



EIGHT AREAS OF AGILE TRANSFORMATION

By Preeti Tanakanti

Introduction and Overview

Traditional Waterfall models, although still useful in many situations, have not been able to keep up with the industries' current push for increased speed to market, project flexibility, and the breaking down of departmental silos. The outcome of the push to Agile can either be the elevation of your organization to a modern productivity powerhouse or throwing your organization into utter chaos. This paper offers guidance and advice on your Agile Transformation by providing high-level direction on the Eight Areas of Agile Transformation.

Areas of Focus for a successful Agile Transformation

We will focus on these main areas of any group to set the stage for an organized transformation. We will be focusing on your organization's thought processes around execution and a core belief in a team driven approach. Commitment and trust during your transition increases throughput to stakeholders and end users, resulting in faster feedback cycles. Change is tough to embrace but both success and failures should be celebrated. You will gain the space you need to take risks, fail as a regular part of your process, and course correct to greatness.

In Agile, changes are expected, frequent, small, and lithe, and a well-planned Agile transition will leverage these elements while minimizing the inevitable pain of change.

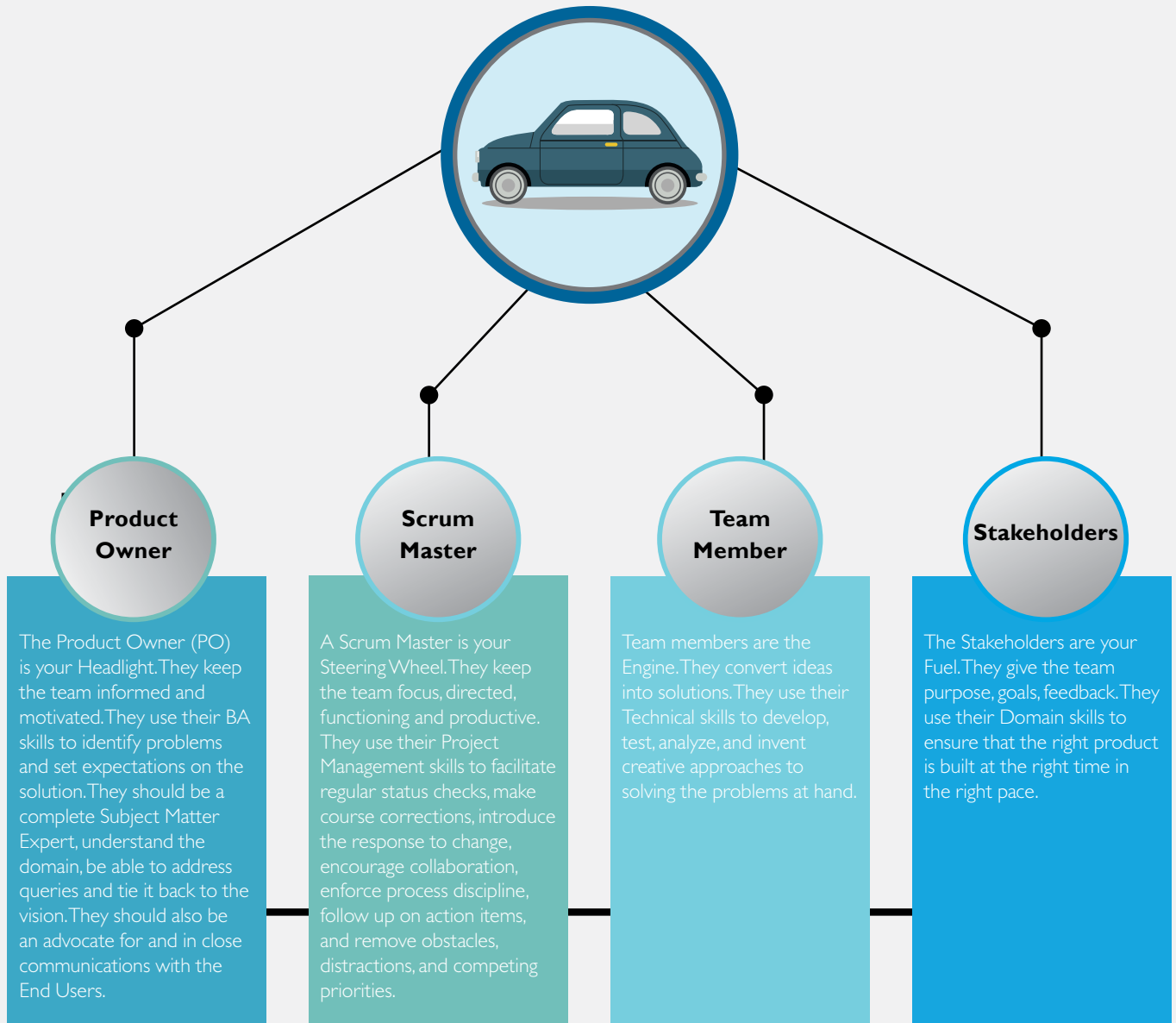


1. The Team

Your Team is the heart of your Agile success. Great plans have often failed without the right people and great people have failed without the right distribution of responsibilities. Every member of your team should understand their role and how it intersects, supports, or feeds into the roles of the other team members. A great team should work like a well-trained performance with hand-offs happening smoothly, decisions handled quickly, and changes are responded to efficiently.

