# Plan-o-gram

|  |  |  |
| --- | --- | --- |
| 0-3 | Plan for people to be late |  |
| 3-5 | Introduction and ground rules |  |
| 5-15 | Determine the number of people and split the stories up. If only five people use top 10 in groups and a quick talk about the other 10 | Timer on screen |
| 15-20 | Go over Roles |  |
| 20-35 | Go over Getting things done |  |
| 35-45 | Virgil Talking |  |
| 45-55 | QA |  |
| Ending | Go over the commitments |  |

Welcome to the overview of how an Agile team functions to the best of its abilities. I like to start out by getting a sense of what you know already so that I can skim the parts you understand and where to go into more detail.

I always like to start with some ground rules and agreements

* We come from a place where we want to listen;
* We realize that we all come from a place of good;
* We allow ourselves to be open to the ideas presented;
* We will respect each other as it relates to differences in opinions, time commitments, and feelings;
* I will listen to what you have to say;
* I will be willing to be challenged and change my mind when it makes sense;
* I will do my best to explain points - while realizing I may be imperfect in that explanation;
* I will be respectful to everyone of you as I know you are coming from a good place.

One of the rules from every Global Scrum Gathering I’ve attended is this: The Rule of Two Feet. If you no longer find this useful to you, then please leave. I don’t want you to be here if you cannot be present in this space. To use something from T-Mobile – Be Here Now; barring an emergency, please keep your phones and laptops away. They become a distraction from the goals and may be distracting to those around you.

True / False / It Depends

1. The Scrum team decides how they will operate
2. The Scrum team is empowered to make decisions
3. The Scrum team should choose their work
4. The stakeholders should ask the developers how the work is going
5. The Scrum Master coaches the team on the process
6. The Product Owner sets the priority of the backlog
7. The Engineering Team needs full details of each PBI to estimate the size or complexity of the item.
8. Management can replace one PBI in the Sprint Backlog with another during the Sprint.
9. The ScrumMaster is responsible for protecting the Engineering Team from both internal and external distractions.
10. During a Sprint, anyone with sufficient hierarchical authority may require the Engineering Team to take on additional tasks.
11. The Scrum Master should be providing updates to the customer directly
12. The Developers do not prioritize the backlog
13. The Scrum Master runs the Scrum Ceremonies
14. The Sr. Manager prioritizes the backlog
15. Periodic testing Sprints are the preferred mechanism for ensuring high quality.
16. The Scrum Team can request an Abnormal Termination if the Sprint Plan is significantly invalidated or interrupted by external factors.
17. The ScrumMaster assigns tasks to Team members.
18. For most efficient flow, the Engineering Team should route all Product Owner questions through the Scrum Master.
19. Scrum allows individual team members to be specialists.
20. With the self-organizing team concept, there is no longer any need for managers.
21. It Depends – how the team functions within the structure is ok. For example, the Scrum Master has the requirement to make sure that they follow the process, the team can choose *how* to implement. For example, the SM builds the house; once the Team understands the layout of the house, they decide what color to paint and where the furniture goes.
22. True. The engineering team makes decisions as to who may do the work, determine what they can accomplish and the team agreements
23. True. Team members, not the SM or anyone else make the decision. The HR manager may influence by providing input to team members for the furtherance of their career goals or to stretch themselves. However, if they or the team needs them to focus on what they can do best, they make the decision.
24. It depends. There are times when that direct communication may be needed or desired. Team members should feel empowered to do that; however if the Scrum Master finds that it is a distraction to the team or the accomplishment of the sprint goal, they should step in. At no point should the stakeholder introduce a change from an agreed to PBI; if there were issues with how they wrote the story, they may close it, write a correct one and prioritize the work.
25. True.
26. True. The Product Owner prioritizes the backlog based on business priorities and other factors.
27. False. The team should have a ‘good enough’ understanding of the requirement including a description, purpose, and acceptance criteria so that they can accomplish the work. If the team notices implementation details are missing, the team should clarify those details before estimation.
28. False. The Product Owner owns the backlog, Management can influence the decision and if the PO agrees they can bring it to the team and determine if it’s possible. The Engineers make the final determination as the PO may not be aware of its implications.
29. True. The SM is responsible for protecting the team in this manner and holding the team to their commitment because they are minimizing these interruptions.
30. False. As part of the cultural shift, the hierarchical authority has no place in Scrum; they influence the stakeholders and Product Owner.
31. True. The Scrum Master using boards provides the team’s progress. When the Product Owner isn’t available, and there is an urgent need, the Scrum Master can give that information.
32. True. The Engineers provide feedback or influence the Product Owner’s Prioritization given technical hurdles or other development concerns which impact dependent stories.
33. False. The Scrum Master ensures that the ceremonies are scheduled and that they occur. They can facilitate the meetings and help keep them focused; as appropriate team members can run the meeting. **Note:**It is incumbent on the Scrum Master to run the meeting if the team isn’t stepping up or is a lack of maturity to accomplish the task. Examples would be if the team members are not allowing others to make their points at a retrospective, they start giving status updates during the stand-up, or are breaking the timeboxes for these ceremonies.
34. False, any manager – regardless of their role – goes through the Product Owner.
35. True
36. True. If the team cannot accomplish the majority of the work, they can request the termination and the PO, SM, and Engineering Team make the call to end the sprint. If terminated, a longer than usual retrospective ***must*** occur to determine how/why it occurred and how to reduce the chances of it from happening again. Additionally, stories must be corrected and reprioritized before beginning the planning of the next sprint.
37. False. Team members are empowered to make those decisions.
38. False. Scrum is about open communication and transparency. If there are questions, the team is empowered to ask them. Where possible, the SM could be part of those discussions. At a minimum, they should be aware when they occur as it may be a ‘smell’ that there is something wrong in the process or correction the team needs to meet their commitment on what makes a good story. The SM could then put it on the list of things to possibly discuss during the retro.
39. True. While we want there to be cross-functional teams, people may have natural specialties based upon their experience. However, they can strive to make sure that the knowledge passes on to the rest of the team, so we have more than a team of experts.
40. False. The concept of self-organization is limited to the Engineering Team deciding how best to realize the Sprint Goal. Self-organization does not extend to administrative or human resources matters.

