

# **Agile Group Model**

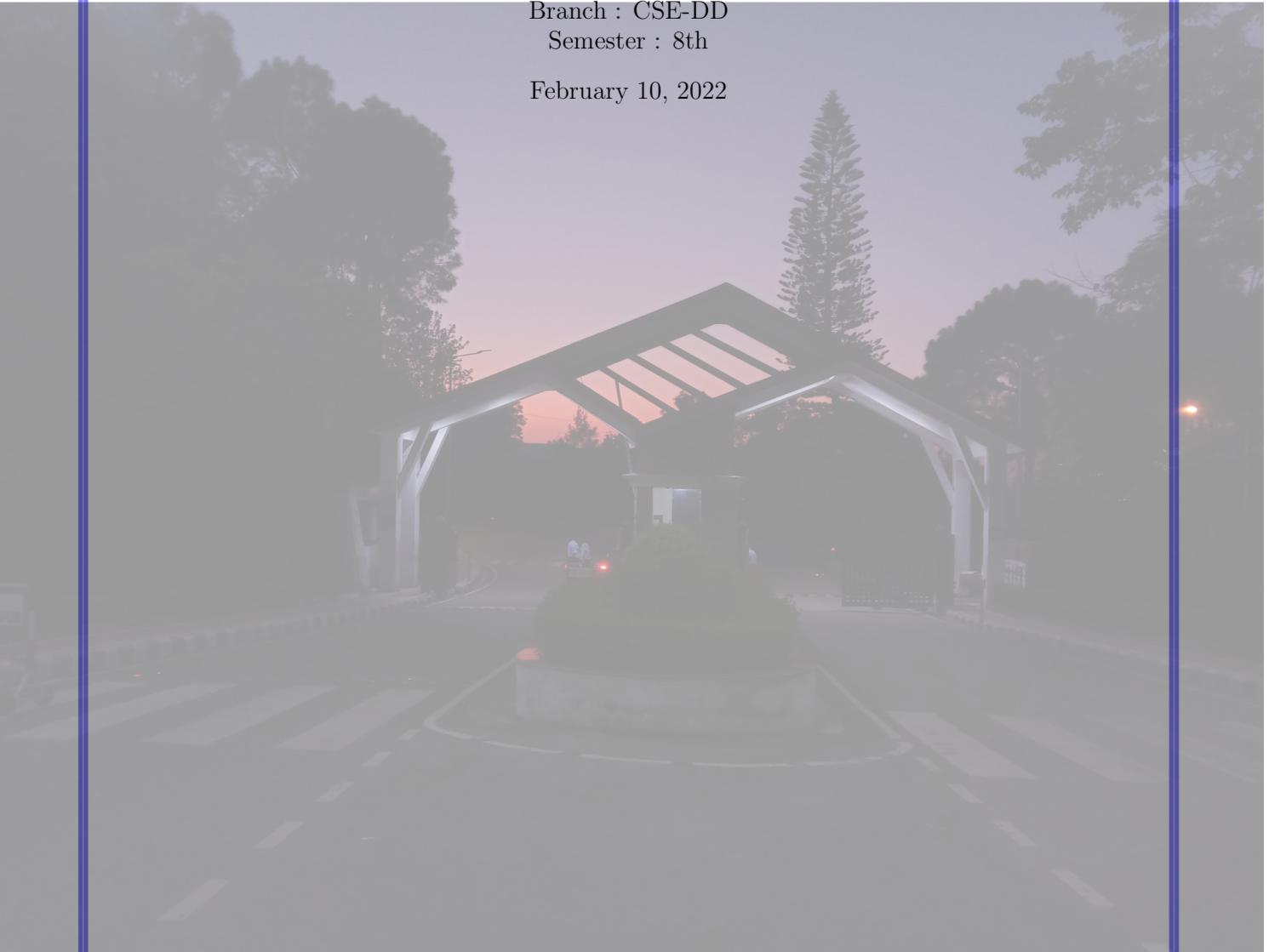
Abhishek Guleri 185509

Kritarth Kapoor 185510

Branch : CSE-DD

Semester : 8th

February 10, 2022



# Contents

<b>1 Agile</b>	<b>3</b>
1.1 What is Agile? . . . . .	3
1.2 Why choose agile? . . . . .	3
1.3 Agile yesterday, today, and tomorrow . . . . .	4
<b>2 Agile Manifesto</b>	<b>4</b>
2.1 The origin story . . . . .	4
2.2 12 Agile Manifesto principles: a culture, defined . . . . .	4
<b>3 Agile Group Model</b>	<b>6</b>
3.1 Scrum . . . . .	6
3.1.1 What is Scrum? . . . . .	6
3.1.2 3 fundamental principles in scrum agile method . . . . .	7
3.1.3 Three essential roles for scrum success . . . . .	7
3.2 Scrum ceremonies . . . . .	8
3.2.1 Sprint Planning . . . . .	9
3.2.2 Daily stand-up . . . . .	9
3.2.3 Iteration review . . . . .	10
3.2.4 Retrospective . . . . .	10
3.3 Kanban . . . . .	11
3.3.1 What is kanban? . . . . .	11
<b>4 Kanban Vs. Scrum</b>	<b>13</b>
<b>5 Kanban vs. scrum: What if you can't choose?</b>	<b>13</b>
<b>6 Agile Project Management</b>	<b>14</b>
6.1 What is agile project management? . . . . .	14
6.2 Estimate, report, and plan . . . . .	14
6.2.1 Agile project estimating . . . . .	15
6.3 Agile reporting . . . . .	15
6.4 Backlog management and grooming . . . . .	16
<b>7 Agile Management Tools</b>	<b>17</b>

# 1 Agile

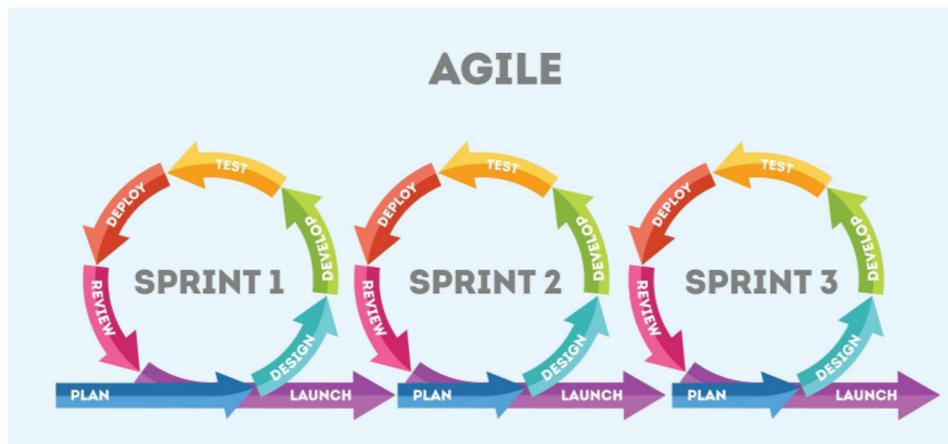
## 1.1 What is Agile?

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Whereas the traditional "waterfall" approach has one discipline contribute to the project, then "throw it over the wall" to the next contributor, agile calls for collaborative cross-functional teams. Open communication, collaboration, adaptation, and trust amongst team members are at the heart of agile. Although the project lead or product owner typically prioritizes the work to be delivered, the team takes the lead on deciding how the work will get done, self-organizing around granular tasks and assignments.

Agile isn't defined by a set of ceremonies or specific development techniques. Rather, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement.

The original Agile Manifesto didn't prescribe two-week iterations or an ideal team size. It simply laid out a set of core values that put people first. The way you and your team live those values today – whether you do scrum by the book, or blend elements of kanban and XP – is entirely up to you.



## 1.2 Why choose agile?

Teams choose agile so they can respond to changes in the marketplace or feedback from customers quickly without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets your team gather feedback on each change and integrate it into future plans at minimal cost.

But it's not just a numbers game—first and foremost, it's about people. As described by the Agile Manifesto, authentic human interactions are more important than rigid processes. Collaborating with customers and teammates is more important than predefined arrangements.

And delivering a working solution to the customer's problem is more important than hyper-detailed documentation.

An agile team unites under a shared vision, then brings it to life the way they know is best. Each team sets their own standards for quality, usability, and completeness. Their "definition of done" then informs how fast they'll churn the work out. Although it can be scary at first, company leaders find that when they put their trust in an agile team, that team feels a greater sense of ownership and rises to meet (or exceed) management's expectations.

