

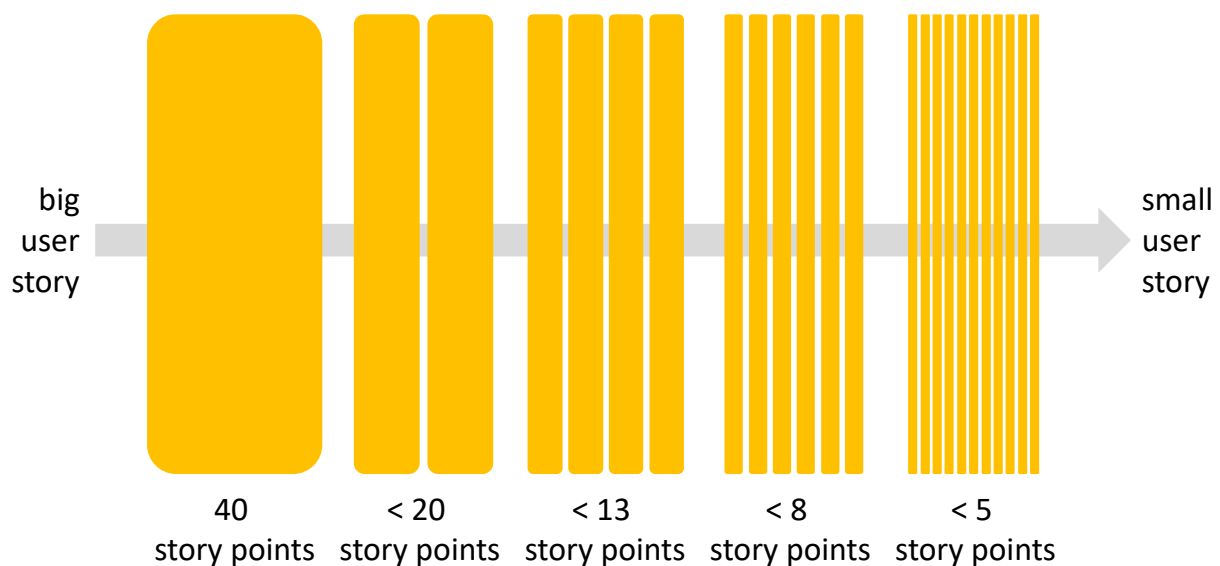
## Sensemaking in the Agile Forest

### Slicing user stories<sup>1</sup>

Henny Portman

#### Introduction

Ever eaten a cake whole? Surely it eats much easier if you serve the cake in small pie slices. It works the same way with large user stories. If you cut these large stories into small pieces, known as 'slices', delivery goes much faster. So read more about slicing user stories here. Parts of this article are based among other things on blogs from Niroshan Madampitige, *7 proven strategies for story slicing*, Ilia Pavlichenko, *9 Reasons to Slice User Stories Better*, Mark J. Balbes, *A practical guide to user story splitting for agile teams*.



Maybe the biggest story can't even be delivered in one sprint. This means you can't demonstrate or deploy or with other words there is nothing to be happy for as a customer. If you slice it in two parts you could hopefully deliver one. Slicing further helps you to be more predictive and deliver multiple working stories and learn faster due to the customer feedback.

Ilia Pavlichenko gives us nine reasons why we must slice user stories:

---

<sup>1</sup> How to cite this article: Portman, H. (2023). Slicing user stories, Sensemaking in the Agile Forest series, PM World Journal, Vol. VII, Issue I, January.