Team Roles

Think of your project like a car. Every member of The Team has a part to play to keep the car going in one direction, as quickly and as efficiently as possible. Enhancements for device drivers for STBs, televisions, automotive multimedia systems, and healthcare systems among others. Drivers include power management, Ethernet, audio, video, graphics, storage, and other peripherals such as I2C, SPI, UART, Parallel ports.



Fitting the Team together

Individual team member descriptions must come together into a predictable and reliable foundation on which you can build a fast-moving, flexible conduit for a project. The table below illustrates the interconnectivity of these teams.

Phase	Responsibilities	Product Owner	Scrum Master	Team Member	Stakeholder
Product Discovery	Elaborate the vision	X	X	X	X
	Identifies User Personas	X			
	Builds Story Map	X			
	Creates Product Road Map	X			
	Writes User Stories	X			
	Grooms and Estimates First Release	X	X	X	
Sprint Planning	Specifies story acceptance criteria	X			
	Identifies Sprint goals and backlog	X		X	
	Grooms Sprint User Stories	X	X	X	
	Estimates User Stories			X	
Daily Scrum	Participates Daily	X	X	X	
	Answers questions and resolve queries	X	X	X	
	Verifies stories against acceptance criteria	X			
	Evaluates the product at end of Sprint	X			
	Facilitates resolution of Sprint impediments			X	
Sprint Review & Retrospective	Presents a demo to Stakeholders	X			
	Collates feedback from demo	X		X	
	Participates during Retrospective		X	X	
	Analyses Sprint metrics		X	X	
General	Co-located during critical phases or meetings	X	X	X	
	Overlaps workhours with offshore for communication	X			
	Presents offshore' s questions to onshore	X			
	Holds detailed Release planning sessions	X			
	Meets monthly with the team to review backlog	X			
	Co-located with team to efficiently run the process		X		
	Drives the meeting and ceremonies schedules, keeping in mind the time zone differences		X		
	Is aware of offshore dependencies		X		

When moving your Team to Agile, remember...

An ideal scrum team should be right-sized, self-organized, empowered to determine the goal of the sprint, able to identify the work results, organized itself towards the result, and has the right to do what it takes to make it succeed, within the boundaries of the project rules. The Team should have the right preparation, the right people, and a solid mindset of Agile best practices.

2. The Framework

Think of your "Vision" as an idea of a better future for your users that that drives brand loyalty. It could be derived from multiple visionaries but there must be one unified Vision that leads to your Minimum Viable Product (MVP). While most people use a word template to capture their vision, a Product Vision Board (like the one below) can help put things into perspective.

3 Areas of Work

- A simple goal – Deliver maximum value with high predictability to the stakeholders
- A simple structure – Set a goal, plan toward it, create demonstrable code, Review and Repeat
- A simple set of rules - Visible progress, improve quality, team accountability, measure progress by finished work

3 Categories of Work

- Opportunity Backlog – the source of all valuable product and feature ideas.
- Product Discovery – Research, exploration, validation and delivery of product solutions. The outcome of product discovery yields product goals, user story maps, simple personas and user interface sketches.
- Product Delivery - Incrementally build and validate product solution

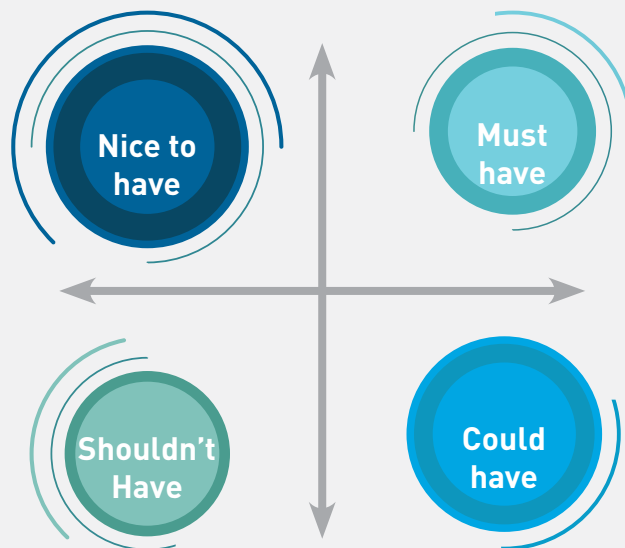
Product Vision Statement <Explain in a simple statement>			
Target Group <Who are the end users/customers?>	Needs <What is the problem you are trying to solve? What is the goal that needs to be achieved?>	Product <What are the top 5 features that can help address the needs?>	Value <What is the value delivered? How does it help business?>
Any impediments?	Any impediments?	Any impediments?	Any impediments?

The content from the four columns should be validated back with the Vision statement so that ideas are focused on achieving the Vision and not diluted with other viewpoints.

