

## Are incremental and iterative the same phenomenon or not? <sup>1</sup>

Henny Portman

During training classes I often notice that students mix the words iterative and incremental together. After reading this article I hope you understand the relationship between incremental and iterative development. I will start with a comparison of a waterfall and an agile approach by delivering a payment app<sup>2</sup>. In the second part of this article, I position waterfall and agile in an incremental versus iterative matrix and shows what happens in the other two quadrants too. As a last step I elaborate on the minimum viable product (MVP) and the minimum marketable product (MMP) and shows where these fit in the different approaches and a story map<sup>3</sup>.

### The development of a payment app

#### Waterfall approach

If this app is developed according to a traditional waterfall approach, the following steps could be observed (see figure 1).

It all started with a project sponsor from the marketing department who was able to free up the necessary funds for this app. It was his assumption that the app will improve the retention figures and the inflow of new clients will grow. He visualized three high level function groups.

At a certain moment this project gets the approval to commence. A project manager is assigned, and a project team built. After many discussions and requirement gathering workshops there was an agreement to deliver a payment app with 250 features. All these features are recorded in an extensive and very detailed requirements document and signed by the project sponsor and customer representative (and some others).

In the next step the project team translates the requirements into a design for the app. The architect checks the design towards the design principles. He checks if all needed data attributes are available in the backend system too.

---

<sup>1</sup> How to cite this article: Portman, H. (2020). Are incremental and iterative the same phenomenon or not? *PM World Journal*, Vol. IX, Issue IV, April.

<sup>2</sup> See YouTube for my mini webinar on this topic (Waterfall vs agile delivery): <https://youtu.be/Ky4BUPEFq7Y>

<sup>3</sup> See YouTube for my mini webinar on this topic (Incremental vs iterative): <https://youtu.be/cWADcFyMX4w>