### 1.3 Agile yesterday, today, and tomorrow

The publication of the Agile Manifesto in 2001 marks the birth of agile as a methodology. Since then, many agile frameworks have emerged such as scrum, kanban, lean, and Extreme Programming (XP). Each embodies the core principles of frequent iteration, continuous learning, and high quality in its own way. Scrum and XP are favored by software development teams, while kanban is a darling among service-oriented teams like IT or human resources.

Today, many agile teams combine practices from a few different frameworks, spiced up with practices unique to the team. Some teams adopt some agile rituals (like regular stand-ups, retros, backlogs, etc.), while others created a new agile practice (agile marketing teams who adhere to the Agile Marketing Manifesto).

The agile teams of tomorrow will value their own effectiveness over adherence to doctrine. Openness, trust, and autonomy are emerging as the cultural currency for companies who want to attract the best people and get the most out of them. Such companies are already proving that practices can vary across teams, as long as they're guided by the right principles.

## 2 Agile Manifesto

### 2.1 The origin story

In early 2001, against the backdrop of the Wasatch Mountains, in Snowbird, Utah, 17 people met to discuss the future of software development. The group's members shared a frustration about the current state of affairs, even if they disagreed about how to remedy the situation.

The problem, they agreed, was that companies were so focused on excessively planning and documenting their software development cycles that they lost sight of what really mattered—pleasing their customers.

Companies may have touted corporate values like "excellence" and "integrity," but these values did little to guide people—especially software developers—toward a better way. That needed to change. Many of the Snowbird 17 already had ideas about how to usher in software development's new era. The trip to the mountains was their chance to hash it out.

The Agile Manifesto emerged from this extended weekend at just 68 words, and the short and sweet document went on to change software development forever. In the nearly two decades since its creation, these words (and the 12 principles that follow) have been embraced (in varying degrees) by countless individuals, teams, and companies.

### 2.2 12 Agile Manifesto principles: a culture, defined

Today's agile landscape can seem cluttered with methodologies that promise to take agile ideals and turn them into real-world realities. But today's methodology madness isn't anything new.

The Manifesto itself was born out of a need to find a common ground among scrum, Extreme Programming, Crystal Clear, and other frameworks.

"They were starting to see that there was something common that they were doing. But at the time, they were very much competitors, at least competitors in thought," said Ian Buchanan, Principal Solutions Engineer for DevOps at Atlassian. "When you put that into context, the fact that they could agree on some set of anything is kind of profound."

The Snowbird 17 wanted to see if representatives of their different disciplines could agree on something—anything. And to their surprise, they did. They agreed on a set of values that defined a culture.

Here they are:

## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

## 12 Principles behind the Agile Manifesto

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

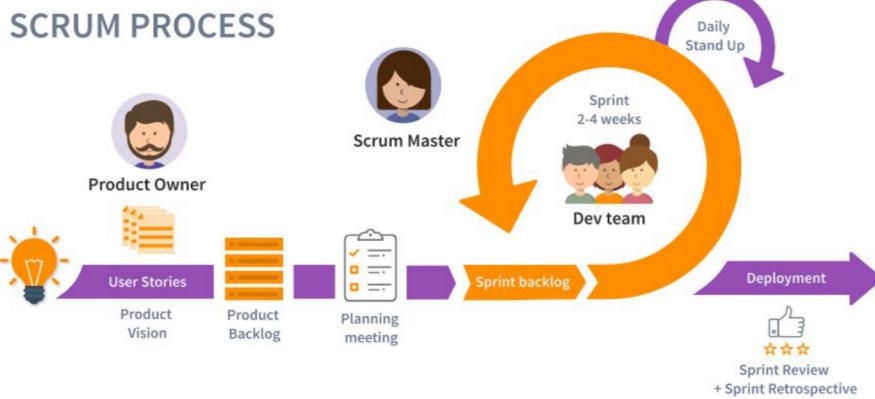
## 3 Agile Group Model

### 3.1 Scrum

#### 3.1.1 What is Scrum?

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

While the scrum I'm talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.



People often think scrum and agile are the same thing because scrum is centered around continuous improvement, which is a core principle of agile. However, scrum is a framework for getting work done, where agile is a mindset. You can't really "go agile", as it takes dedication from the whole team to change the way they think about delivering value to your customers. But you can use a framework like scrum to help you start thinking that way and to practice building agile principles into your everyday communication and work.

