Complete Guide to Agile Project Management

Table of Contents

Basic Terms: Agile Project Management Glossary
What is Traditional Project Management?
What is Agile Project Management?
Difference Between Traditional Project Management & Agile Project Management
1. Agile PM is More Flexible
2. Project Complexity
3. Delivery Speed
4. Transparency Option
5. Project Risk Level
6. Checkpoints & Monitoring Progress

Stages' Difference Between Traditional PM & Agile PM (Lifecycles)
Stages of Traditional Project Management
Stages of Agile Project Management
Agile Methodologies
Related Article: Best Project Management Courses Online (Free & Paid)
When You Shouldn't Use the Agile Project Management Method
1. The final outcome of your project is quite stable and well-understood
2. Your project must produce a repeatable deliverable
3. Perhaps, your customer doesn't want Agile
4. Your company cannot support Agile

Key Differences Between a Project Manager & Agile Project Manager (or Scrum Master)
The 12 Agile Principles
How Agile Project Management Works
Agile Project Management vs. Waterfall
Agile Project Management vs. Scrum
Are Agile & Scrum Right for You?
How to Make the Agile Process Work Effectively?
Best Agile Project Management Software
Best Agile Project Management Certification Courses Online
Conclusion

For a couple of weeks, we've been preparing this **guide to agile project management**. We've gathered all the required information on agile project management and **how to become a certified project manager**.

Today, this kind of profession is one of the **most required ones** in the world.

Before delving into more details of **Agile Project Management (APM)**, let's see the glossary and the **most common terms** you'll be using during your APM work.

Basic Terms: Agile Project Management Glossary

Agile – an **iterative project management approach** to manage projects in the **IT-sphere**.

Agile Development – a term used basically for **iterative software development methodologies** such as **Scrum**, **eXtreme Programming (XP)**, and others.

Agile Manifesto – a brief document that includes **4 values** and **12 principles** for successful agile software development.

Backlog – **prioritized work** that still needs to be **finished**.

Burndown Chart – professionals use this to **monitor the progress** of the project. It shows the **remained work that should still be done** versus the **total time left**.

Cadence – the **time** (number of days, weeks, months) in a **Sprint** or the **team's development cycle length**.

Ceremonies – meetings, often a daily planning meeting, that identify what has been done, what is to be done, and the barriers to success.

Disciplined Agile Delivery (or DAD) – **process-decision guidelines** providing **context-specific guidance** based on your business needs trying to **increase the quality** of your product.

Daily Scrum – a **daily meeting** with the team on **planning**, **monitoring**, and **discussing** the tasks.

DevOps (or Development Operations) – bridges the gap between agile teams and operational delivery to production.

Dynamic Systems Development Method (or DSDM) - this is an **agile project delivery framework** used in the **software development processes**.

eXtreme Programming (XP) – another **agile development methodology** that's used in the **software development process**. It helps to produce **higher quality software** as well as a **higher quality of life** for the software development team members.

Kanban – a **visual system** for **managing work processes**.

Kanban Board – one of the **key components** of the Kanban methodology helping to monitor and reveal the **status**, **progress**, and **issues** related to the work process **visually**.

Lean – a **method** in software development that helps teams to **deliver projects faster**.

LeSS (large-scale Scrum) – agile development method that project managers apply to not just small teams but also to an agile team consisting of **fifteen**, **hundred** or even **thousands** of members.

Rapid Application Development (RAD) – agile development method that **highlights** the use of **software** and **user feedback** to meet **business requirements.**

Requirements – in the case of Agile projects also known as "**user stories**" is the document mentioning all the capabilities you want in a planned system.

Scaled Agile Framework Enterprise (or SAFe) – is another **Agile methodology** used during the software development process.

Scaled Agile – is a toolbox **guiding** companies in scaling **lean** and **agile practices** (agile scaled up to **large projects** or **programs**).

Scrum – one of the **worldwide** known methodologies used in software development helping teams to **review** and **see** the progress of a **single development phase** (also called **Sprint**).

(The) Scrum of Scrums – a technique used for **multiple** teams working on the **same** product/project.

Scrum master – this is (hopefully) you - the person who's **monitoring** the **whole development process** and assures everyone's **delivering** their work **on time**.

Sprint – a **short** development phase within a **larger** project. For example, it's when teams define a **short duration of a Sprint** (in a large project) up to **5 days** to **4 weeks timebox**.

Sprint retrospective – is a **good opportunity** for the team to **review** their **own** work and create a **reshaped plan** for improvements that have them discussed/implemented during the **next Sprint**.

Velocity – some **measure** of the **work done** the working team can **handle** during **one Sprint**.

Waterfall – a **structured** project management approach helping **software development teams** to organize their work.

What is Traditional Project Management?

The traditional Project Management (meaning the **waterfall** approach) is a universal approach that includes a **set of unique techniques** used for **planning**, **estimating**, and **controlling** activities (in **sequence**).

In the case of traditional PM, **every project** follows the **same life cycle**. The standard life cycle includes some stages such as **feasibility, planning, designing, building, testing, production**, and **support**.

The entire project is planned **beforehand** (with **fixed time** and **costs**) with **no changing possibility**. So once the project development starts one should be **extremely careful** not to make any mistakes during its implementation.

What is Agile Project Management?

In short, **Agile** is a **project management methodology**.

The methodology includes sprints that are short development cycles focusing on **continuous improvement** in the development of a product, service, or process.

Today, agile project management is **rapidly developing** and it's one of the top paid jobs in the world. Moreover, **Agile projects** are **28% more successful** than **traditional projects**.

