

Agile Journey discussion series:  
("Sharing is Learning"):

# Agile – Product Owner – Own What?

By Jesper Olsen, February 2021

(Request/Feedback: [jesper@yescoaching.dk](mailto:jesper@yescoaching.dk))

---

**Agile**  
FOR  
**PRODUCT ? OWNER ?**

**Learn:**

**Product Owner**

- COMMUNICATOR**
  - communicates progress to external parties
  - negotiates with stakeholders
  - works with stakeholders on value to be realized
  - creates shared understanding
  - available to the team
  - participate in team events
- PLANNER**
  - plans sprints and releases to maximize the value
  - prioritizes the backlog
  - makes decisions
- VISIONARY**
  - has a clear vision on what is required
  - sells project to the organization
  - defines goals
- DOMAIN EXPERT**
  - has domain knowledge
  - is respected in the organization
  - represents the business
- TEAM PLAYER**
  - motivates team with clear, elevated goals
  - provides team with information required to deliver value

*(The infographic also features a cartoon character with spiky hair and glasses pointing upwards.)*

# Table of Contents

Welcome.....	2
Introduction.....	3
What is Product and ownership? .....	3
Product Owner reflections .....	4
Product Owner reflections .....	9
Wrap up: Product Owner – Own What? .....	12
Appendix.....	13
Apendix01: Source/References and other inspiration:.....	13
Apendix02: Agile Manifesto .....	14
Apendix03: Agile Principles .....	15
Apendix04: “Do the Right Things Right” .....	16
Apendix05: “Product complexity references” .....	17

## Welcome

You just entered my brainstorming room. In this room I share my thoughts on agile subjects and this time the Product Ownership term.

This article and room are primarily for you if you:

- are Product Owner or cooperate with one.
- Think that product flow, strategy, ownership and hierarchy is interesting.
- Work in an Agile environment or would like to.
- Are curious in general on agile and agile elements.
- Just read this article to give me feedback.

Whatever excuse you have - welcome in my brainstorming room. 😊

### Cliffhanger:

**This is not a traditional Product Owner introduction and I promise to give you another view on both Products and Ownership 😊**

# Introduction

If you work with or hear about Agile you have probably met the term Product Owner (PO) and maybe Proxy PO also.

This is for a good reason since this is often the link between IT and Business and often also is the bridge between traditional and new agile way of working.

In this Article I'll touch one role definition – but use most of the time challenging the two primary elements:

- **Product:** Do we know our product hierarchy/dependencies or is it too complicated?
- **Ownership:** Do we know what we own and what does this mean on a day to day business?

The reason why this is important is simply this: if we try to organize and define our PO role on an unknown baseline, we will not succeed.

This article in my “Agile Journey discussion Series” (See reference) is actually just a reflection (Refinement) of another way to look at product and ownership.

The intention is to share reflections, not science – that's why I call it “Brainstorming room for improvement” or in Agile terms: “Inspect”, (“Discuss/Learn”), and maybe” Adjust and adapt”



## What is Product and ownership?

If we take our agile glasses on there are a lot of terms that emphasize how important and central the role **Product Owner** is.

If you open nearly any agile book you meet terms like:

- **MVP** (Minimum Viable Product) –
  - A term to try to just focus on the smallest possible part of a product that can show business value or be a PoC (Prove of Concept)
- **PSP** (Potential Shippable Product)
  - A term from Scrum that again - just like MVP - identifies the necessary part of a product to be of actual use for business.
- **PBI** (Product backlog Item):
  - The element/brick that - together with many other PBI's - make or change the product.

- A.s.o (I think you get the point 😊)

So, a product is and can be quite a central and complicated term. It's often the actual delivery between IT, Business and Customer and the reason why our company exists.

So, for the same reason the ownership and accountability for this is also very important.

But before we come to ownership, I'll use more time on the product, flow and hierarchy so we know what to own 😊

In the next chapters I will go more into details on product and end return for the ownership at the end (so don't be afraid)

## Product ~~Owner~~ reflections

If you ask a person from the industry what they do, they will probably say:

*"We produce products"*

If you ask the same question in a new, modern company, they will probably also add that:

*"they produce value, outcome and much more than just the product".*

So in many ways a product is not just a physical product, but more like the outcome/result of what you can achieve by using the product.