The scrum framework is heuristic; it's based on continuous learning and adjustment to fluctuating factors. It acknowledges that the team doesn't know everything at the start of a project and will evolve through experience. Scrum is structured to help teams naturally adapt to changing conditions and user requirements, with re-prioritization built into the process and short release cycles so your team can constantly learn and improve.

### 3.1.2 3 fundamental principles in scrum agile method

In Scrum, the goal is to define a clear and precise working framework punctuated by short iterations to facilitate the implementation of complex projects. This specific framework revolves around three fundamental principles:

- **transparency:** each team member must have access to all information regarding the product to deliver;
- **inspection:** regular evaluations are essential to readjust the project if necessary;
- **adaptation:** the implementation of new measures is necessary when inspection shows deviations from the measured results.

### 3.1.3 Three essential roles for scrum success

A scrum team needs three specific roles: product owner, scrum master, and the development team. And because scrum teams are cross-functional, the development team includes testers, designers, UX specialists, and ops engineers in addition to developers.

- The scrum product owner
- The scrum master
- The scrum development team

#### **The scrum product owner**

Product owners are the champions for their product. They are focused on understanding business, customer, and market requirements, then prioritizing the work to be done by the engineering team accordingly. Effective product owners:

- Build and manage the product backlog.
- Closely partner with the business and the team to ensure everyone understands the work items in the product backlog.
- Give the team clear guidance on which features to deliver next.
- Decide when to ship the product with a predisposition towards more frequent delivery.

The product owner is not always the product manager. Product owners focus on ensuring the development team delivers the most value to the business. Also, it's important that the product owner be an individual. No development team wants mixed guidance from multiple product owners.

#### **The scrum master**

Scrum masters are the champions for scrum within their teams. They coach teams, product owners, and the business on the scrum process, and look for ways to fine-tune their practice of it.

An effective scrum master deeply understands the work being done by the team and can help the team optimize their transparency and delivery flow. As the facilitator-in-chief, he/she schedules the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and the sprint retrospective.

#### **The scrum development team**

Scrum teams get s\* done. They are the champions for sustainable development practices. The most effective scrum teams are tight-knit, co-located, and usually five to seven members. One way to work out the team size is to use the famous ‘two pizza rule’ coined by Jeff Bezos, the CEO of Amazon (the team should be small enough to share two pizzas).

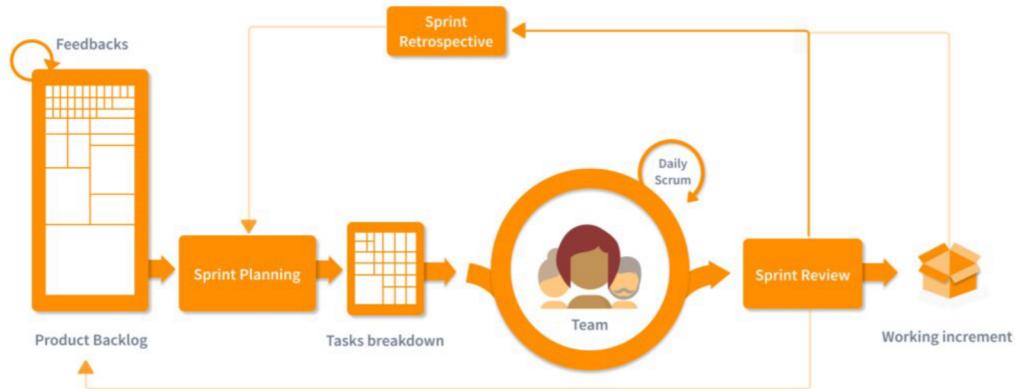
The development team is responsible for transforming the expressed needs into usable functionalities. The team can be multidisciplinary and involve several types of people: developers, software architects, functional analysts, graphic designers, ergonomists, systems engineers, etc.

## **3.2 Scrum ceremonies**

Meetings, or “ceremonies” are an important part of agile development. But they are one of many important elements, and shouldn’t be conducted in a vacuum. (It’s tempting to add some ceremonies to a waterfall project and call it “agile”, but this will get you nowhere.)