Minimal Viable Product

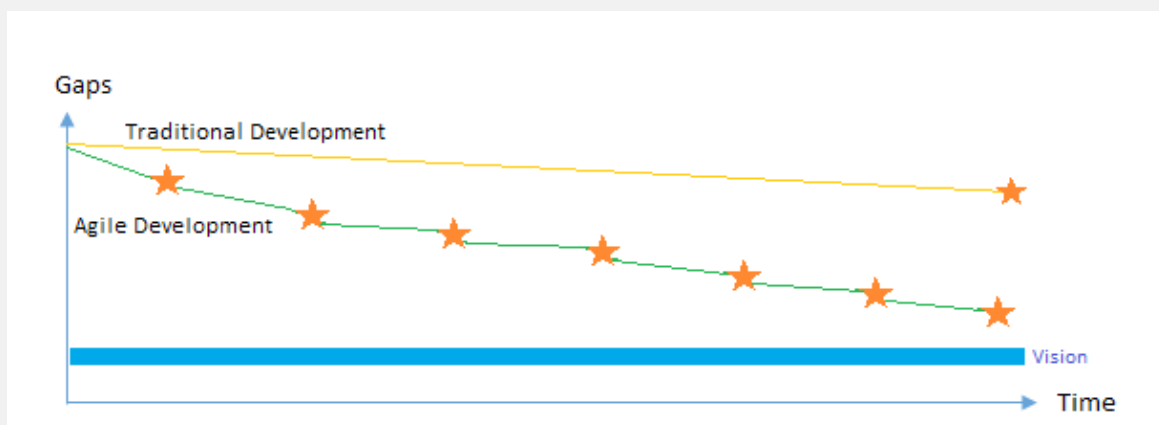
A minimum viable product (MVP) is the most important, bare minimum features that is of use to your end users. It is designed to be shipped off anytime to the end user. When defining what constitutes a MVP, it is useful to ask your customers what is most important amongst a list of pre-qualified features put into 4, prioritized buckets.

Once the priority is set, you can build a backlog of tasks needed to get the maximum value. Once the Must Have and Could Have are developed, we wait for feedback, adjust and then add more features to gain better value.



Releases

Agile encourages you to release early and get feedback that can be adapted into the next release. While traditional methods do a lot of planning towards a release before implementation, Agile gives us ways to get it done faster and better with focused planned and realistic time lines (Refer Section 5). The idea here is that the path towards the actual final product converges over time.



In the figure above, the stars indicate releases. In a traditional model the release happens at the end and since the requirements are already set, the gaps are significantly large. However, in the Agile mode each release receives feedback that is then incorporated back into the implementation cycle. This reduces the gap with every release over time.

When moving your Framework to Agile, remember...

- For a project already in execution, map out the requirements into Epics, Features, and Stories.
- Break down the stories and validate them against the vision.
- Tag the stories that need to go into the first MVP.
- Implement the user stories of the MVP, based on priority and team input.
- During the demo, add back all stakeholder inputs into the backlog.
- Focus on incremental delivery that is tested and demoed at every sprint.

3. The Backlog

The backlog is a prioritized list of user stories that will bring the product to life. A backlog is driven by the factors of market, user needs, or the business goals just like the requirements of any other project. Based on the priority, the backlog is implemented sprint-wise and demonstrated at the end of each sprint. Any feedback arising from the demo should be incorporated back into the backlog.

Characteristics of a Backlog

A Backlog typically comprises of Features, Bugs, Spikes (research required to make future work feasible), technical work (infrastructure that supports the user story), and training (optional, but helpful to keep everyone on the same page).

A Scrum has two types of backlogs: the product backlog containing Product Backlog Items (PBI) and the sprint backlog ties back into the Release planning sheet and clearly identifies the Sprint goals, the size of each sprint, the releases they belong to and their status.

User Stories

A feature translates into one or more user stories. The PO is primarily responsible for writing User Stories in a location that is easily accessible to all roles within the Agile team. A good user story must follow the 3 Cs, the INVEST principle, capture clear acceptance criteria, and follow one of the Story slicing technique as required.

The Three C's

- Smallest piece of functionality that adds business value and should flow the 3 Cs:
- Card – A placeholder for conversation
- Conversation – Discussion between Development and User
- Confirmation – Acceptance criteria to determine when a story is considered finished/done

INVEST Model

- All good user stories need to follow the INVEST principle. They should be:
- I – Independent
- N- Negotiable
- V – Valuable
- E – Estimate-able
- S – Small
- T – Testable

Acceptance Criteria

- Set of conditions that the story must meet in order to be considered complete.
- Not tests but provide inputs to testing and must contain an Actor, Verb, and Observable Result

Story Slicing

- Work flow steps
- Business rules variations
- Major effort
- Variations in data
- Data entry method
- Deferred system qualities
- Operations(CRUD)
- Use case scenarios
- Break out spikes

User Story Mapping

User Story Mapping is an approach to organize and prioritize user stories. It helps to visualize the workflow and the relationships between a parent user story to its children. From a broader view point, it confirms the completeness of your backlog and provides a useful context for prioritization. This enables to plan releases in valuable slices of functionality. The stories are then validated with the goals of the release and tied back into the sprint planning. Children Stories are prioritized and put into sprint buckets based on team's decision for implementation.