On top of this the product is dependant on a lot of other products inside and outside of the company.

So how do we address this product complexity and take ownership ?

Before we sail on the owner'ship' we will continue our brainstorm on products.

Another way to better understand the complexities is to try to map the customer journey, do value stream mapping (VSM) and other similar approaches.

If you go for eg VSM and use the Lean approach you will often see a lot of systems/ teams/ dependencies and identify a lot of potential improvement areas and wastes etc.

Here are two examples:

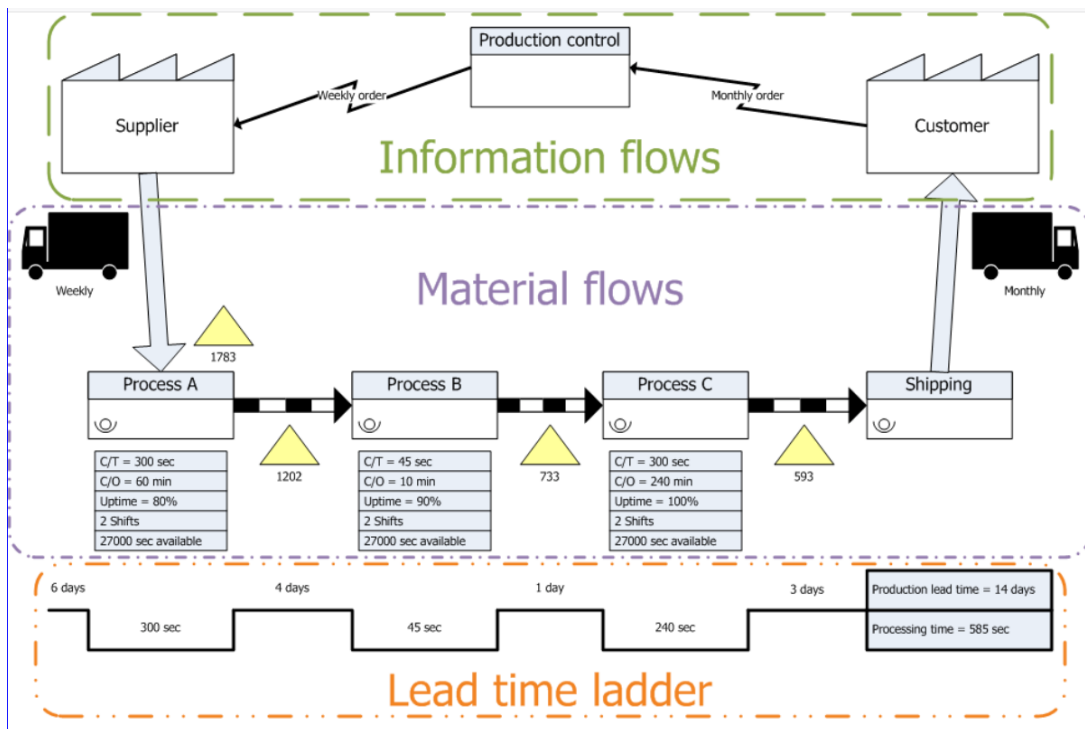


Fig01: Tradition Lean VSM approach (Focus on time/waste)