- Sprint Planning
- Daily Stand-up

- Iteration review
- Sprint Retrospective



### 3.2.1 Sprint Planning

**Attendees:** Development team, scrum master, product owner

**When:** At the beginning of a sprint.

**Duration:** Usually up to two hours per week of iteration. e.g. a two-week sprint kicks off with a four-hour planning meeting.

**Agile Framework:** Scrum. (Kanban teams also plan, of course, but they are not on a fixed iteration schedule with formal sprint planning)

**Purpose:** Sprint planning sets up the entire team for success throughout the sprint. Coming into the meeting, the product owner will have a prioritized product backlog. They discuss each item with the development team, and the group collectively estimates the effort involved. The development team will then make a sprint forecast outlining how much work the team can complete from the product backlog. That body of work then becomes the sprint backlog.

### 3.2.2 Daily stand-up

**Attendees:** Development team, scrum master, product owner

**When:** Once per day, typically in the morning.

**Duration:** No more than 15 minutes. Don't book a conference room and conduct the stand-up sitting down. Standing up helps keep the meeting short!

**Agile Framework:** Scrum and kanban.

**Purpose:** Stand-up is designed to quickly inform everyone of what's going on across the team. It's not a detailed status meeting. The tone should be light and fun, but informative. Have each team member answers the following questions:

- What did I complete yesterday?
- What will I work on today?
- Am I blocked by anything?

There's implicit accountability in reporting what work you completed yesterday in front of your peers. No one wants to be the team member who is constantly doing the same thing and not making progress.

### 3.2.3 Iteration review

#### Attendees:

Required: Development team, scrum master, product owner

Optional: Project stakeholders

**When:** At the end of a sprint or milestone.

**Duration:** Typically 60 minutes per week of iteration-e.g. a two-hour review following a two-week sprint.

**Agile Framework:** Scrum and kanban. Like planning, review for kanban teams should be aligned with team milestones rather than on a fixed cadence.

**Purpose:** Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

### 3.2.4 Retrospective

**Attendees:** Development team, scrum master, product owner

**When:** At the end of an iteration.

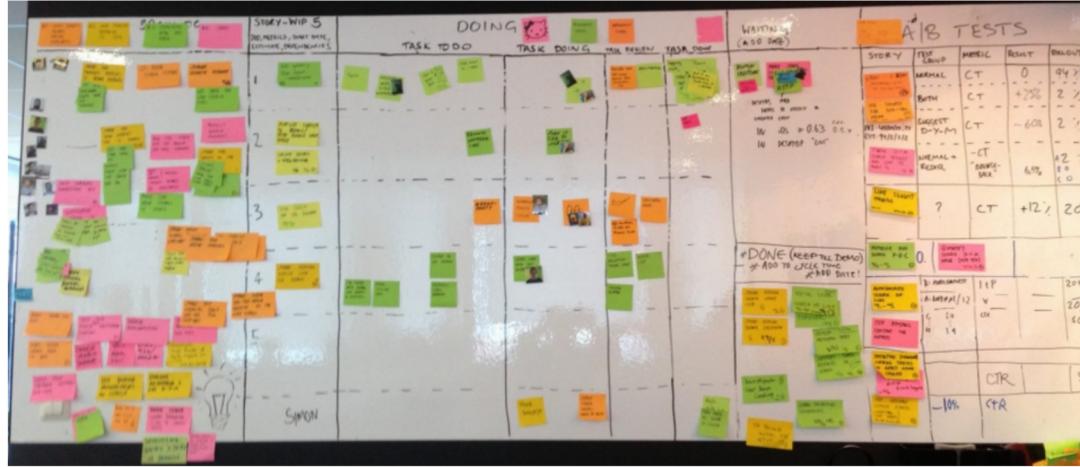
**Duration:** Typically 45 minutes per week of iteration-e.g. a 90-minute retrospective after a two-week sprint.

**Agile Framework:** Scrum and kanban. Scrum teams do sprint retrospectives based on a fixed cadence. Kanban teams can benefit from occasional retrospectives, too.

**Purpose:** Agile is about getting rapid feedback to make the product and development culture better. Retrospectives help the team understand what worked well-and what didn't.