Roles

Product Owner

* Voice of the customer
  + Has a clear understanding of customer requests
  + Can speak for the customer when they aren’t around
  + Works with the customer to write/review stories for the team to work
* Owner of the Backlog
  + Responsible for prioritizing the work
  + Responsible for taking team input, understanding its impacts, and prioritizing correctly
  + Communicating the backlog to all interested parties and can defend the prioritization

Scrum Master

* Defender of the Process
  + Done by educating and coaching the team and leading by example
  + Done by educating and coaching the product owner
* Defending the Team
  + Helping to eliminate roadblocks where possible
  + Helping to decrease interrupts
  + Being an information radiator by keeping externally facing reports up to date
* Holding the Team accountable
  + Making sure the team meets their commitments to increase trust
  + Watching team members hold each other accountable
  + Making sure they work on the agreed to prioritization
* Empower the team
  + Hold the team responsible for following the process while still allowing them to work as they want to accomplish the goals;
  + Ensure they are heard, and where it makes sense, test out the suggested change.

SCRUM Engineering Team

* Understanding the Technologies
  + Taking the time to review or POC items which are new
  + Informing the Product Owner and Scrum Master how this impacts the backlog.
  + Seeing if there are new ways to do the work.
* Owning the work
  + Making sure they understand acceptance criteria
  + Committing to complete the story
  + Working with others to increase their cross-knowledge
* A team of specialists and generalists
  + Can have expert knowledge
  + Pairing so that others can gain the knowledge
  + Need to be able to set up testing environments to make sure the code is stable
  + If they are a tester, gain an understanding of what is reasonable to test
* Innovate
  + Find ways to work better
  + Find other ways to accomplish the customer goals
  + Adopting the process and, after adoption, providing positive feedback to improve the process.

Getting Things Done

**True/False**

Agile/Scrum lets us get things done faster:

False – Agile’s purpose is to find what doesn’t work quickly by producing quick feedback loops with the customer.

Agile/Scrum doesn’t provide the exact time of completion

True – much like ‘traditional’ project management there isn’t a way to give a fixed date. Agile can offer a decent range of the delivery, but the process was designed to deliver value incrementally.