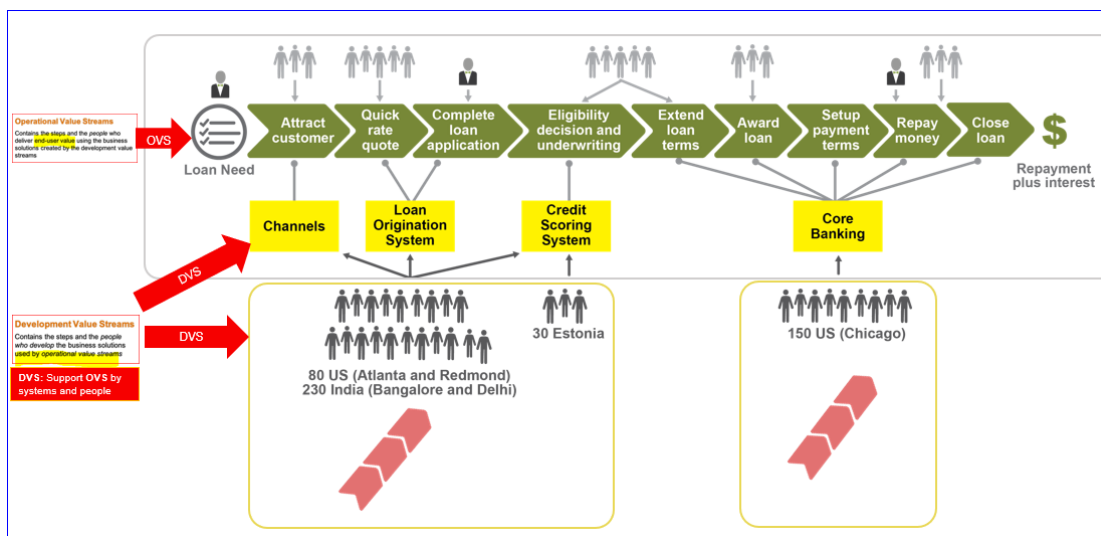


Fig02: SAFe VSM focus (Focus on Operation Value Stream and supporting system/teams and Development Value streams)

So the point here is not to introduce VSM and other appoces for you, but to highlight how complicated this product-/Value flow could be.

So returning to the question:

**“What product does the product owner own and how does this work with all this dependancies” ?**

Often the approach to “solve” this is to change the organisation to fit this product strcure/flow and in Agile would scale to fit this.

But inspired by a short and quite interesting Podcast (See Ref\_A05\_01 and Ref\_A05\_02 in appendix 5 ) we maybe we should turn this a little bit around.

**“Should we try to organise around a very complex product structure or should we simplify our product structure to fit a more simple organisational structure ?**

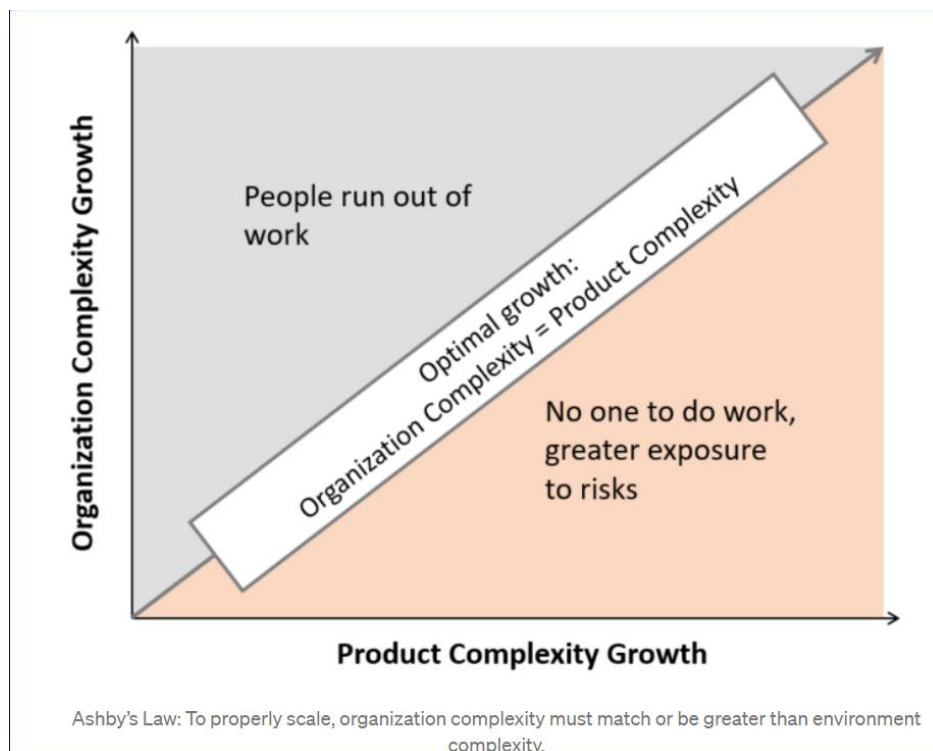
So basicly NOT try solve an complex product hirachi with just another complex organisation to handle this.

Interesting, but of course not that easy – but let us give it a try or at least a thought.