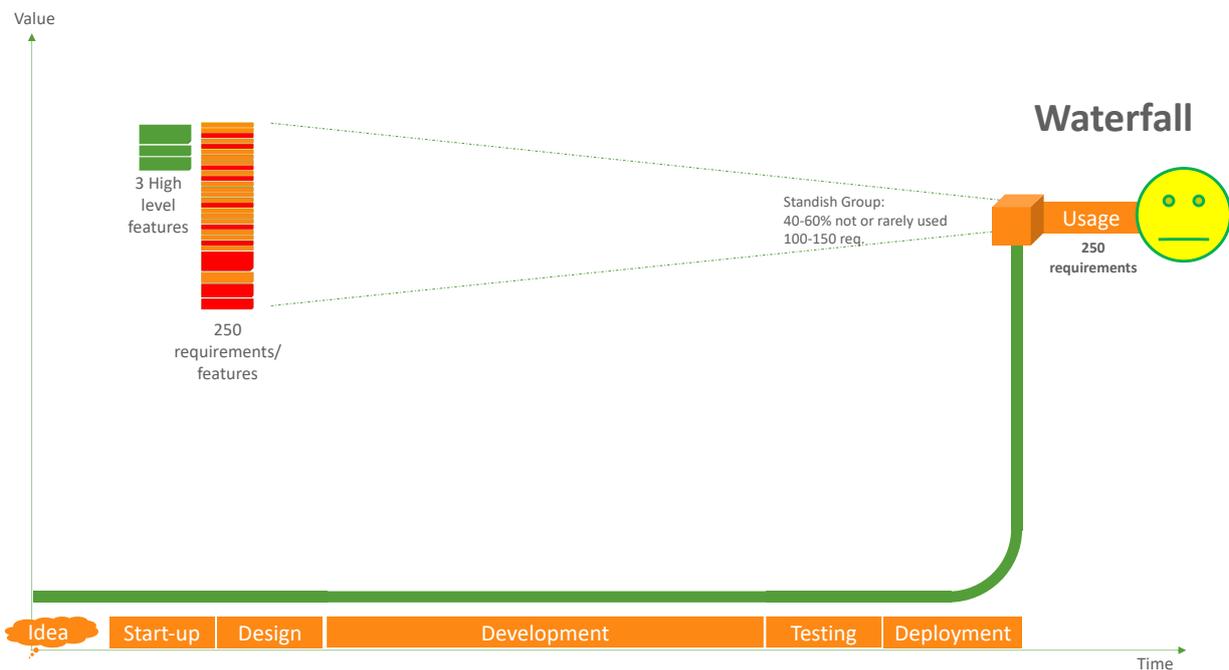


Figure 1: waterfall delivery of an app

We are now two months underway and the customer hasn't seen anything working yet, only some progress reports. Probably some sort of 'melon' reporting<sup>4</sup> so they have no clue if the project is on track or not. It takes six months to develop the app and when that's done the customer representative is asked to deliver some people who can help with a user acceptance test. During the test it becomes clear that several features are not working. The project team doesn't understand why. It's exactly what was described in the requirements document. Lots of discussions, rework, delays and customers who aren't happy with the results. If we look at the final result, we can also notice that many of the developed requirements are not or rarely used<sup>5</sup> by the customer. It could even be worse. Suppose the development of the app took 1.5 year and another bank delivers a payment app when you are halfway. What would you do at that moment? Would you still have a viable business case to continue and finish your own app?

If you look at figure 1 it becomes clear that in case of a waterfall approach the scope and underlying quality criteria are fixed with a single delivery. All steps are performed once for the entire project and management control is focusing on cost and time. Value delivery to the customer only takes place after the deployment of the complete app.

### Agile approach

If we develop the app in an agile way, we see the following pattern. The development team stated that they are able to deliver the first two features prioritized by the product owner (balance

<sup>4</sup> 'Melon' or better 'watermelon' reporting is a progress report with green indicator(s). In reality the green indicator is red inside.

<sup>5</sup> Standish Group research shows that 60% of the developed requirements are not or rarely used (<https://www.standishgroup.com>).

information and submit feedback) in the first iteration. The project team delivers every three weeks (sprint or timebox) an increment of the product. After the first deliveries or increments we see a customer who has confidence that the project will deliver. He/she already has a working app. He understands that not all features are there yet but what he has is working. Looking at the latest release and the provided features, he mentions a completely new feature. One that nobody had on his mind at the start of the project but a feature that can make his life as a customer much more productive. After every increment the customer feedback results in new features (not on the list) or adjustments of potential features and the product becomes more and more mature. Every time when the customer receives a new version you can see a happier customer.

If we look at figure 2, we see a steady flow of delivery within a fixed duration and using permanent agile teams (fixed costs). The scope and underlying quality criteria are flexible (dynamic) with frequent small deliveries (increments). All steps are performed repeatedly (iterative) to deliver a feature or user story until the required quality is reached. Management control is focusing on customer value delivery. Value delivery to the customer takes place after every deployment of an increment of the app.