There is a lot to cover in scrum but can't be convert in this report. You can go through the following:

- [Atlassian/Scrum](#)
- [Scrum.org/Scrum](#)



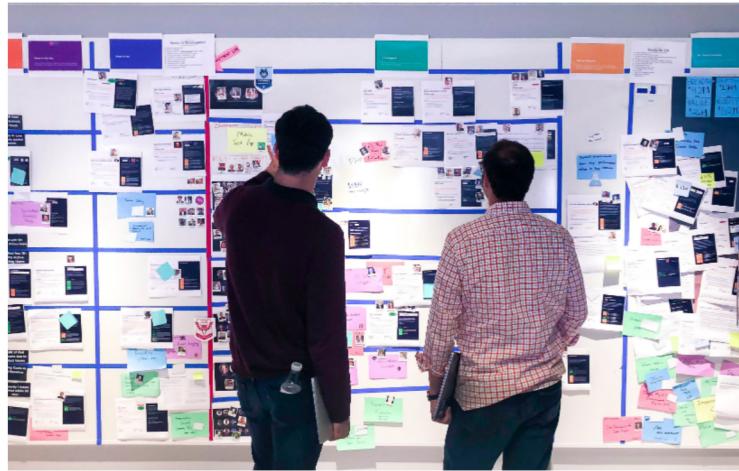
Srcum board used by Spotify Team under Spotify Model

### 3.3 Kanban

#### 3.3.1 What is kanban?

Kanban is a popular framework used to implement agile and DevOps software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.

Kanban is enormously prominent among today's agile and DevOps software teams, but the kanban methodology of work dates back more than 50 years. In the late 1940s Toyota began optimizing its engineering processes based on the same model that supermarkets were using to stock their shelves. Supermarkets stock just enough product to meet consumer demand, a practice that optimizes the flow between the supermarket and the consumer. Because inventory levels match consumption patterns, the supermarket gains significant efficiency in inventory management by decreasing the amount of excess stock it must hold at any given time. Meanwhile, the supermarket can still ensure that the given product a consumer needs is always in stock.



**Physical Kanban board used by the Atlassian Team**

Column	Task Description	Added by
To do	Change an alert when attempting to view blocked profile	Guleri24
To do	Review stories switching lag issue	Guleri24
To do	Review screen switching animations	Guleri24
Developing	Add social share buttons to all blog articles	Guleri24
Developing	Stories to Gif converter	Guleri24
Developing	Opening page delay	Guleri24
Testing	Testing log4j latest version to cope with log4j ddos vulnerability	Guleri24
Testing	Bugs reported by users	Guleri24
Done	Socket is returning old value when changing the nickname	Guleri24
Done	Broken links	Guleri24
Done	Connect an open source league statistics plugin for Github pages	Guleri24
Done	Text edits on landing	Guleri24

**Simple example of a Kanban board designed for the presentation.**

Agile software development teams today are able to leverage these same JIT principles by matching the amount of work in progress (WIP) to the team's capacity. This gives teams more flexible planning options, faster output, clearer focus, and transparency throughout the development cycle.

The work of all kanban teams revolves around a kanban board, a tool used to visualize work and optimize the flow of the work among the team. While physical boards are popular among some teams, virtual boards are a crucial feature in any agile software development tool for their traceability, easier collaboration, and accessibility from multiple locations.

## 4 Kanban Vs. Scrum

“Kanban vs. scrum” is a discussion about two different strategies for implementing an agile development or project management system. **Kanban methodologies are continuous and more fluid, whereas scrum is based on short, structured work sprints.**

Agile is a set of ideals and principles that serve as our north star. DevOps is a way to automate and integrate the processes between software development and operations teams. When it comes to implementing agile and DevOps, kanban and scrum provide different ways to do so.

**Kanban** is all about visualizing your work, limiting work in progress, and maximizing efficiency (or flow). Kanban teams focus on reducing the time a project takes (or user story) from start to finish. They do this by using a kanban board and continuously improving their flow of work.

**Scrum** teams commit to completing an increment of work, which is potentially shippable, through set intervals called sprints. Their goal is to create learning loops to quickly gather and integrate customer feedback. Scrum teams adopt specific roles, create special artifacts, and hold regular ceremonies to keep things moving forward. Scrum is best defined in The Scrum Guide.

## 5 Kanban vs. scrum: What if you can't choose?

Scrum and kanban are “agile by-the-books.” They work in a tried and true fashion that is quite frankly hard to argue against. Borrowing from another famed catch-phrase, you might say that, “No one gets fired for choosing scrum.”