When moving your Backlog to Agile, remember...

- Assess the top 30 requirements in the project and break them down into User Stories that cover only one simple functionality.
- Converse with Product Owner and Users to cover all aspects of the identified User Story.
- Define the user story: "As a <user>, I want to be able to <function> so that <supporting need>."
- Acceptance criteria must be written before the start of the sprint.
- Estimate conversations should be recorded in notes attached to the story
- The PO must defer and unclear/undefined User Stories.
- A story should never enter a Sprint without a decided and understood acceptance criteria.
- It is a good practice to be able to tie back a User Story to a specific feature.

4. The Sizing

Story Points are a unit of measure to indicate the size of a User Story. When User Stories are compared and relative sizing is assigned then the total overall understanding of effort is gained within the team.

Figure 5 Guidelines for Sizing

Goal

- *Rough project size*
- *Project feasibility*
- *Performance Baseline*

Process

- *Separate risk from estimation*
- *Get everyone's buy-in*
- *Prioritize what's important*
- *Highlight unstated assumptions*

Estimation & Velocity

- The size of a story is based on 4 key factors: **Volume:** How much? - **Complexity:** How Hard? - **Knowledge:** What's known? - **Uncertainty:** What's unknown?
- Agile recommends using the Fibonacci series to estimate the size of a story (1, 2, 3, 5, 8, 13 ...). It captures the inherent uncertainty in estimating the stories.
- Velocity is the total story points of all the stories completed during a Sprint. Knowing Velocity is key to planning for future iterations since the team will aware when they are taking on more than what their previous velocities would allow.

Poker Estimation

Poker estimation combines expert opinion, disaggregation, and analogy for quick and reliable estimates. The rules of poker estimation are:

- Include all team members
- Each estimator is given a deck of cards with the Fibonacci sequence
- PO participates but does not estimate
- SM estimates only if he/she is involved in development work
- PO reads the description of the story from backlog to be estimated
- Questions are asked and answered
- Each estimator privately selects a card
- All participants show their cards simultaneously
- High and low estimators explain their estimates
- After discussion, participants re-estimate
- The estimates are expected to converge; if they don't repeat the process

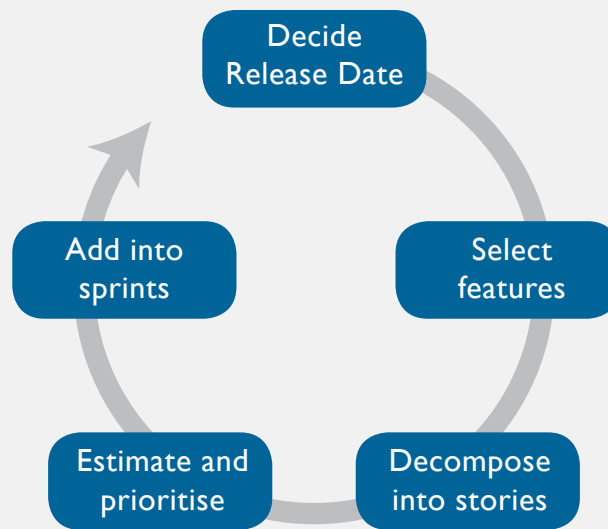
When updating your Sizing to Agile, remember...

- Ensure the biggest story size is 8 points or less.
- A 1 story point story should take about ½ a day for development and ½ a day for testing.
- While it is difficult to move from hour estimate to story pointing, it is helpful to time box development and testing tasks in the first few sprints
- A story that is coding complete but not tested should not be broken down just to add to the velocity. The whole story will need to be moved to the next sprint. It will normalize over time.

5. The Plannin

Release Planning

A scope-boxed Release Plan prioritizes the features over a fixed date. A time-boxed Release Plan focuses on calendar dates. Both cases influence the importance given to a feature in different ways. In many cases, a time-boxed plan is useful in allowing investors to visualize decisions that realize the most value in the delivery. The time-box Release Plan starts with the decision of a release date. Using the Product Vision as a guide, the minimum marketable features are selected. They are decomposed into high level stories and estimated. The stories with the highest value and the lowest cost are pushed to the top of the release. The analysis continues with moderate value & moderate cost user stories and so on. Taking into account the past velocity, risks, dependencies and story estimates, a prediction can be made on how many user stories can be completed in each sprint. The sprints are then filled up with these User Stories till the Release date is reached.



Iteration Planning

Related User Stories are bundled into a Release. Always retain flexibility to prioritize future stories in a Release, as the need arises. Any change should get into the backlog as a new story.