Also, according to **various sources**, the **salary** for agile project managers varies from **$66,137** to **$154,063 per year**.

So if you're still fluctuating whether to **shift your career** and **become** an agile project manager, it's **worth** it!

To delve **even more** into Agile PM spend some minutes and **explore** what the Agile methodology is in the video below from **Linkedin**:

https://www.youtube.com/watch?v=MJR-EgHTA4E

Difference Between Traditional Project Management & Agile Project Management?

Many project managers and experienced developers use project management skills to **handle** their projects on time.

However, many of them **prefer to use the Agile methodology** instead of **Traditional PM** because of **some reason**.

1. Agile PM is More Flexible

In the case of **traditional project management**, one should define **every single process** right at the start of each project.

It **cannot** deal with any big change (or especially feedbacks) **requiring** some change in the process. In almost all the projects **final delivery time** and the **costs along with budget** are **fixed**. So one can change it very **rarely**.

When doing **Agile project management**, you **do have** a huge space for **feedback** from the clients' side. Here the process is very flexible.

Moreover, clients **follow** the process and they can provide feedback at **any stage of the project**. In the end, you get a better product **without a headache**.

The reason is that the Agile PM methodology **isn't linear** (compared to the Traditional PM) and **doesn't** require a **top-down approach**. So you shouldn't suffer from **last-minute changes** that can disrupt the project **time frame**.

2. Project Complexity

Overall, people use **traditional** project management methodology basically for not big and **less complex projects**.

As I've mentioned before, it **doesn't like** unexpected changes. So here the **Agile** could be your best option to handle **big** and **complex** projects.

But why **prefer** Agile PM?

Because complex projects include **multiple interconnected phases** or **just one stage**. And here comes Agile PM to help you **finish** the bigger projects.

3. Delivery Speed

The speed **does matter**!

When it comes to finishing a project **on time**, **anything** can happen.

The Agile PM **ensures** that your team is **safely** and **constantly** delivering **good-working products**. This means that any client can **get access** to the product **anytime** without waiting for the **final working version**.

For example, they can wait for **6 months, 1 year, 2 years**, or even more get the **final version** of the product. Agile PM makes the process **much faster** seeing the changes in much **shorter intervals** (very often every **2 to 4 weeks**).

4. Transparency Option

The Agile PM methodology is quite **transparent**. The clients (or other people) can get access to the **project's every single phase** (initiation, planning, review, testing).

The **agile methodology** allows a project's team members to **view the progress** from the **very start** to the **end**.

5. Project Risk Level

During the Agile PM, the **risk** of a **project failing** is **minimal**.

A good project manager **divides** a large project into some **iterations**. Thus, they **decrease** the risk level of **failing** it.

In the case of traditional PM, you **realize** the real risk of the project when almost everything is done or when you get the project **finalized**.

Every project **manager**, **project owner**, or **project financing organization** wishes to use **less time, effort**, and **money** but get **good final results**. So this is their chance to get what they want and **avoid** any problems.

5. Checkpoints & Monitoring Progress

The traditional PM method advises spending much time **planning the project** at the analysis and design stage of the project. In this case, the focus is on **smoothing the processes** and **not the product itself**.

When the **planning** is over, the team members should follow the steps in that with **minimal guidance**.

One **assesses the effectiveness** of the project when it's already **completed**. No regular meetings unless the PM receives any **guidelines**.

The **Agile methodology** encourages team members to meet and have checkpoints at **regular intervals**. When meeting every **2 or 4 weeks**, it's quite easier to check all the **work done**, its **effectiveness**, and discuss any **issues** on the way to the finish line.

If the work process requires **more frequent meetings**, you can use one of Agile other methodologies - **Scrum**. In this case, team members or numerous teams meet on a **daily basis** to **discuss** the **work done** after the last meeting, to **make changes**, or **tackle the obstacles**.

To **visually show** you the difference between traditional PM and Agile PM, we have **researched** and **provided** you the best infographics on the topic.

One of the **best visual overall pictures** I could found, you can see below (infographic from Kanbanize.com):