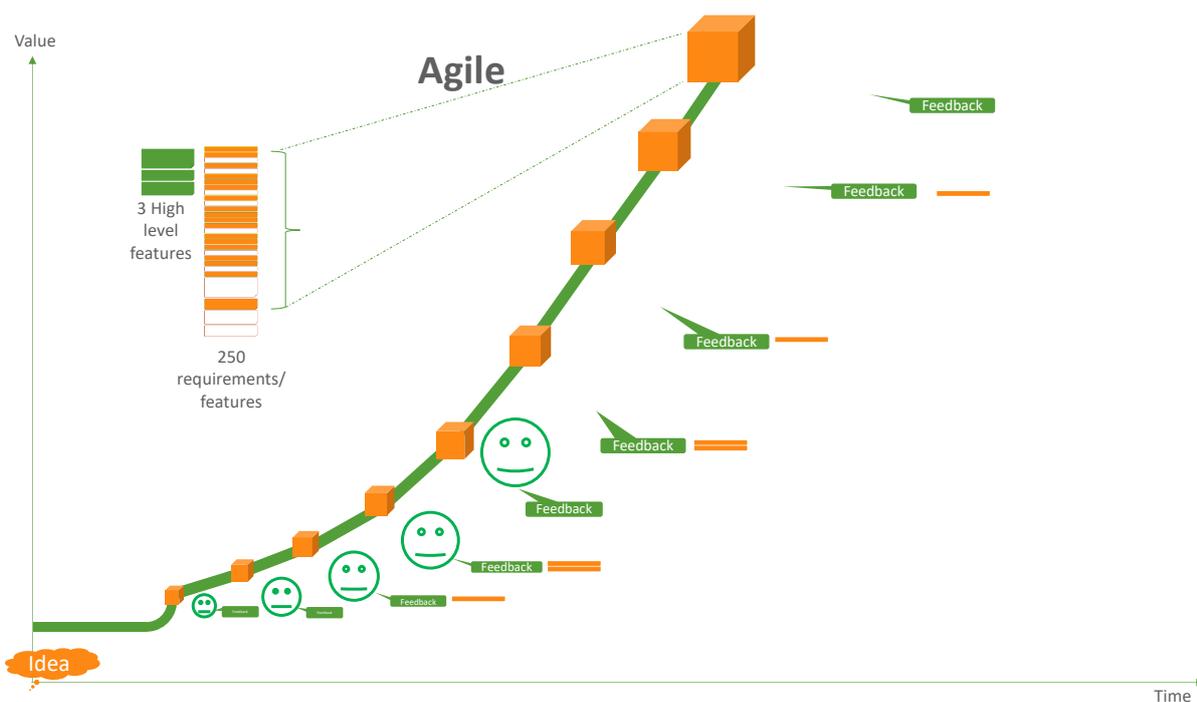


Figure 2: Agile delivery of an app

### Waterfall versus agile delivery results

If we take a closer look at the two products from both the waterfall as well as the agile approach, we see a product with 250 features and a not so happy customer and one with only 150 features and a very happy customer (see figure 3).