As I understand the Podcast (Listen youself), this is some of the product design principles that could be relevant to consider to make it more simple:

- 1) Product split on stakeholders
- 2) Product split in functional needs (Features?)
- 3) Product split on remove unused functionality/Features (Creative Destruction)
- 4) Remove product split on OSI layers (e.g. Applications/database layer)
- 5) Consider more microservices/widgets etc. that work separately

Another article “The Product Complexity Paradox” (See Appendix 5) reflects on this even more. For example, this article has a very nice graf explaining how product/organisation complexity have to follow each other to keep Optimal growth.



Another article “PRODUCT COMPLEXITY MANAGEMENT” (See Appendix 5) also have som management ideas on how to reduce product complexity by for example avoiding these pitfalls:

**Complexity usually builds up over time.** Many smaller decisions along the way can end up leading to an excessive amount of product complexity.

Below are examples of actions that add complexity:

1. Adding a product to make a range more complete
2. Adding a new product for a large customer
3. Keeping an outdated product as it still sells
4. Testing new products and innovations in the market
5. Designing products from a blank paper and not thinking about possible reuse of components
6. Local autonomy when designing products
7. Having divisions and business units source their own components, as they know their local market
8. Keeping entrepreneurship by not fully integrating acquisitions

**The bottomline is to consider another product split and flow if this has become too complex and has too many dependencies between Products and Teams.**

Another simple metaphore for this:

Consider the product portfolio as a lot of very different LEGO bricks that each are based on the customers' specific needs. Then this could be combined in 1000 ways and adjusted according to marked changes and actual customer needs.



In that picture, the Product owner “just” owns his brick and the primary standard is just that the interface should fit other bricks.

Actually, this approach was a surviving gamechanger when the new CEO of LEGO took over in 2004 and did a simplification of their product hierarchy and invented a framework for Product Development and Portfolio Management (See appendix 05).

Another metaphore is the MS office packet that includes a lot of elements: Word, Excel, PowerPoint etc. - and each product can be released seperately and even in itself could be split up in even smaller bricks. Here again, the primary dependancy is fullfilling some interface standards and format and the PO's role is "just" one application level.

So the primary take away from this reflection is this:

**"Consider you product hierarchy design before you add more complexity to solve that."**

Now let's reflect on the Ownership of Product owner.



# Product Owner reflections

So, if we now have a better idea of the product or at least the complexity, it's time to reflect on what this ownership means.

This is not so easy as it may sound, and I've seen many definitions of this.

Here are some examples of definitions:

**Product Owner**

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Source: Scrum guide (Version 2020)

**Product Owner**

The *Product Owner (PO)* is a member of the **Agile Team** responsible for defining **Stories** and prioritizing the **Team Backlog** to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the **Features** or components for the team.

The PO has a significant role in maximizing the value produced by the team and ensuring stories meet the user's needs and comply with the **Definition of Done**. For most enterprises moving to Agile, this is a new and critical role, typically translating into a full-time job, requiring one PO to support each Agile team (or, at most, two teams).

This role has significant relationships and responsibilities outside the local team, including working with **Product Management**, **Customers**, **Business Owners**, and other stakeholders.

Source: SAFe guide (Version 5.0)

So even according to these well proven "Best Practices" there are also differences - for example on the accountability and responsibility on value creation.

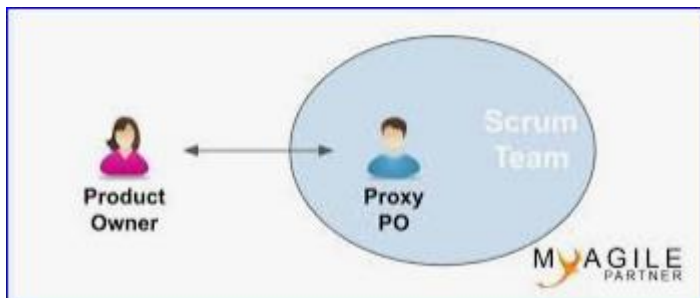
And in real organisation implementations there is even more tailoring of this to fit for purpose/use and organisation.

For example, you often see extra roles supplementing these roles - for example: “Proxy Product owner”, “Product Manager” and I have even seen “Technical Product Owners (TPO)” and “Product Development Managers(PDM)”.