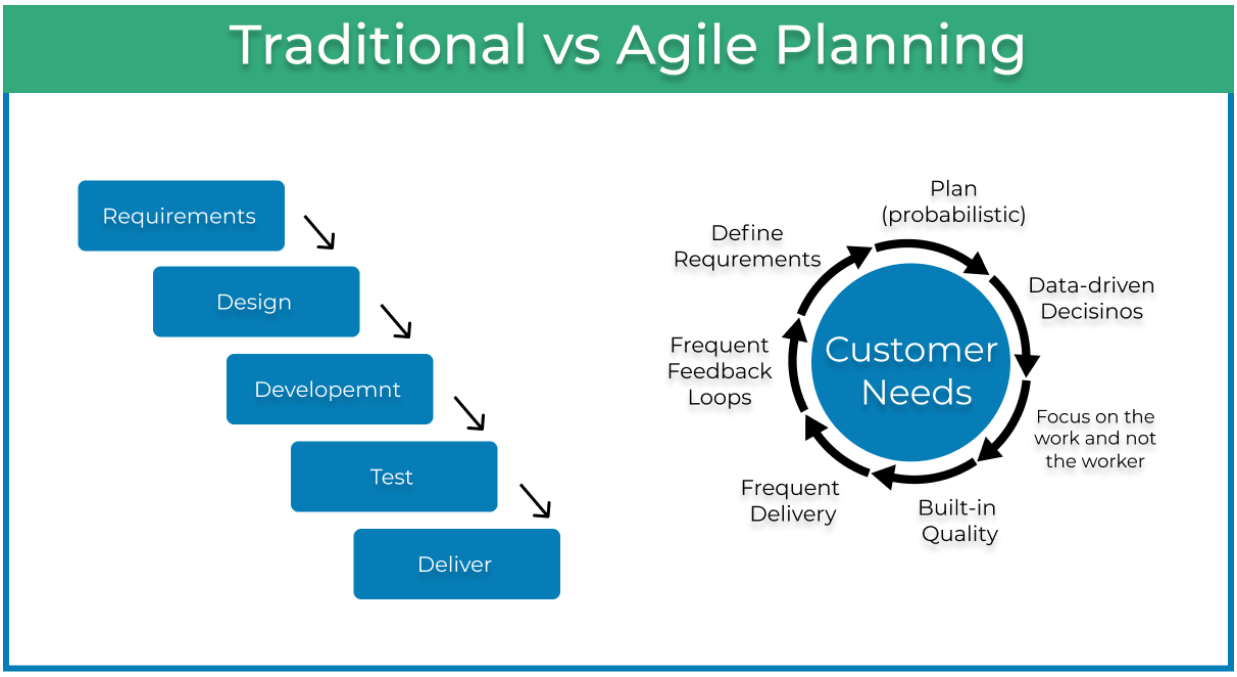


Image Source: Kanbanize.com

Here's another good **infographic** from Kissflow.com:

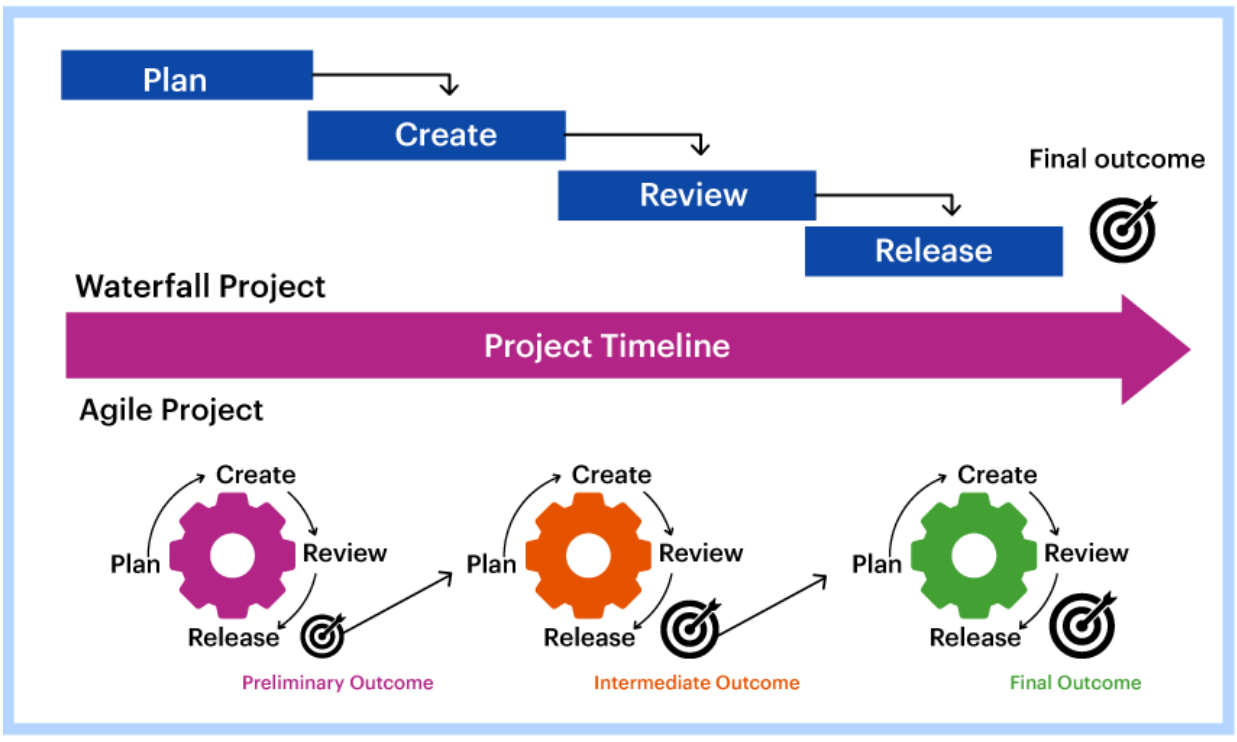


Image Source: Kissflow.com

And another infographic to fully show you the **visual comparison** of Agile project management and traditional project management (infographic from Knowledgehut.com):