<b>Faster feedback</b>	A team that works with small stories and finishes them in 1-3 days can get feedback from the product owner and stakeholders without waiting till the end of the Sprint. If your Definition of Done (DoD) includes acceptance testing (UAT), then this is very important.
<b>Faster mistake correction</b>	It is important to note that the developer remembers the code he has written in the last 24 hours perfectly well. If during this time he receives any comments from other team members or some issues found, he will be able to correct things way faster.
<b>Postponing less valuable work</b>	When stories are properly sliced into a few smaller ones, the most valuable stories can go to the top and the rest may go to the bottom of the Product Backlog. The Scrum Team does not waste any time on performing something that has less value from the Product Owner's perspective.
<b>Better forecasts</b>	Let's assume the Development Team has an average velocity of 17 story points per Sprint. They usually work with big stories of 8 and 13 story points. There is a big chance that by the end of the Sprint, one of these stories will not be finished. Thus, the velocity fluctuates greatly from Sprint to Sprint (8 to 21 and even more).  If the development team had small stories, say, of 3 story points each, the fluctuations would be less noticeable. Consequently, the development team could come up with a more reliable forecast during the sprint planning and release planning for the product owner.
<b>More trust</b>	The development team, which stably finishes on average the same number of stories from Sprint to Sprint, becomes more predictable for the Product Owner and stakeholders. This helps establish an atmosphere of trust.
<b>Better spirit</b>	Finishing 6-10 user stories in each sprint strengthens the development Team's spirit, because people like the feeling of accomplishment. Think about it, which are your favorite football matches – those that end with a score of 0:0 or 3:2?
<b>Saving time on estimations</b>	If the development team delivers 6-10 small stories during a Sprint, it is very likely that those are approximately equal in size. This means that over time the development team will not have to estimate each story individually, just calculate the number of them.
<b>Risk mitigation</b>	Better slicing means a more detailed analysis of the items. Thus, a big chance that the development team identifies and mitigates risks that otherwise would have surfaced in the upcoming sprint. As they say, being forewarned is being forearmed.
<b>Better distribution of work within the Sprint</b>	Development teams can get overloaded at the end of each Sprint. This problem is very complex, but a finer slicing of stories is one of the ways of dealing with it. Small stories in the sprint lead to better parallelization of the work and the development team's performance usually improves.

## Slice horizontally versus vertically

Every user story consists of several superimposed architectural layers such as the underlying structure, customer logic and user interface. If you now slice the story horizontally, separate structure, customer logic and user interface stories emerge. These are three interdependent stories that can only deliver value when all three are working together. So, slicing only makes sense if it is done vertically, i.e. both a piece of structure, customer logic and user interface. Only then is it possible to deliver value sooner.

## INVEST acronym

Every user story, big or small, must comply with the INVEST rules:

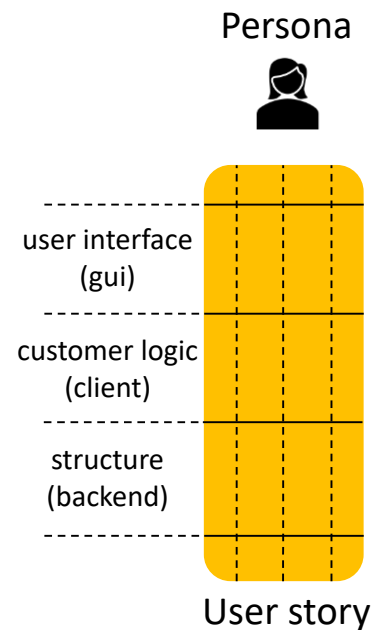
- **Independent** User stories must be able to be developed independently of each other.
- **Negotiable** The details in which user stories are described must leave room for negotiation.
- **Valuable** The user story must add value to both the user and the organization.
- **Estimable** If the user story cannot be estimated, it is apparently too vague, too big, or too risky.
- **Testable** The user story must contain acceptance criteria so that it can be tested.

## Slicing strategies

To make user stories smaller, you can choose from several strategies. Here, all split-out stories must still comply with the INVEST rules.

### Slicing based on functionality:

- Slice by capabilities offered. This is the most obvious way to slice a large feature. Look at the different capabilities being offered and split each one into its own story. For example, the capabilities “sort and search” may each be its own story. Splitting further, each way of sorting or searching may be its own story.
- Use the functional acceptance criteria as a tool to slice the user story. Consider a user story to enter name and address details. Acceptance criteria are then, for example, ‘name must contain alphabetic characters, postcode must be valid, house number without addition must be numeric, etc.’.
- look at testing the user story. What is the smallest possible piece of functionality that the team can test? This can then be the first user story.



- If a user story is intended for multiple groups of people, or ‘personas’, then the team can slice the story by group of people or ‘personas’.
- if the user story uses different types of data, e.g. national versus international data, you can create a story per data type.
- if the user story offers both complex and simple functionality, you can slice the user story into a simple and a more complex story.
- if the user story involves several interfaces, consider splitting the story per interface.
- does the user story combine different operations, e.g. create, retrieve, modify, delete, etc.? If so, you can create a story for each operation.