But your decision doesn’t need to be so black and white. Hundreds of teams are using hybrid models influenced by both scrum and kanban.

Team-managed projects, as the name suggests, allow teams to pick and choose the agile features that make sense for them; whether that’s scrum, kanban, or a mix of both. Instead of implementing one framework on day one, team-managed projects allow you to progressively layer on more and more powerful features as you learn what works for your team (and what doesn’t).

You can confidently choose team-managed scrum or team-managed kanban knowing that both templates can evolve to suit the needs of your team.

Regardless of what you choose, stick with it for a little while. Take some work from the backlog all the way to done and then ask your team what went well and what went poorly. By trying scrum and kanban and asking these questions, you’re well on your way to agile bliss.

Parameter	Scrum	Kanban
Origin Ideology	Software Development Learn through experiences, self-organize and prioritize, and reflect on wins and losses to continuously improve.	Lean Manufacturing Use visuals to improve work-in-progress
Cadence	Regular, fixed-length sprints (i.e. two weeks)	Continuous flow
Practices	Sprint planning, sprint, daily scrum, sprint review, sprint retrospective	Visualize the flow of work, limit work-in-progress, manage flow, incorporate feedback loops
Roles	Product owner, scrum master, development team	No required roles

## 6 Agile Project Management

### 6.1 What is agile project management?

Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration.

Software teams that embrace agile project management methodologies increase their development speed, expand collaboration, and foster the ability to better respond to market trends.

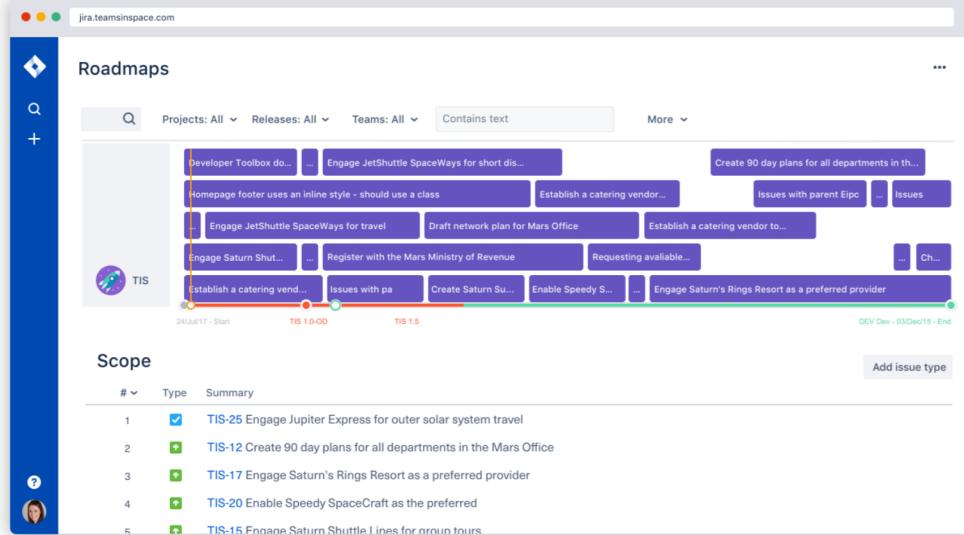
Program Manager	Project Manager
Plan strategies	Plans projects
Provides advice to stakeholders	Tracks progress of projects
Review and advise on projects	Allocates resources
Offers audits and QA	Manage risks
Mentorship to project teams	Communicate

### 6.2 Estimate, report, and plan

Whatever agile framework you choose to support your software development, you'll need a way to see your team's progress so you can plan for future work or sprints. Agile project estimating helps both scrum and kanban teams understand their capacity. Agile reports show the team's progress over time. And backlog grooming helps project managers keep the list of work current and ready for the team to tackle.