So yes, there is a good reason for some confusion and any definition is off course only **Best Practices** if it works in the organisation.

But let’s look at two of the “extra” roles before I return to my key recommendations.

**Proxy Product Owner (PPO):** Is often used to cover for a very busy Product Owner, to cover a “missing link” between business and IT and hired by IT side. This could also be a symptom of a very complex product or technical demanding product.

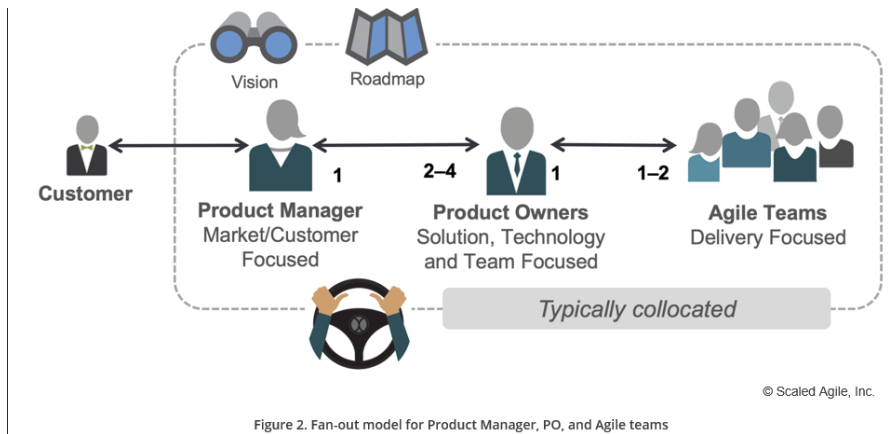


**Product Manager (PM):** This is often used if the product setup is so complicated that a product consists of more than one product item. Each having its own Product Owner. Another reason for a PM is because the recurring product strategy and stakeholder Management simply is to critical and time consuming. Again, a symptom (positive or negative) of a very complex product hierarchy or marked situation.

Below some examples of SAFe recommendations on this PM/PO spilt:

Product Manager <i>Drives the PI and product</i>	Product Owner <i>Drives the Iteration</i>	Agile Team <i>Drives program execution</i>
Owns Program Backlog	Owns Team Backlog(s)	Builds Quality-In, evolves Agile architecture
Defines Features, PIs, and Releases	Defines Iterations and Stories	Owns estimates
Owns Vision, Roadmap, pricing, licensing, ROI	Contributes to Vision, Roadmap, ROI	Evolves the Continuous Delivery Pipeline
Collaborates on Enablers	Accepts Iteration increments	
<i>Build the right thing...</i>		<i>...Build the right way</i>

© Scaled Agile, Inc.



So, there is no real bible to decide on this Product Ownership, but here are my recommendations and “Principles at thumb”:

- 1) Make sure you know your product and in a complex product hierarchy know the borders, dependencies and have good agile governance structure to handle this.
- 2) Consider the Product Owner as a daily and reachable member of the agile team
- 3) Let the Product owner be part of the Business and do GEMBA (Being at “the actual place” where it happens) so he/she know the actual use/value/outcome of the product
- 4) Know where in the Value Stream this products fit’s and how this creates value
- 5) Be an accountable owner, but empower the responsibility to the rest of the Agile team to add value at all levels
- 6) Don’t be a manager, but part of the Agile team as a sort of **Servant Product Leader**
- 7) Be on top of motivating and refining the items of the Product Backlog and at the right level (Use Definition of Ready (DoR))
- 8) Keep the focus of Business value and make sure this is reflected in priorities in Product backlog at any time.
- 9) Be yourself and respect the 4 P’s (Passion, People, Product and Process)
- 10) Inspect, Adjust and Adapt your role and above “rules” when relevant.

In other words: **“Be Agile”**

# Wrap up: Product Owner – Own What?

Congratulations, you've reached this sweet spot, hope you enjoyed my reflections and maybe even built your own reflections on top of that.