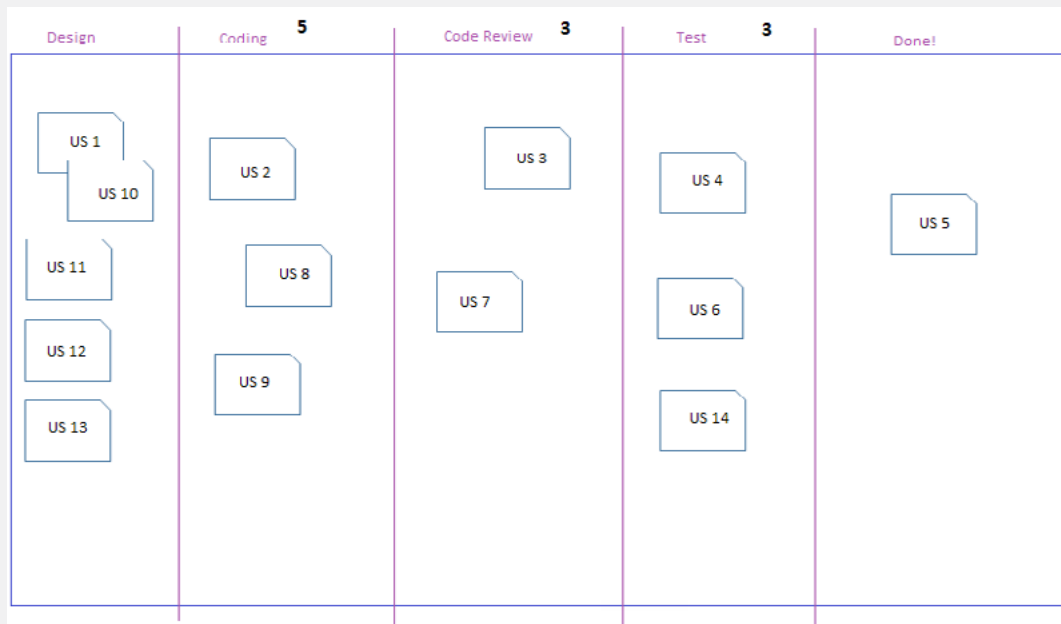
Figure 6: Planning Legend

Use a Focus Factor (0.6 - 0.8)	Account for leaves and holidays	Task should complete in a day	Tasks should facilitate swarming	Use a "Done thinking Grid"	Keep it consistent	Make it visible(try a Wall Grid)	Review during stand ups	Commitment driven planning	Sanity check the commitment
Capacity Planning		Task Breakdown		Definition of Done		RAID Log		Individual Commitment	
Planning Legend									

Post Planning

Team members volunteer on the stories planned for the sprint and take control of their work. During this entire session, a Kanban board should be used to indicate the flow of user stories within the iteration. Each stage within the board should have a Work in Progress (WIP) limit that encourage the team to collaborate in a streamed fashion and prevent multiple items from clogging up the iteration. A sample board is illustrated here.

In this example, there is a work limit of 3 on the Test phase. This means that team members who have bandwidth will volunteer to take on additional code review tasks so that the stories can continue moving to the Accepted Phase. Having lower WIP limits enhances more collaboration and throughput but having it too low might dilute the focus of the developers. Having too high WIP limits will clog some stages and make it cumbersome to manage. An ideal balance can come from experience and can vary over time to nail down the right numbers.



When updating your Planning to Agile, remember...

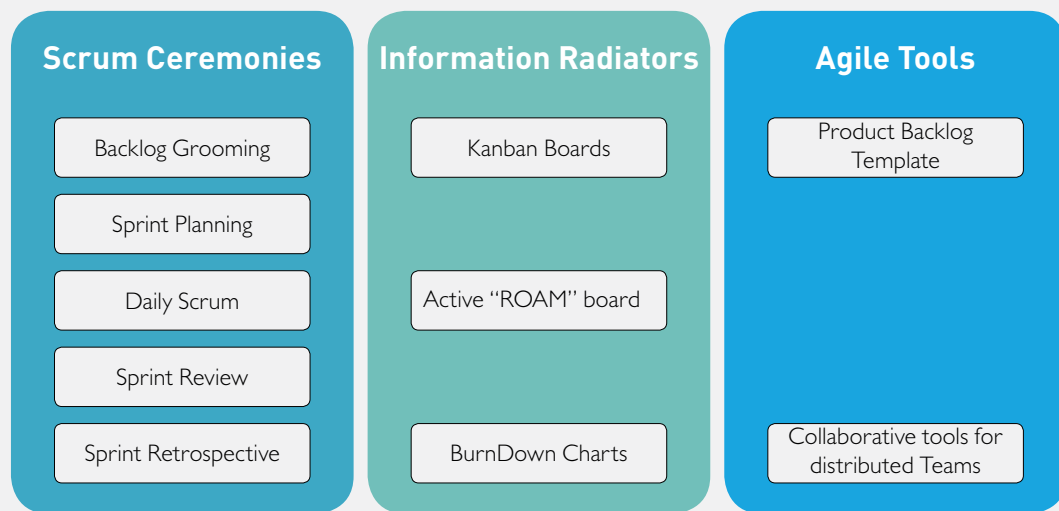
- Ensure you have a Release Goal for every Release.
- The Release Goals must tie back to individual Sprint Goals.
- The Planning Legend needs to be discussed with the team to have a common understanding of the process and the expectations.
- Planning is effective only if you are able to release a product that has value proportional to the investment made to create it.
- Planning tools must be visible to all members of the team. Walls in works areas serve this purpose well. For distributed teams, online tools such as Slack or Trello should be considered.
- Stages on the Kanban board can also include exit stages such as Ready for Coding or Ready for Code review. These stages are like a holding areas that give better visibility on tasks ready to enter the following stage.