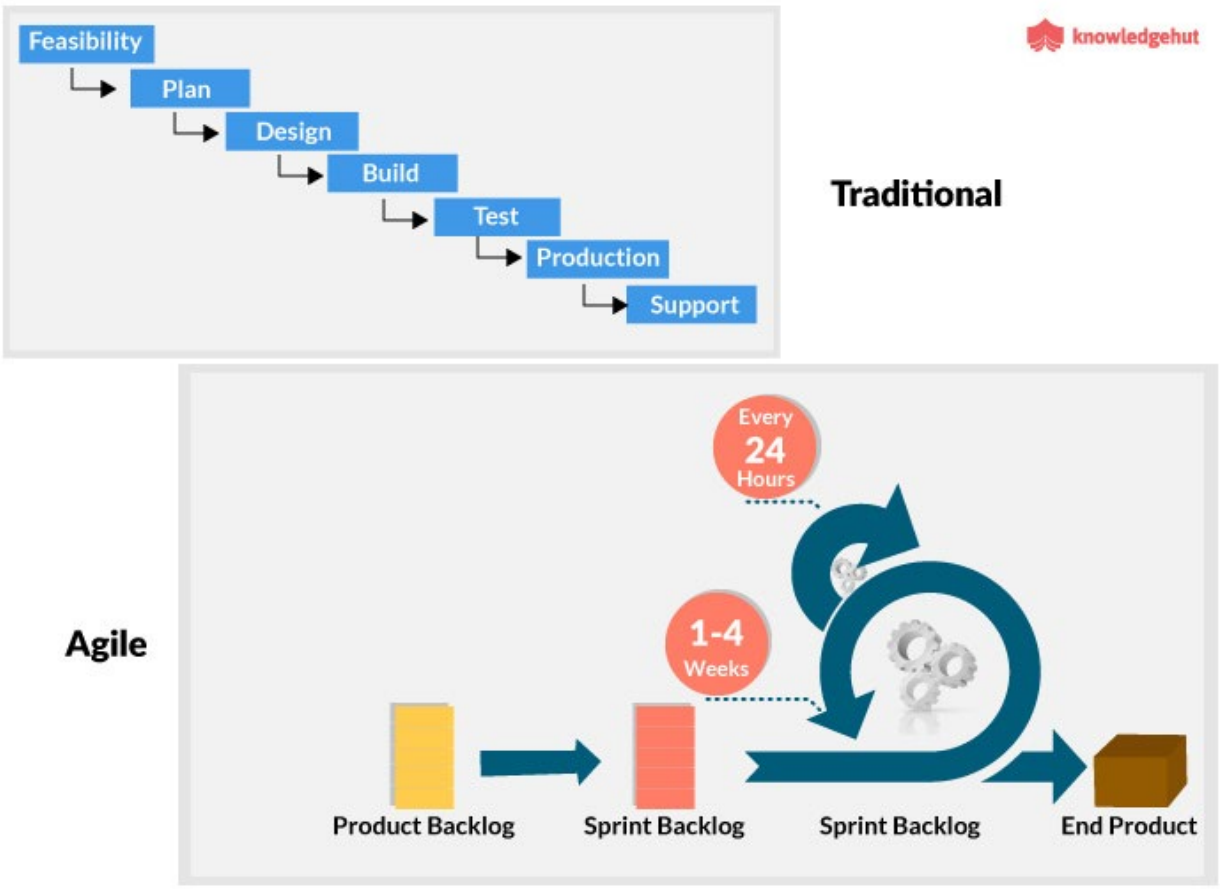


Image Source: Knowledgehut.com

Also, you should **pay attention** to the **main characteristics** of the two approaches.

Such characteristics include but not limited to a **linear** or **iterative process**, **project's scale** (small, medium or large), **client's involvement** in the work process, and others.

The **full list of characteristics** you can see in this **infographic** below:

knowledgehut

Characteristics	Traditional Project Management	Agile Project Management
Organizational structure	Linear	Iterative
Project scale	Large-scale	Small and medium scale
Development model	Life cycle model	Evolutionary delivery model
User requirements	Clearly defined before coding or implementation	Interactive input
Client involvement	Low	High
Restart Cost	High	Low
Development process	Life cycle model	Evolutionary delivery model
Development model	Fixed	Easily changeable
Testing	Once coding is done	Every iteration
Architecture	Creates current and predictable requirements	Creates current requirements
Requirements	Standard and known in advance	Emergent with rapid changes

Infographic source: Knowledgehut.com

Stages' Difference Between Traditional PM & Agile PM (Lifecycles)

Stages of Traditional Project Management

Basically, there are **5 project management process groups**:

- **Initiating**: This is the **very first stage** of the Traditional PM - it should **define the project**.
- **Planning**: Next, the planning phase includes developing the **full roadmap** for the **project's every member** to follow.
- **Executing & Monitoring**: In this **executing** phase, you're **starting** the project. When having the project **started**, the PM creates the **working team** and **deliverables**. Afterward, they **monitor** and **measure** project **performance** to ensure **how it's going**.
- **Closing**: The **completion** of the project. After finishing the project, it's being **transferred** to another team who will maintain it.

Stages of Agile Project Management

In general, there are **6 basic stages** of the Agile Development Life Cycle:

1. **Scope out and prioritize projects**
2. **Diagram requirements for the initial sprint**
3. **Construction/iteration**
4. **Release the iteration into production**
5. **Production and ongoing support for the software release**
6. **Retirement**

We'll not make it even **longer** and will **definitely advise** you to read the article from **Lucidchart.com** on "**The stages of the Agile software development life cycle.**"

Agile Methodologies

There are a **variety of Agile methodologies** in the world today.

We've mentioned some of the **most frequently used** by the Agile project managers. Those methodologies **include but not limited to** the following:

- **Kanban**
- **Scrum**
- **Disciplined Agile Delivery (DAD)**
- **Agile Unified Process (AUP)**
- **Scrumban**
- **Crystal**
- **Crystal Clear methods**
- **Adaptive Software Development**
- **Feature Driven Development**
- **Agile Modeling**
- **Lean Software Development**, etc.

All in all, the **key role** of each Agile method is to **avoid** any obstacles (by adapting to change, like **Chameleons**) and deliver working software (or a good-working final product/project) as **quickly** as possible.

However, even though the **goal** of each **mentioned** methodologies is the same, each team's **process flow may vary** depending on the **project's** or **product's specifications**.

Related Article: ***Best Project Management Courses Online (Free & Paid)***

When You Shouldn't Use the Agile Project Management Method

Although the Agile has huge **advantages** and **benefits**, it's **not** considered for **every project/organization**.