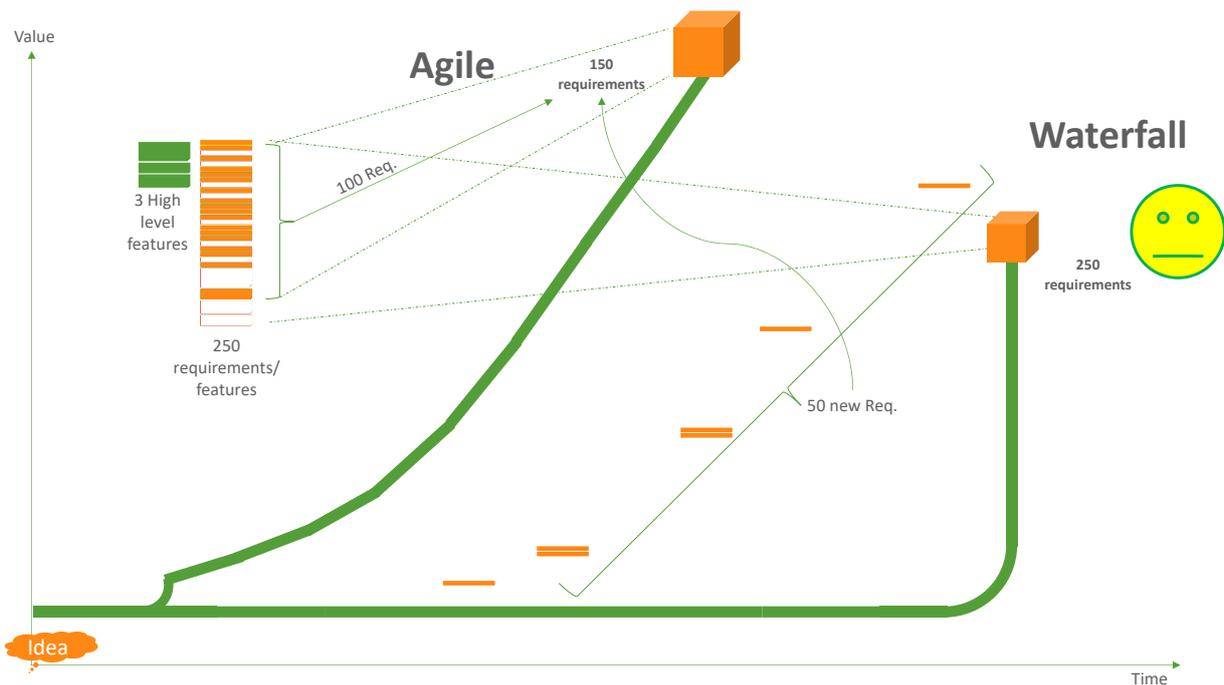


Figure 3: Waterfall versus agile delivery results

And if we even look in more details to the product delivered by the agile approach, we only see 100 features from the original list and 50 are new or adjusted features. This is in line with some important principles of the agile manifesto<sup>6</sup>: *Simplicity – the art of maximizing the amount of work not done – is essential* (only 150 features instead of 250 features) and *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage* (50 new or adjusted features were delivered). And as a result, a very happy customer (*Our highest priority is to satisfy the customer through early and continuous delivery of valuable software*)!

## Incremental versus iterative

### Different approaches

As stated, I noticed that students often mix the words iterative and incremental together. In figure 4 you can find four quadrants as a result of a horizontal line presenting incremental or not and a crossing vertical line representing iterative or not<sup>7</sup>.

In the left lower corner, we see the approach where there are no iterations and no increments. This is the waterfall approach. All activities (design, analysis, build, test and deploy) are performed once for the entire project. In this case we see a single delivery of the final product based on a fixed scope. Customer value can only be achieved after the delivery of the final product. One of the key goals in this approach is to manage cost. See also the waterfall approach to develop the payment app in the first part of the article.

<sup>6</sup> <https://agilemanifesto.org>

<sup>7</sup> See YouTube for a very simple version of this figure: <https://www.youtube.com/watch?v=20SdEYJEbrE&t=31s>

In the lower right quadrant, we see an incremental approach without iterations. This is a staged or incremental delivery of smaller parts the product. All activities for a given stage (design, analysis, build, test and deploy) are performed once. Within a given stage the scope is fixed, but the total product is based on a more dynamic or flexible scope. Customer value can be achieved after every delivery of the product. One of the key goals in this approach is speed of delivery.