Agile does use estimation and a dynamic ‘velocity’ so that as things change we can provide more immediate updates as to the estimated completion date.

Agile/Scrum handles change easily

True – Unlike other methods, we embrace change – and can do it with a lesser amount of ‘paperwork.’ Having a clear goal, acceptance criteria, and vision is still necessary.

**However,** if there is a delivery window the team must hit, there must be a set of value-driven features for the release. All parties strive only to change those things related to those features within the delivery window.

Agile/Scrum handles interruptions easily

False – Interruptions, by their definition, are detractors from the team’s progress and are not related to the sprint goal or prioritization.

While some interruptions are unavoidable, therefore the Scrum Master and Product Owner need to be part of that discussion. By being in two-week sprints, the question could be – does this need to be done now or can it wait a week for us to start? Because the waste in task switching, interruptions earlier in the sprint it is easier to minimize the impact. When they occur towards the end could have a very detrimental effect on the customers and are discouraged. At that point, teams are focusing on code quality, appropriate testing, and final confirmation that the result meets the expected value.

**How do we predict**

**Prediction**

Prediction is made based on the team's estimate via story points. A story point is a measure of time plus complexity (or risk). It is beyond the scope to discuss story points, but realize that they are like any range measurement; 2 is twice the effort of 1, 5 is more than 3 times the effort of 1, but less than 5 times the effort.

**Note:** Takeestimates made far in advance with a grain of salt. We don’t know what else will come up, get deprioritized, or people’s leave. As a general rule, we don’t get ‘good’ estimates until one or two sprints in advance; architects or other people could put in estimates for what they *believe* the work would be – however this should be on a larger scale than what is used by the team.

The reason why we don’t estimate too far forward with a relative precision is that we don’t want to spend a lot of time doing work that may not be needed as change occurs – that is waste and something that agile tries to reduce. Finally, we don’t predict based upon single expert for several reasons:

1. There is no guarantee that the expert is doing the work. Sometimes you need the expert to do an interrupt, and other team members may need to step up;
2. The team may realize that the expert is the single point of failure and they would choose to take on the work and use that expert as a resource rather than the primary person;
3. The team’s collective recollection of the prior attempt may serve to remind the expert of the previous problems.
4. It’s a team commitment, not a personal one – everyone has to be on board, or finger pointing may result.

**Note**: The Scrum Master does not just sit back and idly record the results as a secretary. They challenge the team by:

* Making sure that those people whose estimates are outside of the consensus get a chance to voice the concerns and be heard;
* Making sure they are not spending a lot of time on story pointing; if there is that much discussion, perhaps the story is not ready for estimation
* Making sure they have broken down a story. If a story seems too large and they can’t break it down into smaller stories, the SM should lead a discussion on why that it is not possible. Even without knowing the technical details, they may be able to hear a logical break the team didn’t catch.   
  If there isn’t a good break, they don’t stop the process but point out the inherent risk of large stories to their completion in a single sprint and the delivery of value.

**Velocity**

Generally, after three sprints (6 weeks), the team will have done enough work to get a good average of the story points per two-week sprint. By having a prioritized and generally pointed backlog, The Scrum Master or Product Owner can do simple division to provide a prediction of the time to complete the known backlog.

Realize that the average story points would include *recorded* interruptions. Additionally, when interruptions come into the sprint, we have to give up something up. You could have the team under commit to account for those interruptions, but it weakens the commitment for several reasons:

1. If the Product Owner knows they are sandbagging, they can either lose faith in the team being transparent or feel that since the team planned for interruptions, they can toss additional work into the sprint with no consequence;
2. The team *may* think that if disruptions do not occur, they can work on their projects. While this is not a bad thing, it *may* lose the *team’s focus* on the sprint goal or decrease the time to reach MVP;
3. Most importantly – if outside influencers know the team is under-committing, they don’t feel pressure. Those individuals should make sure that their interruptions are a high priority. They need to make sure they have a business case to make to the Product Owner and all those involved. The team committed to getting a unit of work done – is your request more critical than that commitment? Can it wait four days to get started?

