**Scrum Artifacts**

Scrum’s artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.[[1]](#footnote-1)

**Product Backlog**

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.[[2]](#footnote-2)

A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, its Product Backlog also exists.[[3]](#footnote-3)

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value.[[4]](#footnote-4)

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.[[5]](#footnote-5)

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product. A Product Backlog attribute that groups items may then be employed.[[6]](#footnote-6)

DEEP

DEEP is an acronym to summarize key attributes of a good product backlog. DEEP stands for:

* **Detailed Appropriately.** User stories on the product backlog that will be done soon need to be sufficiently well understood that they can be completed in the coming sprint. Stories that will not be developed for a while should be described with less detail.[[7]](#footnote-7)
* **Estimated.** The product backlog is more than a list of all work to be done; it is also a useful planning tool. Because items further down the backlog are not as well understood (yet), the estimates associated with them will be less precise than estimates given items at the top.[[8]](#footnote-8)
* **Emergent.** A product backlog is not static. It will change over time. As more is learned, user stories on the product backlog will be added, removed, or reprioritized.[[9]](#footnote-9)
* **Prioritized.** The product backlog should be sorted with the most valuable items at the top and the least valuable at the bottom. By always working in priority order, the team is able to maximize the value of the product or system being developed.[[10]](#footnote-10)

**Assessment: Product Backlog**

* Definition[[11]](#footnote-11):
	+ Identify the Product Backlog as an ordered and emerging list of user needs plus anything else that is required to fulfill the Product Vision.
* Contents[[12]](#footnote-12):
	+ Describe how the detail of the Product Backlog items will be tied to their position (or order) and how the Product Backlog contents will change over time.
	+ Describe how the Product Backlog will contain functional, non-functional, architectural, and infrastructural elements as well as risks that need to be removed or mitigated. [Explain] why wherever possible, items on the Product Backlog will be in vertical slices (i.e., each providing value to the user).
* Responsibility and Participation[[13]](#footnote-13):
	+ Identify why the Product Owner is ultimately responsible for the content and state of the Product Backlog, though anyone is able and encouraged to contribute to the Product Backlog.
* The Product Backlog, understand what the product backlog is (and is not)[[14]](#footnote-14):
	+ Understand the product backlog essentials: The product backlog is a list of requirements and other deliverables necessary to turn the vision into a successful product.
		- The product backlog must be: Detailed appropriately, Estimated, Emergent, and Prioritized (DEEP).
	+ Understand that the product backlog is not a substitute for a requirements specification.
		- The product backlog evolves and changes constantly; many of its items are coarse-grained and sketchy to start with; they are then progressively decomposed and refined. The emphasis shifts from documentation to conversation, from specifying requirements to having an ongoing dialogue.
	+ Understand that the product backlog can have different forms and shapes; it may be, for instance, a collection of paper cards or be held electronically as a spreadsheet or in a specialized tool. The product backlog can be flat or structured employing several columns and grouping items into themes, for instance.
	+ Understand that a product backlog can be augmented as appropriate with other artifacts, for instance, a spreadsheet showing business rules, a Visio diagram showing a workflow, a user interface prototype or mockup.
	+ Understand that these should be only used when necessary and kept as light as possible.
1. Scrum Guide July 2013, [www.scrumguides.org](http://www.scrumguides.org) [↑](#footnote-ref-1)
2. SG [↑](#footnote-ref-2)
3. SG [↑](#footnote-ref-3)
4. SG [↑](#footnote-ref-4)
5. SG [↑](#footnote-ref-5)
6. SG [↑](#footnote-ref-6)
7. Roman Pichler “Agile Product Management with Scrum” and Mike Cohn <https://www.mountaingoatsoftware.com/blog/make-the-product-backlog-deep> [↑](#footnote-ref-7)
8. Pichler & Cohn [↑](#footnote-ref-8)
9. Pichler & Cohn [↑](#footnote-ref-9)
10. Pichler & Cohn [↑](#footnote-ref-10)
11. CSM Learning Objectives, www.scrumalliance.org [↑](#footnote-ref-11)
12. CSM [↑](#footnote-ref-12)
13. CSM [↑](#footnote-ref-13)
14. CSPO Learning Objectives, [www.scrumalliance.org](http://www.scrumalliance.org) [↑](#footnote-ref-14)