6. The Execution

Scrum ceremonies set the cadence for execution while viewing its progress can be achieved through several enablers. All the ceremonies should be facilitated by the Scrum Master, maintain mandatory participation by all team members, be time-boxed, and have a specific agenda.

Information Radiators should be big and visible (often referred to as “big visible information radiators” or BVIR). They support the theory of transparency in agile to keep everything known to everyone on the team. The team should have sufficient information be able to proactively offer help when needed so that, as a team, they are able to do whatever it takes to make it happen.

Agile tools enhance collaboration amongst the team. There are excel templates available for most artifacts such as Product Backlog, Release Planning, or Product Vision. Other tools facilitate communication and transparency between teams that are not co-located.



Sprint Zero

Sprint zero is the first Sprint of the release and should focus on setting the framework for the project. The focus should be to put a minimal design in place that future sprints can build upon to create more value. This covers development, testing, Infrastructure, and quality.

Development Track

- User story mapping and requirements detailing that results in a Release Plan sign off and an MVP Sign Off
- UI Prototyping
- Architecture and Design (High level Architecture, Conceptual Data Model, Spike POCs for Validation of design)

Testing Track

- User Story Mapping
- Test Strategy, Test automation feasibility, Spike POCs for validation scripts and tool validation
- MVP Sizing

Infrastructure & Quality

- Software Quality tools
- Setup of continuous integration, build management, code quality
- Setup of Agile lifecycle management and dashboards

Iteration Execution

During the iteration, the Kanban board needs to be monitored to identify areas of growing concern such as underestimations, lags in the project flow, or resource underutilization. This view prompts the team to stop, think, and act proactively.

A commitment is made by the team to ensure they'll do everything to complete the stories but when challenges are encountered, there are three options for resolution:

1. Change the attack to bring the team back on track. Ex: Perhaps through dev/QA pair work
2. Review the scope for reduction or tasks simplification.
3. Consider splitting a story.

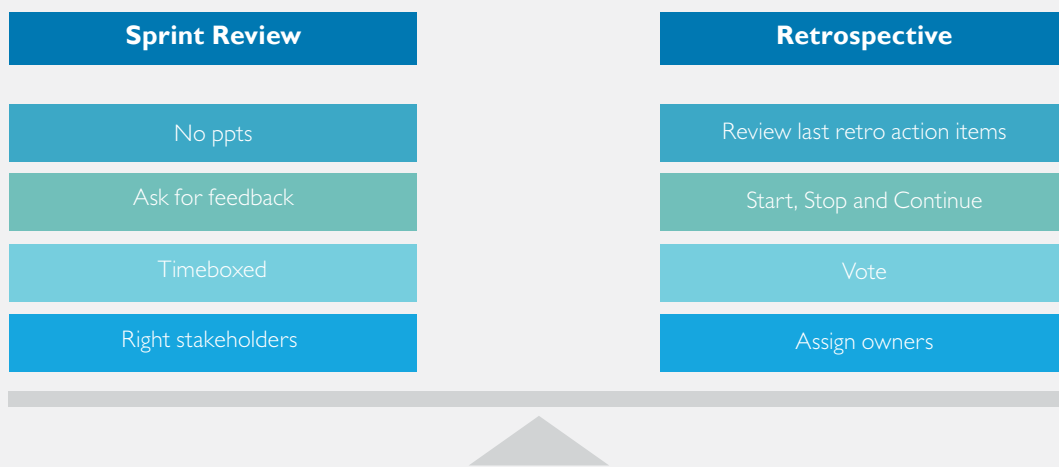
If you are able to succeed after changing the plan the iteration is still a success. However, the reasons for change must be communicated to the stakeholders along with an action plan to prevent similar issues in the future. In no case should the iteration date be moved. Some teams might cancel or admit a failed sprint when things are beyond the point of recovery. After all, no team is perfect. If this is a rare occasion, learn and move on. If it's too often, an in-depth analysis with an experienced mentor would be advised.

When updating your Execution to Agile, remember...