### **Slicing based on process steps:**

- first describe a matching MVP as the first user story.
- user stories describe the correct situation on the one hand, and what to do when handling errors and exceptions (happy/unhappy path) on the other. The team can pick up the handling of errors and exceptions in a separate story.
- User stories will often also include some quality criteria or non-functional requirements (NFRs for short). Examples include safety, accuracy, timeliness, et cetera. Each quality criterion can be added in a separate story.
- Split by target device. You can’t assume that users are interacting with your system using a standard computer. Various smartphones and IoT devices need to be considered in your stories. Splitting stories by device provides a more natural experience for your users.

### **Hyper-fine story slicing**

Mark J. Balbes offers a rule you can use when the user stories are still too big after slicing. In this situation, you can use a “zero/one/many” rule to spur the creative process and continue breaking down stories into vertical slices.

With the zero/one/many approach, you ask these three questions:

- What happens if we are dealing with nothing?
- What happens if we are dealing with one thing?
- What happens if we are dealing with many things?

### **Conclusion**

Slicing of user stories leads to smaller stories that the team can deliver faster. And therefore, the team adds value sooner. It results in faster feedback, faster error correction and deferring work with less added value. Moreover, smaller stories can be estimated faster and better so that predictions are more reliable. And that leads to more trust, higher team spirit, risk reduction and an easier division of labor.

## Sensemaking in the Agile Forest series

This article is part of a series of articles called *Sensemaking in the Agile Forest*. This series<sup>2</sup> consists of the following parts:

- [Portman, H. \(2022\). What is Agile? Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue I, January.](#)
- [Portman, H. \(2022\). What is Scrum? Sensemaking in the Agile Forest, series article 2, PM World Journal, Vol. XI, Issue II, February](#)
- [Portman, H. \(2022\). Is agile always better? Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue III, March](#)
- [Portman, H. \(2022\). The ideal Product Owner, Sensemaking in the Agile Forest series, PM World Journal, Vol. IX, Issue IV, April](#)
- [Portman, H. \(2022\). The Ideal Scrum Master, Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue V, May](#)
- [Portman, H. \(2022\). Is an agile team always autonomous? Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue VI, June](#)
- [Portman, H. \(2022\). What do iterative and incremental mean in Agile? Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue VII, July](#)
- [Portman, H. \(2022\). The Minimum Viable Product \(MVP\) unraveled; Sensemaking in the Agile Forest, series article, PM World Journal, Vol. XI, Issue VIII, August](#)
- [Portman, H. \(2022\). Prioritizing in an agile team, Sensemaking in the Agile Forest, series article, PM World Journal, Vol. XI, Issue IX, September](#)
- [Portman, H. \(2022\). Multitasking, task-switching or monotasking; Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue X, October](#)
- [Portman, H. \(2022\). Being predictable as an agile team; Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue XI, November](#)
- [Portman, H. \(2022\). Self-managing or self-organizing agile teams, Sensemaking in the Agile Forest series article, PM World Journal, Vol. XI, Issue XII, December](#)
- Slicing user stories
- Agile management products (burn-down and burn-up charts)
- Agile user testing (cohorts, A/B testing)
- The Kanban board (WIP-limit, cumulative flow diagram)
- Culture makes or breaks your agile transformation
- Getting started as an agile team (a pilot)
- The evolution of agile frameworks
- ?

Please let me know if you would like to add specific agile topics to this series.

---

<sup>2</sup> This series is based on a number of short Dutch blogs I made for Forsa Advies, a project management training organization in the Netherlands (<https://www.forsa-advies.nl>).

## About the Author



### **Henny Portman**

The Netherlands



**Henny Portman**, owner of Portman PM[O] Consultancy and was partner of HWP Consulting, has 40 years of experience in the project management domain. He was the project management office (PMO) thought leader within NN Group and responsible for the introduction and application of the PMO methodologies (portfolio, program, and project management) across Europe and Asia. He trains, coaches, and directs (senior) programme, project and portfolio managers and project sponsors at all levels, and has built several professional (PM(O)) communities.

Henny Portman is/was accredited in a variety of qualifications, including P3O, PRINCE2, MSP, MoP, PRINCE2 Agile, AgilePM, AgilePgM and AgileSHIFT trainer and an SPC4 SAFe consultant and trainer. He is a P3M3 trainer and assessor and PMO Value Ring Certified Consultant (PMO Global Alliance). On behalf of IPMA, he assesses mega and large projects for the IPMA Project Excellence Award. In addition to this, he is an international speaker, author of many articles and books in the PM(O) field, and an active blogger ( [hennyportman.wordpress.com/](http://hennyportman.wordpress.com/)).

Henny can be contacted at [henny.portman@gmail.com](mailto:henny.portman@gmail.com).