My wrap up/conclusion is:

**Both Product and ownership are complex and if the product hierarchy and organisation are not agile, aligned and product oriented – the role can be nearly “Mission impossible”.**

So, some final learning points to make it more possible:

- 1) **“Simplify your product hierarchy before you add more complexity to solve that.”**
- 2) **Make your own “Principles at thumb”**
- 3) **“Don't just Do Agile - Be Agile” and part of the Agile Team**
- 4) **Remember this is journey and that you learn by doing and crying.**



I hope you got some inspiration and would like to join me on other agile journeys or parties (See References)

Final favourite quotes/souvenirs:

**“The journey of a thousand miles begins with one step” (Lao Tzu)**

**“The first giant step is in your mind” (Yesper)**

**“You do everything twice – first in your mind then in reality (Stephen Covey)**

**“Change your mind for a change” (Yesper)**

With this I thank you for joining my so far on my journey – to be continued....

**YESper** (Feedback: [Yesper@yescoaching.dk](mailto:Yesper@yescoaching.dk))

# Appendix

## Appendix01: Source/References and other inspiration:

Ref. 01: LinkedIn profile: <https://www.linkedin.com/in/jesperbergerolsen/>

Ref. 02: Own coaching company [www.yescoaching.dk](http://www.yescoaching.dk)

Ref. 03: Inspiration by “Coaching Agile Teams” by Lyssa Adkins

Ref. 04: Inspiration by course “Advanced Agile Team Coaching” and book “The Hitchhikers’ Guide to Agile Coaching” by Agile42

Ref. 05: Inspiration by “Clean Agile – Back to Basics” by Robert C. Martin

Ref. 06: Book: “Seven habits of highly effective peoples” (Steven R. Covey)

Ref. 07: Book: “Systemisk Coaching” by Hanne V. Moltke and more

Ref. 08: LinkedIn Doku: “Agile introduction made simple”:

[https://www.linkedin.com/posts/jesperbergerolsen\\_agile-introduction-made-simple-version-02-activity-6695996989156454400-pZL9](https://www.linkedin.com/posts/jesperbergerolsen_agile-introduction-made-simple-version-02-activity-6695996989156454400-pZL9)

Ref. 09: LinkedIn Doku: “Does Agile Framework actual work”:

[https://www.linkedin.com/posts/jesperbergerolsen\\_does-agile-frame-actual-work-activity-6680542323400564736-ilpc](https://www.linkedin.com/posts/jesperbergerolsen_does-agile-frame-actual-work-activity-6680542323400564736-ilpc)

Ref. 10: LinkedIn Post: “Agile Hangover and Back to the Basic”:

[https://www.linkedin.com/posts/jesperbergerolsen\\_agile-scrum-safe-activity-6670335483358277632-dEGH](https://www.linkedin.com/posts/jesperbergerolsen_agile-scrum-safe-activity-6670335483358277632-dEGH)

Ref. 11: LinkedIn Post: “Be on Track with DevOps”:

[https://www.linkedin.com/posts/jesperbergerolsen\\_agile-scrummaster-safe-activity-6618435085634977792-gWoG](https://www.linkedin.com/posts/jesperbergerolsen_agile-scrummaster-safe-activity-6618435085634977792-gWoG)

Ref. 12: LinkedIn Post: “Definition of Well Done”:

[https://www.linkedin.com/posts/jesperbergerolsen\\_agile-scrummaster-safe-activity-6612745994717147136-Wjl6](https://www.linkedin.com/posts/jesperbergerolsen_agile-scrummaster-safe-activity-6612745994717147136-Wjl6)

Ref. 13: LinkedIn Post: “Product Owner – more than just the owner” : [https://www.linke](https://www.linkedin.com/posts/jesperbergerolsen_agile-scrum-productowner-activity-6610869358396751872-DdH6)

[din.com/posts/jesperbergerolsen\\_agile-scrum-productowner-activity-6610869358396751872-DdH6](https://www.linkedin.com/posts/jesperbergerolsen_agile-scrum-productowner-activity-6610869358396751872-DdH6)

Ref 14: LinkedIn Doku: “Are you a real Agile coach?”

[https://www.linkedin.com/posts/jesperbergerolsen\\_real-agile-coach-activity-6744575035144495104-tHJQ](https://www.linkedin.com/posts/jesperbergerolsen_real-agile-coach-activity-6744575035144495104-tHJQ)

Ref 15: LinkedIn Doku: “Agile Party for NewComers?”