The final point is why we run two-week sprints; it reduces the pain of the influencers that their work won’t get done because if they ‘don’t get it to them now it will never get done.’ While it may be true, it doesn’t get done any time soon because of the priority, at least they know it’s on the list

“But we need to tell customers when it will get done.”

While true; how did you tell that this in the past? Were we always on time? Did we pad our dates so we’d build in that space for uncertainty? Were we truthful or transparent to the customer? With frequent updates to the customer throughout the process – at a minimum of every two weeks – they can determine if they want to call it “done” and deliver that *value*.

In the past, maybe they wanted to know *when* it will get done, but did it meet the need? Delivering something by a specific date which wasn’t what they wanted isn’t offering value. We had to go back and redo the work which resulted in delays. With fast feedback loops, we determined *what* the team delivered and had the expected value.

We can, with some degree of accuracy, let customers know where we are at and when we might get it done.

**We Estimate at the Last Responsible Moment**

With agile, given that the landscape from the customer changes frequently, we don’t spend a significant amount of time doing the *detailed* estimation. We still need to do some estimation up front, but (again) the is an order of magnitude estimate *after* we have determined the most important stuff to do. As stated earlier, the items we think would be part of the first six weeks would be broken down to their logical, deliverable, work products and estimated accordingly.

Generally, we do not go deeper into the backlog to break down other stories *unless* we know that an item requires us to write another in advance. A story further on the horizon may not be needed as we approach it, or will require changes based on updated information.

Nothing in agile says that we can’t estimate the entire backlog If there is a need. However, there must be an acceptance that some of the work may be wasteful if the client needs change, technologies change, or T-Mobile goes in another direction. Additionally, the team will spend up-front time going through the entire thing, breaking stories down to their logical conclusion – all the while not producing value driven work.

By estimating at the last responsible moment, the customer:

1. Is driven to decide what is most important first;
2. Is encouraged to make sure they fully expressed their needs;
3. Will not wonder how long it will take for stuff to get moving;

**Side note**: With the Product Owner present, they can challenge the team’s estimation if they think they have over or underestimated the level of effort. That is, they can’t argue the number of points, but could say “it seems like it shouldn’t be as hard as other stories of the same size.” The Product Owner, looking at the size of the work, may realize:

1. Something was unclear in the story. The lack of clarity caused the team’s estimation either included or excluded effort that the wasn’t intended. This activity allows the team to break down a story or go back ‘to the drawing board;’
2. The customer was asking for something that is complex and perhaps they need to take a smaller step. Again, back to the drawing board;
3. They need to deprioritize the story because there are other things which should take precedence based upon the size.

Stakeholder Responsibility

The Scrum Team and the Product Owner collaborate to ensuring the team completed the stakeholder’s ask. It may be that the Product Owner examines the stories as the team completes them mid-sprint, providing feedback early to resolve discrepancies. Sometimes the Product Owner only gets them at the end of the sprint. Regardless, the stakeholder becomes involved to ensure the work met their definition of value.

To that end, the stakeholders must take time every couple of weeks to be present at the Sprint Review and provide feedback on the team delivered features. Lacking that Agile, like any other methodology, fails because the work wasn't validated early enough to account for the change.

Commitment

Agile requires commitment from all parties, from the Management on down to the engineers.

The managers must be committed to the process, realizing that Organization Change is difficult and takes time to get done. They must also recognize that they are influencers as opposed to directors of the work.

The Stakeholders also need to release their control of time and make sure their requests are clear enough that when the team delivers the perceived value, that it matches their work.

The Product Owner (or Agile Project Manager) must take responsibility for being the Voice of the Customer (or Stakeholder) and ensure that the work is in its value order as described.

The Scrum Master is committed to:

* Protecting the team (including the PO as needed) from undue interruptions,
* needs to know what is going on from an overall perspective to watch for impediments – and escalating as appropriate,
* encouraging and holding the team to their commitments,
* coaching the team on the process and guiding them on the journey.

The team is committed to

* Helping each other when needed,
* Creating value for the stakeholders,
* Enhancing meaningful and clear stories so that the PO can ensure that what the team is building matches the value,
* Completing the teams committed work by helping each other out as needed.

**Everyone is committed to**

* **Continuous Improvement,**
* **Working the Process Long Enough to see what works,**
* **Coming from a Good Place by Listening to Each Other**