems and disruptions, badly affecting the customer experience.

What may help to address this, is to encourage the Scrum Team to collectively assume accountability over the Technical Debt. The Scrum Team may adopt a pattern that addresses anything outside acceptable limits together. The Definition of Done should result in a shared understanding of the Product's state. Furthermore, they may use visualization techniques to make the impact of Technical Debt visible to the wider organization.



## What if...



*The Product Owner asks you if the Sprint Goal is immutable. Can it be changed during a Sprint?*

After the Sprint Goal is set as a commitment by the Scrum Team it should not be adapted. A Scrum Team is expected to adapt the moment it learns anything new to optimize their ability to meet the Sprint Goal. To do so, the Sprint Goal itself must be a clear, fixed objective. When a Scrum Team does not enjoy a shared understanding of that objective, they can be encouraged to revisit it, to increase everyone's understanding of it.

It may be the case that a Sprint Goal becomes obsolete. In such a situation, the Product Owner may decide to cancel the Sprint and start a new one. This would start with a Sprint Planning event where a new Sprint Goal can be formulated, and a new plan towards it can be drafted.

## What if...



*The Product Owner asks you why the Sprint Retrospective is held after the Sprint Review. The Product Owner suggests to switch the Sprint Review and Sprint Retrospective around as that would better suit the availability of a Stakeholder.*

The Sprint Retrospective is held after the Scrum Teams and Stakeholders have inspected the outcome of the Sprint and determined future adaptations. The Scrum Team can plan its improvements attuned to the learnings and determinations resulting from the Sprint Review. The most impactful improvements can therefore be addressed as soon as possible. They can identify what is needed to adapt anything concerning individuals and their interactions, processes, tools, and their Definition of Done. If the Sprint Retrospective were held before the Sprint Review, the opportunity will be lost to timely address the feedback, learnings, and determinations from the Sprint Review. Team improvements may become misaligned with determinations resulting from the collaboration with Stakeholders during the Sprint Review. I may follow up with the stakeholder in question to learn more about their availability.

## What if...



*Developers ask you to coach them in ways to make the progress towards the Sprint Goal visible.*

A Scrum Master is not serving a Scrum Team when they do something that the Scrum Team could do themselves. But they don't need to be a passive observer when asked for support. A Scrum Master may instruct the Developers in ways to maintain the Sprint Backlog as a highly visible, real-time picture of the work needed to achieve the Sprint Goal. Ultimately it is the Developers' discretion and it is their accountability.



## What if...

*A stakeholder reaches out to you to learn more about Scrum Values. The Scrum Guide states: “The Scrum Team and its stakeholders are open about the work and the challenges”.  
Is Openness the only relevant Value for stakeholders?*

All Scrum Values are relevant to Stakeholders. When people live the values, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust. This is what enables the Scrum Team to generate value. To make the best possible progress toward these goals, stakeholders need to understand that the Scrum Team must remain focused on its goals and they should not distract them. Furthermore, it is an ethical standard for Stakeholders to respect the Scrum Team members in being capable individuals. Without this respect, their ability to self-manage will be impeded. That would diminish the Scrum Team's ability to openly and creatively respond to challenges, and they will be less inclined to create value for them. The field may not be in perfect condition, but they can trust the Scrum Team to stay committed to supporting each other, doing the right thing while working through challenges.

# What if...



*Developers are familiar with the purpose of the Daily Scrum. They facilitate this event themselves. As a Scrum Master, you serve the Scrum Team ensuring that all Scrum events take place and are positive, productive, and kept within the timebox. How will you know this is the case for the Daily Scrum without attending it yourself?*

A Scrum Master is not banned from attending. They can still attend the Daily Scrum if and when deemed necessary.

The Scrum Master can inspect the outcome of the Daily Scrum. One way to do so is to inspect the Sprint Backlog to see if it is indeed a real-time highly-visible picture of the work of the Sprint and the progress towards the Sprint Goal. Impediments may be made visible. Would you know if Developers are communicating and making decisions timely? Are they reducing the need for other meetings?

The Sprint Retrospective provides an opportunity to ask how Developers are experiencing the Daily Scrum and how they are fulfilling its purpose (and what is challenging them).

# Jump to Conclusions...

Keep jumping! We're nearing completion of the Bootcamp, so it's time to jump to conclusions.

Invite the veterans to run through their notes and list their key takeaways.

Can they share their takeaways with the group using an actual jumping rope?

If jumping a rope (or simply jumping up and down) is not possible, invite participants to imagine their own physical movements while jumping to their conclusions.



# Cooling down!

Let's end the Bootcamp by cooling down.

Participants are free to share what was cool and refreshing about their Bootcamp while enjoying some refreshing drinks.

Give each other a pat on the back and share a few encouraging and appreciative words.

Completing Bootcamp validates the courage and commitment of the participants.



# Appendix: Training Map

## 1. Warming Up

### Learning Outcomes:

1. Establish a behavioral baseline for responding to challenging scenarios;
2. Self-correcting the applicability of Scrum Values;
3. Brainwriting the characteristics of a Scrum Team.

### Plays:

1. Fill in the Blanks
2. 2-minute Challenge
3. Stick 'm High

## 2. Rules of Scrum

### Learning Outcomes:

1. Understanding how the rules in Scrum guide interactions as a baseline for responding to challenging scenarios;
2. Self-correcting what is must, may, could or should in Scrum;
3. Brainwriting, teaching back and minimizing Sprint rules.

### Plays:

1. Fill in the Blanks
2. 2-minute challenge
3. Scatter and Chatter
4. Mapping the Blanks

## 3. Heavy Lifting

### Learning Outcomes:

1. Defining what to service to whom as a baseline for responding to complex scenarios:  
Mapping the blanks;
2. Assessing responses to challenging Scrum Scenarios.

### Plays:

1. Pair & Share Scenarios
2. Teachback
3. What ifs

## 4. Cooling Down

### Learning Outcomes:

1. Jumping to conclusions by sharing key take aways;
2. Share feedback on the Bootcamp;
3. Positively appraise the completion of the commitment.

### Plays:

1. Jump to Conclusions
2. Cooling Down



*R2M*



*BOOTCAMP!*

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
- Freepik for vector illustrations;
- And others attributed in the various plays.

For feedback and questions, please contact:  
[sjoerd.nyland@gmail.com](mailto: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*  
**Bootcamp!**

*STOERD NIJLAND*

*WWW.ROADZMASTERY.COM*