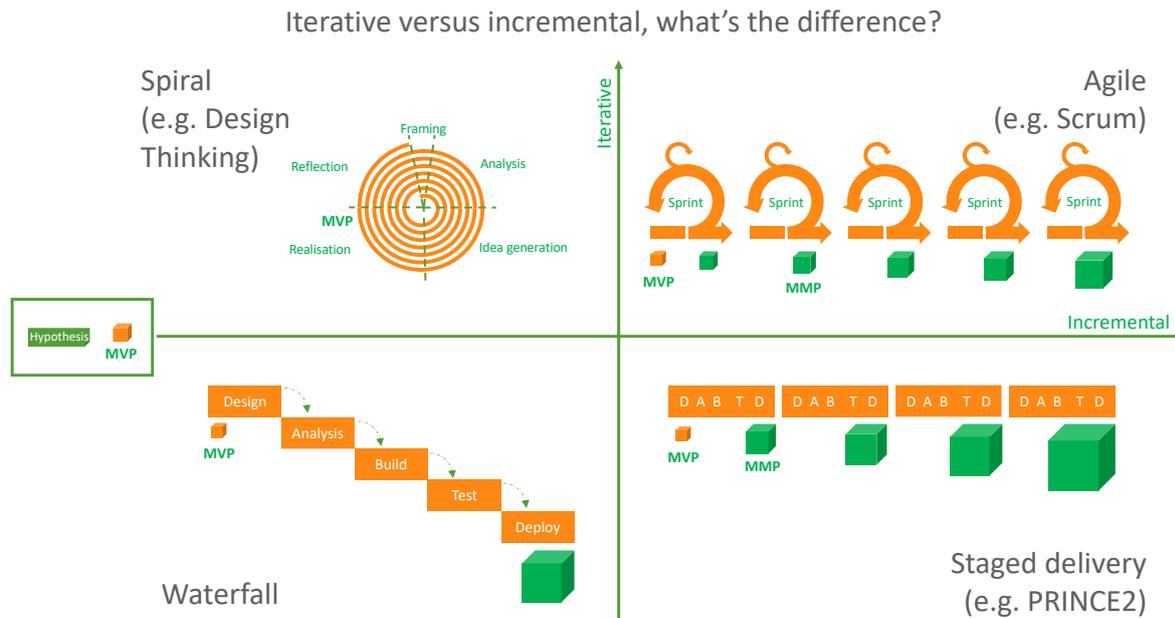


Figure 4: Iterative versus incremental

In the left upper quadrant, we see a spiral or iterative approach without increments. This is a single delivery where the final product is created by using several iterations. A good example of this approach is design thinking. In the graph you see a sequence of the activities framing, analysis, idea generation, realization and reflection. This sequence will be performed repeatedly or iteratively where in every iteration you get closer to the final, correct or required, product. In many cases this final product is a prototype or model. In this spiral approach we have a dynamic or flexible scope. Customer value can only be achieved after the delivery of the final product. One of the key goals in this approach is the correctness of the solution.

In the upper right quadrant, we see the agile approach with increments and iterations. Scrum is a good example<sup>8</sup> of this approach. At the end of each increment, often called a sprint or timebox there is a delivery of an increment of the product. This increment is the result of many iterations to develop small but correct parts, often called user stories or backlog items, of the product. The final product will be delivered piece by piece. In this agile approach we have a dynamic or flexible scope. Customer value can be achieved after every delivery of the product. One of the key goals

<sup>8</sup> See my article with more than 70 agile frameworks or approaches: Portman, H. (2019). A bird's eye view on the agile forest; *PM World Journal*, Vol. VIII, Issue X, November. <https://pmworldlibrary.net/wp-content/uploads/2019/11/pmwj87-Nov2019-Portman-birds-eye-view-on-agile-forest.pdf>

in this approach is customer value via frequent deliveries and customer feedback. See also the agile approach to develop the payment app in the first part of the article.

### MVP or MMP?

In figure 4 you can also find the acronyms MVP and MMP. MVP stands for minimum viable product (Eric Ries, Lean Startup) and is a version of a new product or service which allows a team to collect the maximum amount of validated learning about customers with the least effort. The MVP for the Dropbox service was a simple movie. This means that the P in MVP could be a completely different product in comparison with the final product.

I often use the following example of a new financial product. An enthusiastic sales manager has a great idea regarding a new financial product. He thinks that they can sell at least 100.000 of these products. Together with some finance experts, they design the product in a couple of months. A development team is assigned, and it takes them 4 months to develop the product. In parallel commercial brochures are designed and the product is launched with a big event. Unfortunately, just a few people buy the product. If we follow Eric Ries' approach we could make the assumption that 10% of their web users are interested in this product. To test this hypothesis, they develop a MVP. In this case a simple button on the homepage. If you click you get a screen with a message regarding this new product and the question to leave your mail address behind if you are interested. Less than 1% of the visitors pressed the button. The product will not be developed. They saved a lot of scarce resources.

If we take a closer look at figure 4, we see the potential usage of MVP's in all quadrants. When using a waterfall approach, you could create a MVP in the first design phase to check if there is a business justification for the project. The same can be done in the first phase of the first increment when following a staged delivery. In some cases, the result of your spiral, or design thinking approach could be a MVP. At the beginning of an agile approach the MVP could be beneficial too.

Many people see the first product delivered at the end of your staged delivery as the MVP. This could be the case but, in most cases, this is not a MVP but a MMP. MMP or minimum marketable product is the smallest product that can bring value to your customer. See the staged delivery or agile approach where a MMP could be deployed.