But **how to identify** when the Agile PM **isn't the best methodology** to use?

To make your job easier, I have gathered **4 cases** when the Agile development model **isn't the best option** for your projects.

1. The final outcome of your project is quite stable and well-understood

As you may already know, people use Agile to **reduce** the **cost** of the occurred **change** and **uncertainty**.

In contrast, if there are **very little uncertainty** and a **low possibility of change**, then you can use another methodology rather than Agile.

For instance, if you work in a **patent industry** with **heavy regulations** you **don't need** iterative planning and multiple drafts.

2. Your project must produce a repeatable deliverable

From the very **beginning** to the **end**, Agile's core function is to **create a unique product**, **service**, or **result**."

But imagine a situation when a customer **asks** you to construct **10 identical houses**. In this case, your final product/result will be **10 unique houses** and **not 5 identical ones**.

So one of the **disadvantages** of Agile is that it's **not considered** for **reproducibility**.

3. Perhaps, your customer doesn't want Agile

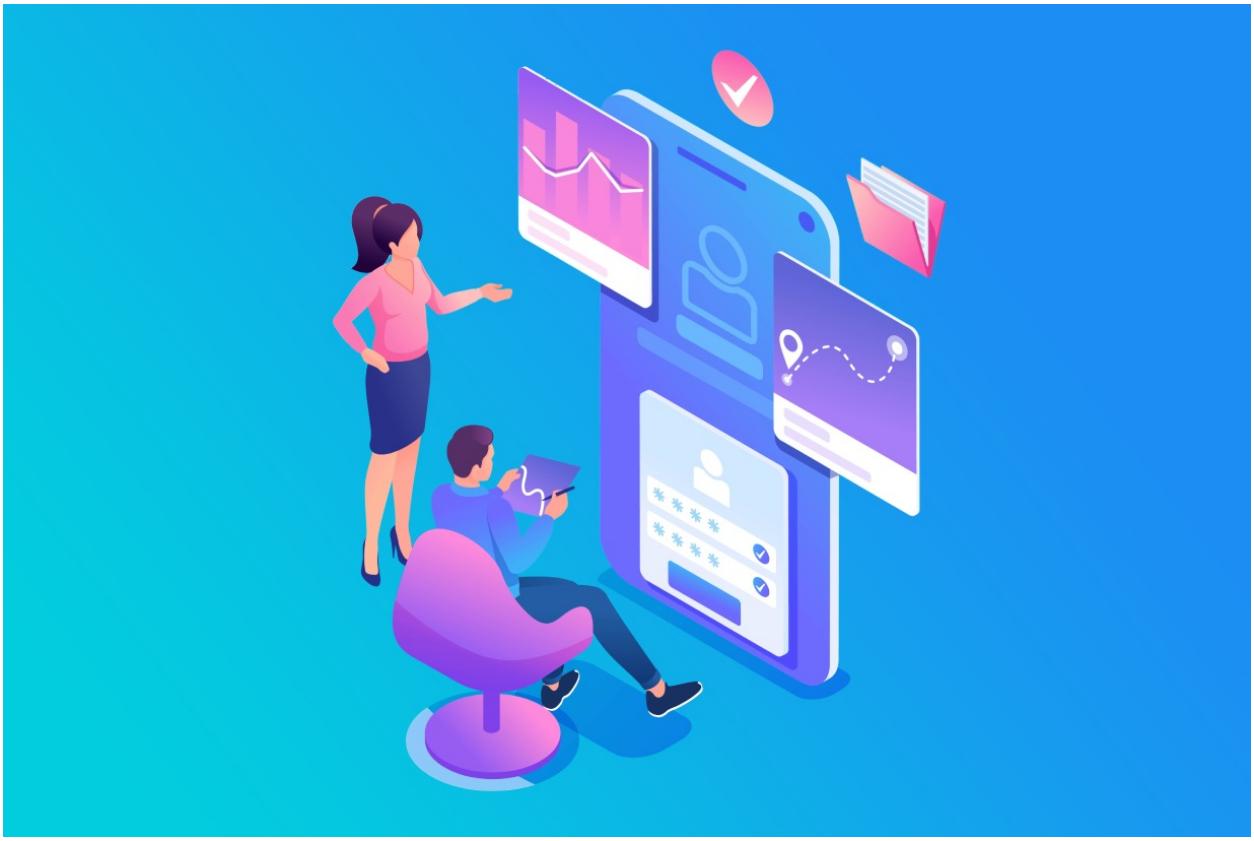
An Agile project **requires** being constantly in touch with your **end customer**.

But there are some cases when customers **don't have** the **time**, **will,** or **ability** and don't desire to **dedicate** to a project. So if the project contains a **low value** or **low risk**, they may **prefer** a more traditional methodology instead of using the Agile one.

4. Your company cannot support Agile

If your **company**, **product**, or **project team** isn't ready for Agile, trying to adopt it can contain some risk to your **project**. So be sure they have all the resources to cross the **finish line**.

Key Differences Between a Project Manager & Agile Project Manager (or Scrum Master)



The **Project Manager** manages a project on a day-to-day basis and is the **only person** with monitoring on the changes of the project.

Also, if there are **no managers** within the company they implement **Project Support** and **Team Management**.

On the other hand, the **main responsibility** of the **Scrum Master** is to monitor and manage the **Scrum Framework** (according to the Scrum Guide).

For the detailed information, you can read this article from **Medium** on **Scrum Master vs Project Manager - An overview of the differences**.