- Scrum ceremonies need people who will dedicate time to the activity and need the entire team to be present.
- Grooming must happen before the start of the Sprint and should target to groom stories at least for the next 2 sprints.
- Planning should happen on the first day of the Sprint.
- Reviews should be 1 hour for every week of the sprint. (Ex: 2 week sprints should have a 2-hour review)
- Demos can be recorded for later reviews or for absent team members.
- All Information Radiators should be in a common place on a wall or a meeting room that the team uses. It never comes down and is constantly updated by everyone in the team at least once a day.
- Agile Tools like JIRA or Rally may be procured but there are lots of templates that can be built using excel spreadsheets that can be revised over time. Whatever the medium, it must be in a common location for the team to access at will and check for updates.
- Sprint Zero does not need:
 - The entire team recruited and onboard.
 - To deliver value to the customer. (focus is on delivery of a framework for future sprints)
 - To have in depth design. (a scalable design is all we need to set up)
- In case of distributed teams, a seeding visit to the client place is recommended during critical phases of the project. This could be the whole team or a few members of the team.

Closing Ceremonies

The closing ceremonies include the Sprint Review and the Sprint Retrospective. The Review is intended to get feedback from the business for The Team and the Retrospective is to get feedback from The Team on how things went in the sprint. A big part of Agile is to evolve from listening and adjusting to align closer to the vision.



An effective Sprint Review usually has little or no presentation slides. The focus should be on a working application. The entire demo should be time-boxed with the right audience. If someone cannot attend, a backup representative from the business side should be made available.

The Retrospective ceremony should start with the review of new and existing action items and their current status (Stop, Start, and Continue). The group votes on the listed items and assigns owners to the top 3 items. During the Retrospective, consider the Speed boat technique. In this technique, the facilitator draws a Speed boat. Feedback is now received which manifests correlates to:

- A Propeller (what drives you forward)
- Life preserver (what saves you)
- Anchor (what holds you back)
- Rocks (where you can crash)

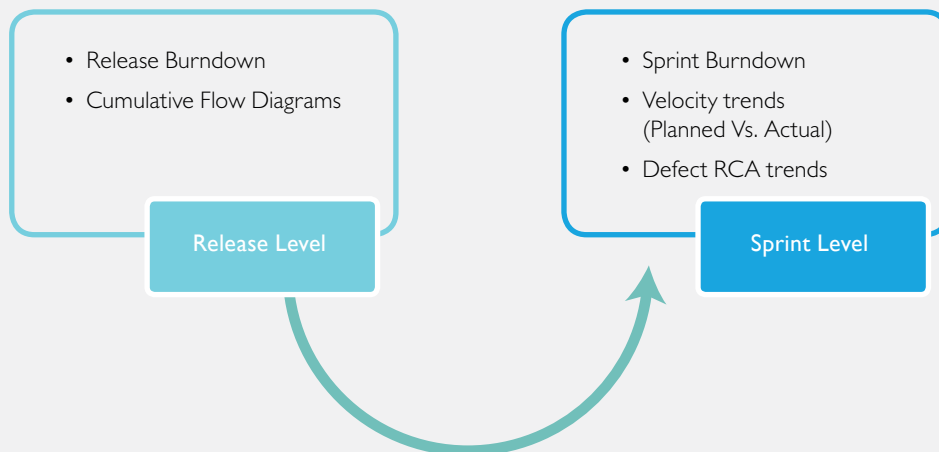
This is then voted upon, assigned owners, and reviewed in the next Retrospective meeting.

When updating your Closing to Agile, remember...

- Sprint Review
 - The team should not spend more than 2 hours preparing for a Sprint Review meeting.
 - The Scrum Master is a silent participant and the PO drives the meeting.
 - The demo is not aimed at getting signoff on the development but to get feedback from a focus group discussion.
 - Put all feedback back into the backlog.
 - Ensure that all stakeholders are able to view the demo. (Use a screen sharing mechanism in case of distributed teams.)
- Retrospective
 - The Scrum Master should drive this meeting.
 - If a team is uncomfortable in speaking out, use an anonymous approach.
 - The top 3 voted items must be worked diligently for improvement or the team might lose faith in the ceremony.

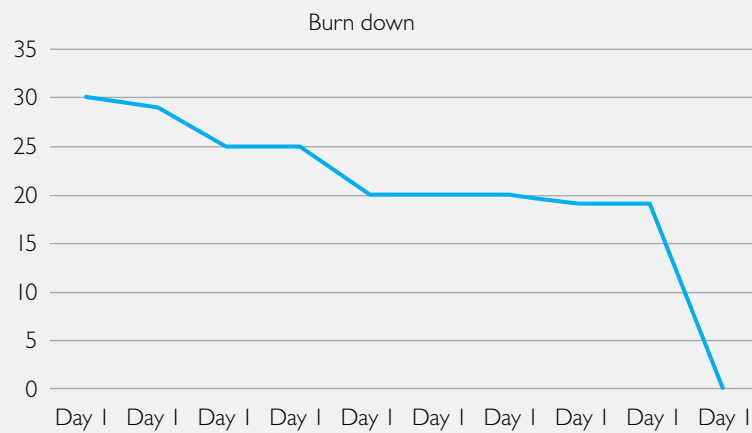
8. The Metrics

While getting all the stories “Done” in a Sprint is a healthy trend, there has to be an objective way to measure progress. Release Burndowns, CFDs reflect the macro views while Sprint Burndowns, Velocity and RCA Trending help us monitor the micro. This information is useful to the entire team to gauge on how far they are from the next version then want to take to the end customers.