### What does an agile delivery look like?

Now it is clear that incremental and iterative are not the same and we understand the usage of MVP and MMP, we can explore in more detail how an incremental and iterative delivery could look like. You probably have seen Jeff Patton's famous example of the Mona Lisa where the painting is created piece by piece (staged delivery) or in the first increment only a rough sketch and every new iteration more details are added to the sketch and ultimately you have the final painting (incremental and iterative or agile delivery). In the first situation you must already have a detailed idea of the final product and in the second situation you just need a high-level outline and changes are much easier to make. If we look at figure 5, we see a story map for a new product called ABC.

### Story map for product ABC

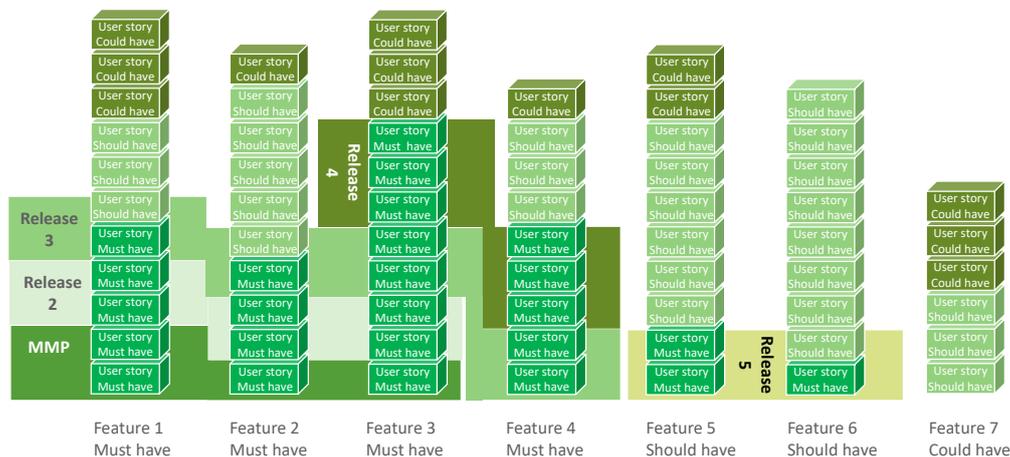


Figure 5: Story map for product ABC

The product owner envisioned seven features for this product. The first four features are must haves. Feature 5 and 6 are should haves and the last feature is a could have. Many would call this MoSCoW prioritization (Must haves, Should haves, Could haves and Won't haves). Every feature in itself can be sliced down in smaller parts. In the figure you see features with must, should and could have user stories. A feature can be a must have but that doesn't mean that all the underlying user stories are must haves too. Or a feature can be a should have but if you implement that feature some user stories are must haves and others are should or could haves.

To implement this ABC product, you see five increments or releases. The first one is the minimum marketable product. This MMP consists of the first two must have user stories of feature 1, and the first must have user stories of feature 2 and 3. Release 2 contains the next two must have user stories of feature 1, 2 and 3 (iterative development). Product development continues by implementing the next releases. With every release the customer value increases. After release 5 the product owner stops implementing user stories. Feedback from the customer showed him that the ABC product is 'fit for purpose' and he takes the Agile manifesto's principle *Simplicity – the art of maximizing the amount of work not done – is essential*, into account and stops further development.

## About the Author



### **Henny Portman**

The Netherlands



**Henny Portman**, a partner of HWP Consulting, has 40 years of experience in the project management domain. He was the thought leader within NN Group of the PMO domain and responsible for the introduction and application of the PMO methodologies (portfolio, programme and project management) across Europe and Asia. He trains, coaches and directs (senior) programme, project and portfolio managers and project sponsors and built several professional (PM(O) communities. He is an accredited P3O, PRINCE2, MSP, MoP, PRINCE2 Agile, AgilePM, AgilePgM and AgileSHIFT trainer and a SPC4 SAFe consultant and trainer too. He is a P3M3 trainer and assessor and PMO Value Ring Certified Consultant. In addition, Henny is international speaker and author of many articles and books in the PM(O) field and blogger ([hennyportman.wordpress.com](http://hennyportman.wordpress.com)). Henny can be contacted at [henny.portman@hwpconsulting.nl](mailto:henny.portman@hwpconsulting.nl).