The 12 Agile Principles

Overall, **Manifesto** for **Agile Software Development** highlights **12 Agile principles** for all the projects to **follow**.

We'll keep **short** and just mention the 12 Agile Principles:

1. The top priority is **customer satisfaction**: one should achieve it via **fast** and **continuous delivery**.
2. Project managers can implement changes at **any stage** of the process (for customer's competitive advantage).
3. Project managers **deliver** the product or service with a higher **frequency**.
4. Both **stakeholders** and **developers** (or their team members) **cooperate** with each other closely on a daily basis (not to miss any issues).
5. It's more efficient to **build projects** around **motivated people**. The role of PM should be to provide all the **necessary environment** and **support** they need to do the job in the **best possible way**.
6. All in all, **face-to-face meetings** stay the most effective way for **project success**.
7. **Working software** is the **primary measure of progress** - This means that delivering a good-working and functional software to the client is the **most important factor** that measures progress.
8. **Constant development** is being done via agile processes. Next, the development teams and stakeholders are having the opportunity to maintain an **ongoing pace**.
9. Agile PMs should make the **Agility** stronger by **continuously** focusing on **technical excellence** as well as **proper design**.
10. **Simplicity** (the strategy of revealing and maximizing the amount of missed work) is an **essential element**.
11. It's all about self-organizing teams: they develop the best practices along with **architectures** and **designs** meeting program **requirements**.
12. **Regular intervals** help teams to **work** and **become** more effectively (along with fine-tuning behaviors).

How Agile Project Management Works

First and foremost, Agile Project Management **guides** the teams to **continuously evaluate** time and cost throughout the **whole work process**.

To make the process **simpler**, Agile PMs use **velocity**, **burndown**, and **burnup charts** to assess the ongoing work.

In contrast, Traditional PMs use **Gantt charts** and **project milestones** to track and measure the **implementation** of their work.

As mentioned before, **Scrum** is a **framework** for Agile PMs including **iterations of sprints**.

Generally, there are 4 ceremonies that **bring detailed structure** to each sprint.

Sprint Planning	Sprint Demo	Daily Standup	Retrospective
Team Planning Meeting: this determines what tasks to implement in the coming sprint.	*Sharing Meeting:* here the team shows what they've already done in that sprint.	Also known as a "Stand-up", a 15-minute small meeting to get together and coordinate the team's activities.	A review of what the software development team has implemented at the end of a project/process to discuss success, failure, or possible improvements for future work.

To fully reveal **how Agile PM works**, we've found this video below (from **ProjectManager.com**) that will help you **visually** get the **full picture**:

https://www.youtube.com/watch?v=DvBKevrItcc

Agile Project Management vs. Waterfall

Why in many ways the **Agile PM** is a **better option** than the **Waterfall methodology**?

The thing is that the **Waterfall Model** follows **strict rules** including a **strict sequential approach** to projects. The process starts with **gathering** all the requirements needed for the project **before** the work begins.

That's the reason why **in 2001 17 software developers** published the **Agile Manifesto** where they outlined **12 principles of Agile Software Development** (mentioned a little above).

In the Agile Manifesto, they made it **possible** to **make changes** in any project/program development phase.

Overall, the **waterfall method** has such a picture:

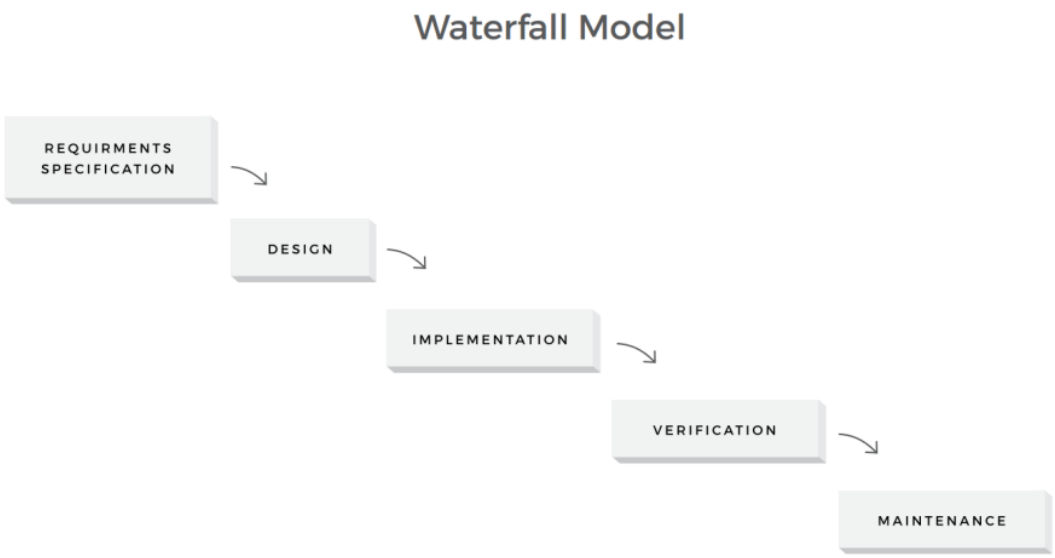


Image Source: Altexsoft.com

On the other hand, **Agile methodology** has another picture:

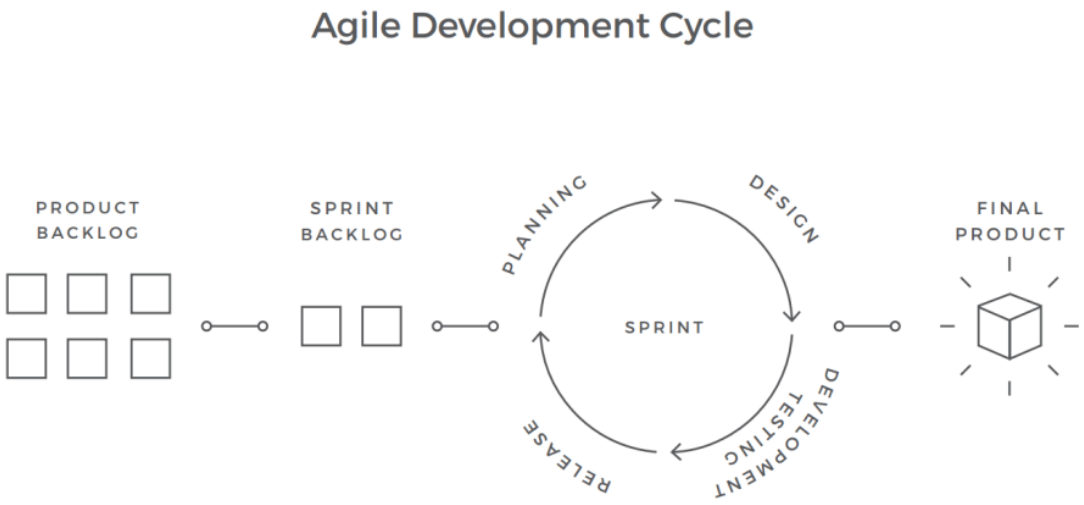


Image Source: Altexsoft.com

You can explore the **full differences** in this **table** below from **Guru99.com**:

Agile	Waterfall
It separates the project development lifecycle into sprints.	The software development process is divided into distinct phases.
It follows an incremental approach	The waterfall methodology is a sequential design process.
Agile methodology is famous for its flexibility.	The waterfall is a structured software development methodology so most times it can be quite rigid.
Agile can be considered as a collection of many different projects.	Software development will be completed as one single project.
Agile is quite a flexible method that allows changes to be made in the project development requirements even if the initial planning has been completed.	There is no scope of changing the requirements once the project development starts.
Agile methodology, follow an iterative development approach because of this planning, development, prototyping, and other software development phases that may appear more than once.	All the project development phases like designing, development, testing, etc. are completed once in the Waterfall model.

The test plan is reviewed after each sprint.	The test plan is rarely discussed during the test phase.
Agile development is a process in which the requirements are expected to change and evolve.	The method is ideal for projects which have definite requirements and changes not at all expected.
In Agile methodology, testing is performed concurrently with software development.	In this methodology, the "Testing" phase comes after the "Build" phase.
Agile introduces a product mindset where the software product satisfies the needs of its end customers and changes itself as per the customer's demands.	This model shows a project mindset and places its focus completely on accomplishing the project.
The agile methodology works exceptionally well with Time & Materials or non-fixed funding. It may increase stress in fixed-price scenarios.	Reduces risk in the firm-fixed-price contracts by getting risk agreement at the beginning of the process.
Prefers small but dedicated teams with a high degree of coordination and synchronization.	Team coordination/synchronization is very limited.
Products owner with team prepares requirements just about every day during a project.	Business analysis prepares requirements before the beginning of the project.
The test team can take part in the requirements change without problems.	It is difficult for the test to initiate any change in requirements.
Description of project details can be altered anytime during the SDLC process.	Detail description needs to implement the waterfall software development approach.
The Agile Team members are interchangeable, as a result, they work faster. There is also no need for project managers because the projects are managed by the entire team.	In the waterfall method, the process is always straightforward so, the project manager plays an essential role during every stage of SDLC.

Agile Project Management vs. Scrum

Things can be a bit confusing as you'll see and use the "**Scrum**" and "**Agile**" thousands of times when you first enter the PM world.

However, they are **not the same**!

Here's an example of a **detailed Scrum introduction**:

https://www.youtube.com/watch?v=K7YMEFjh724

Scrum is only a **framework** that PMs use to implement **Agile development**.

On the other hand, **Agile** is a **whole process** for getting the **desired results**. And for the final results, the Agile is using numerous **simple methodologies** including **Scrum**.

Generally, there are **3 scrum team roles** in the PM industry:

1. **Product Owner** who manages the product vision and make sure a good ROI (return on investment).
2. **Scrum Master** - this is the person who manages the scrum process.
3. **Self-Managing Team** - the project/product team members who implement the daily tasks required for each sprint.

As for the scrum meetings, there are **5 types of scrum meetings**:

1. **Sprint Planning Meeting** - this type of meeting is held at the beginning of each sprint.
2. **Daily Scrum & Sprint Execution Meeting** - daily small meetings (for example, 10-15 minutes).
3. **Sprint Review Meeting** - held after the last sprint meeting to provide a working product demonstration to the product owner.
4. **Sprint Retrospective Meeting** – held at the end of a sprint. The purpose of this type of meeting is to evaluate the working process and identify if there is space for improvement.
5. **Backlog Refinement Meeting** – held before the next Sprint Planning Meeting to ensure the backlog is ready for the next sprint.

After the current sprint, the **next** one begins. The meetings **help** scrum teams to **make decisions** by analyzing the current situation.

Are Agile & Scrum Right for You?

Generally, the Agile method is **more relevant** for the cases where customers and project stakeholders should provide **input, functional portions**, or **quick changes**.

Why consider it?

Because flexibility is at the **core** of **effective collaboration**.

When you have projects with **uncertain requirements**, **technological projects**, or especially projects where most probably would be numerous **changes**, **Scrum** is right for you.