[https://www.linkedin.com/feed/update/urn:li:activity:6747087224568901632/?commentUrn=urn%3Ali%3Acomment%3A\(ugcPost%3A6747087223474200576%2C6750021231346536449\)](https://www.linkedin.com/feed/update/urn:li:activity:6747087224568901632/?commentUrn=urn%3Ali%3Acomment%3A(ugcPost%3A6747087223474200576%2C6750021231346536449))

Ref 16: LinkedIn Doku: “Agile – Out of Control?”

[https://www.linkedin.com/posts/jesperbergerolsen\\_agile-out-of-control-new-version-20200108-activity-6753309571168686080-LrWj](https://www.linkedin.com/posts/jesperbergerolsen_agile-out-of-control-new-version-20200108-activity-6753309571168686080-LrWj)

# The Agile Manifesto

<b>Individuals and interactions</b>	over	Processes and Tools
<b>Working Product</b>	over	Comprehensive Documentation
<b>Customer Collaboration</b>	over	Contract Negotiation
<b>Responding to change</b>	over	Following a plan

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


[www.agilemanifesto.org](http://www.agilemanifesto.org)

## Comments:


- **Individuals and interactions over processes and tools**
  - Pasion, People and empowered team beats structure
- **Working software over comprehensive documentation**
  - Real result is outcome for business not administration.
- **Customer collaboration over contract negotiation**
  - Business should be considered is part of Agile team and not “the other side”
- **Responding to change over following a plan**
  - Embrace change and remember plan is not outcome

## The 12 Agile Principles


**1 Satisfy the customer**




**2 Welcome change**




**3 Deliver frequently**




**4 Work together**



**5 Trust and support**




**6 Face-to-face conversation**

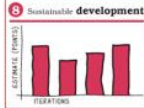


1. Customer satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Deliver working software frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be **trusted**
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software (business value / outcome) is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly


**7 Working software**




**8 Sustainable development**




**9 Continuous attention**




**10 Maintain simplicity**



**11 Self-organizing teams**



**12 Reflect and adjust**



### Comments:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - a. Focus on value, Output, Innovation and Outcome (See appendix03)
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - a. Embrace changes, focus on enablers for this and keep options open
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - a. Design for flexibility, T2M and DevOps
4. Businesspeople and developers must work together daily throughout the project.
  - a. One Agile team and Product Owner is more than just owner
5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
  - a. Prioritize the two first P's **Passion and People** and the Job will get done (Product/Process).
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - a. Walk the talk and "Listen to be understood"
7. Working software is the primary measure of progress.
  - a. Result, Output, Outcome and value (and more than just software)
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - a. Consider long-term and Non-Functional Requirements (e.g. Maintainability)
9. Continuous attention to technical excellence and good design enhances agility.
  - a. Including reducing Technical Depth and long-term enablers as part of the job.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
  - a. Just go for DoD and remember Software is just an enabler for real outcome (See Appendix03 and "Mona Lisa painting")
11. The best architectures, requirements, and designs emerge from self-organizing teams.
  - a. Good Product comes from Pasion, People, motivation and real empowerment
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
  - a. Never stop learn and unlearning

## Apendix04: "Do the Right Things Right"





## Appendix05: "Product complexity references"

Ref\_A05\_01: Podcast on product management, split and scaling:

<https://podcasts.apple.com/dk/podcast/the-liberators-network/id1475286669?l=da&i=1000496597211>

Ref\_A05\_02 : Larman's Laws of Organizational Behavior:

[https://www.craiglarman.com/wiki/index.php?title=Larman%27s Laws of Organizational Behavior](https://www.craiglarman.com/wiki/index.php?title=Larman%27s_Laws_of_Organizational_Behavior).

Ref\_A05\_03: Product development and portfolio management processes at LEGO:

[http://apppm.man.dtu.dk/index.php/Product\\_development\\_and\\_portfolio\\_management\\_processes\\_at\\_LEGO](http://apppm.man.dtu.dk/index.php/Product_development_and_portfolio_management_processes_at_LEGO)

Ref\_A05\_04: The Product Complexity Paradox: <https://productcoalition.com/the-product-complexity-paradox-6978c1305c3d>

Ref\_A05\_05: PRODUCT COMPLEXITY MANAGEMENT:

<https://implementconsultinggroup.com/product-complexity-management/>