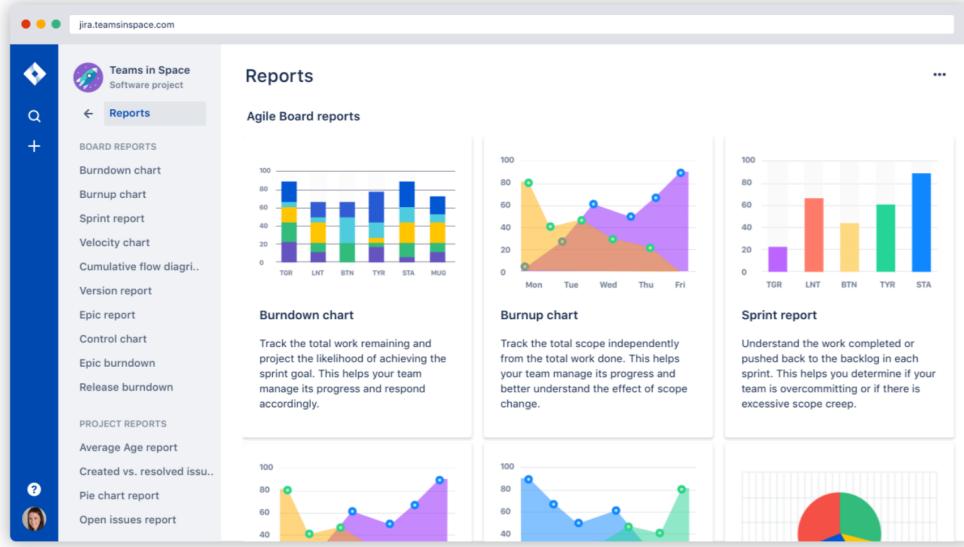
### 6.2.1 Agile project estimating

Project estimating is an extremely important aspect of both kanban and scrum project management. For kanban, many teams set their WIP limit for each state based on their previous experiences and team size. Scrum teams use project estimating to identify how much work can be done in a particular sprint. Many agile teams adopt unique estimating techniques like planning poker, ideal hours, or story points to determine a numeric value for the task at hand. This gives agile teams a point of reference to refer back to during sprint retrospectives, to see how their team performed. Jira Software can be customized to capture your teams' unique project estimations.



### 6.3 Agile reporting

Project estimations come into play at the beginning and end of each sprint. They help teams determine what they can get done at the beginning of the sprint, but also show how accurate those initial estimates were at the end. Agile reports, such as Burndown charts, show how many "story points" are completed during the sprint. Jira Software offers dozens of out-of-the-box reports with real-time, actionable insights into how your teams are performing. Having data to support your retrospectives is an invaluable way for agile teams to improve.



## 6.4 Backlog management and grooming

A product backlog is a prioritized list of work for the development team to do that comes from product roadmap and its requirements. The development team pulls work from the product backlog for each sprint.

Grooming and maintaining your backlog helps teams achieve their long-term goals by continually adding and removing items based on the team's long-term capacity and changing business objectives. Jira Software lets teams groom huge backlogs with multi-select ranking and order user stories and bugs by dragging and dropping issues. You can also filter with Jira Software's flexible search to find a particular user story or bug.

**Sprint 2** 3 issues Start sprint

<input checked="" type="checkbox"/> ATMOS-25	Engage Jupiter Express for outer sola...	<b>SPACE TRAVEL PARTNERS</b>	5
<input type="checkbox"/> ATMOS-37	Warranty - it's not possible to redirect...	<b>LARGE TEAM SUPPORT</b>	5
<input type="checkbox"/> ATMOS-9	When requesting user details, it shoul...	<b>SEE SPACEEZ PLUS</b>	5

+ Create issue

**Backlog** 19 issues Create sprint

<input type="checkbox"/> ATMOS-7	After 100,000 requests, the SpaceEZ s...	<b>SEE SPACEEZ PLUS</b>	5
<input type="checkbox"/> ATMOS-10	Session affinity - SessionBasedAnony...	<b>LARGE TEAM SUPPORT</b>	5
<input type="checkbox"/> ATMOS-19	500 Error when requesting a reservati...	<b>SEE SPACEEZ PLUS</b>	5
<input checked="" type="checkbox"/> ATMOS-41	Bad JSON data coming back from hot...	<b>SPACE TRAVEL PARTNERS</b>	5

+ Create issue

## 7 Agile Management Tools

Proprietary project management tools for agile teams

- Jira
- Github supports Agile development (Kanban style and custom)
- Trello
- Pivotal Tracker
- Inflectra and soon.

Open source project management tools for agile teams

- MyCollab
- OpenProject
- Phabricator
- Gitlab supports Agile development
- Tuleap and soon.