Sprint Burndown

Release and Sprint Burn downs are an effective way of tracking how steady the progress is towards the goals. In an ideal world, it is never a linear line but, from a Best Practices standpoint, it should also not drop down vertically on the last day of the Sprint.

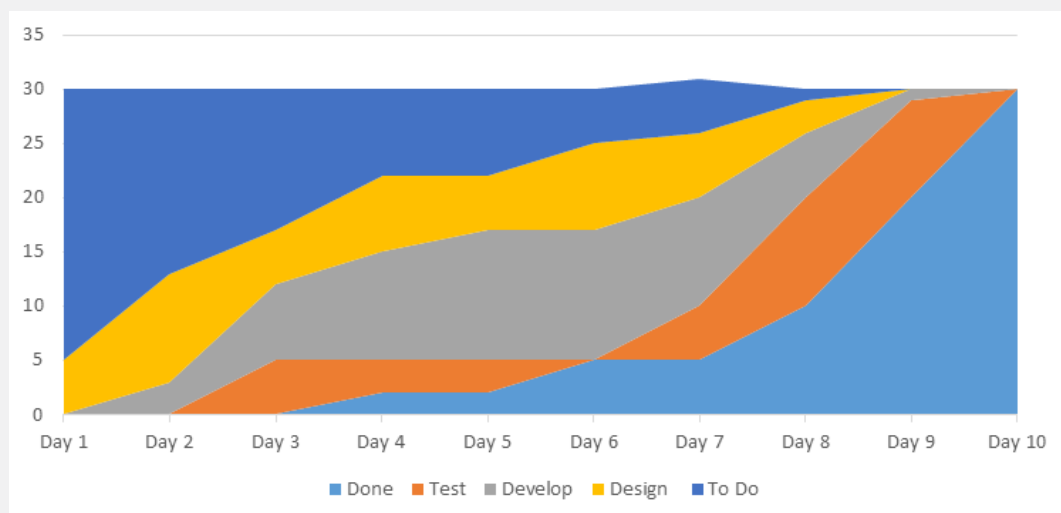


That unhealthy pattern indicates that the team is waiting till the last day of the sprint to close all user stories. This could be because either the WIP limits are not ideal or the team is actually following a mini waterfall model rather than doing Agile development. In some cases it actually helps to break user stories into tasks and then assign them individual story points. This ensures that every task burnt is reflected in the burn down chart and shows the steady progress towards the goal (which is what we should realistically see).

Release Burndown charts gives a chance to take action proactively in case the features are not burndown as planned. This de-scoping will keep stakeholders involved as they should also see these metrics on a regular basis.

Cumulative Flow Diagrams

A CFD aims to make visually clear the blocks in the execution of a sprint. Whenever a band of stage starts to narrow it's an indication that the stories are not getting done in a way that flowing through the stages of the project.



In the example above, it's clear that on day 6, nothing moved into the Test phase and the bulk of effort was towards development. The CFD charts also provide a way to calculate the cycle time to the "done" graph on any particular day. This allows for predictability of how many stories could get done by the end of the sprint. The responsibility of monitoring these charts lies with both the scrum master and the team and should be mandatorily reviewed during the retrospectives.

When updating your Metrics to Agile, remember...

- Keep metrics to a bare minimum.
- You should be able to answer the question, "Exactly what question is my metric answering?"
- Be on the lookout for unhealthy patterns.
- It's a good idea to review metrics with stakeholders once a month, pause, analyze, and respond.
- Metrics must be made available to all members of the team.

Conclusion

The journey to Agile is not something that happens overnight. There is apprehension around how to move existing projects and how to align the project team members, management, and processes in order to meet the Agile vision. While the decision to move or to convince management may seem difficult, the shift is really a learning process for both novices and experienced team members.

It is also worth mentioning that no two projects are the same. It's not just the business priorities that change, it's also the people. The beauty of Agile is that, along with providing the template, guidelines, and frameworks, it also provides the flexibility for the team to take control. Team members who are passionate about development will thrive in such an environment where innovation is rewarded and applauded.

We have provided some advice on how HARMAN has successfully guided our clients in their transition to Agile and helped our client's teams appreciate the full potential of agile over time while encouraging others in the team to provide suggestions and contributions that could help further their vision. We are proud of our success but even prouder of theirs.



**Success is not final, failure is not fatal:
it is the courage to continue that counts.**

- Winston Churchill

Partner with an industry expert

HARMAN Connected Services is design led engineering services partner. As a wholly owned subsidiary of HARMAN International we deliver next generation enterprise and consumer software solutions. We help organizations harness digital infrastructure and agile methodologies to build modern, scalable and intelligent platforms. With help of our prolific partner ecosystem and flexible engagement models, we enable our customers to hyperscale their business with outcome certainty.

Visit our website at <http://services.harman.com>