In Scrum, a set of **defined processes** can help you manage even the **most complex projects** (we've talked about the **benefits** of using Agile a little above).

If your project **requires knowledge generation** and getting **new products** with small self-organized teams, Scrum is the **best way** to get things done **faster**.

How to Make the Agile Process Work Effectively?

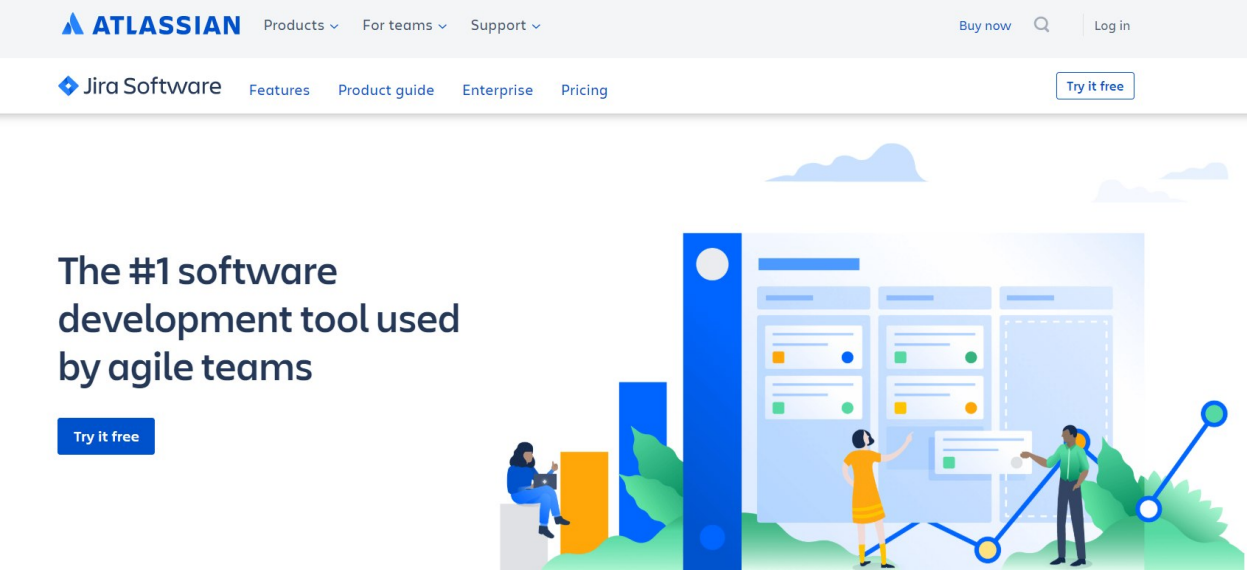
When working with the Agile methodology, **not everything** can go **smoothly**. You should have some **strategy** to make things done in a more efficient way.

Here are some **activities** that can help in adoption the **Agile workflow**:

- **Daily Meetings** - One of the **top tactics** Agile PMs use today to keep each iteration **moving forward**.
- **Live Demonstrations** - Deliver **live demonstrations** to the team and show each iteration's final product **progress**.
- **Share Feedback** - Receive or ask **feedback** from customers and stakeholders. Next, **share** the received feedback with the working team **before** the next iteration.
- **Remain Agile** - **Feedback** is the **key**. Make changes to your process based on feedback to ensure each iteration improves the last

Numerous project managers use this **strategy** to make the Agile process work even more **effectively**. So why not try it?

Best Agile Project Management Software



Companies using numerous **software programs** to **handle** project management **tasks**.

Here are some of the **Agile software solutions** for you:

- **Atlassian Jira**: This is probably the **most famous software tool** use by Agile project managers. It supports **Scrum**, **Kanban**, and **mixed methodologies** as well.
- **VersionOne**: This PM solution is ready to **support** the **Scaled Agile Framework** at **all levels**.
- **Trello**: Trello allows its users to work **more collaboratively** and get more **done**. Trello is an easy PM online tool to help you keep **tracking** every single task in **one place**.
- **Monday.com**: This is a project management tool that powers teams to implement **processes**, **workflows**, and **projects** in **one digital workspace**.
- **Nifty**: The PM tool is a **collaboration hub** providing people the opportunity to manage **tasks**, **timelines**, and **workloads** clear to your team members and clients.

Also, it's worth to mention that **choosing** an Agile project management tool **depends** on your organization's accepted **policy**, **budget**, and **practice**.

For example, if they use **Jira** for a **long time**, you will have to **accept** that and **move on** with that option even if you like another tool **more**.

For more tools, you can explore **these Agile tools for software teams**.

Best Agile Project Management Certification Courses Online

The **Agile industry** is rapidly growing along with the **demand** for **Scrum Masters** or **Agile professionals**.

Here's a good article on "**What is a Scrum Master**" that includes the **definition** of a scrum master, their **responsibilities**, and even more.

In addition, providing you with **five of the best agile-focused certifications** to get you **certified** and smash:

- Agile Certification Institute
- Scrum Alliance
- Strategyex Certificate (Associate or Master's) in agile
- International Consortium for Agile (ICAgile)
- Scaled Agile Academy

Conclusion

A shift in career is a huge **responsibility** and **opportunity** at the same time.

One should meticulously **explore every detail** and excel in this **guide to agile project management** to be ready for the **issues** and **trials** of becoming a good Agile Project Manager.

Hopefully, the provided information will **lead** you through the **tough process** of starting a new career in the Agile world.

I will finish my article with the words of **Paulo Coelho**:

"One day you will wake up and there won't be any more time to do the things you've always wanted.
Do it now!"

Paulo Coelho

Good Luck! ☺