

## ***Sensemaking in the Agile Forest***

### **Is an agile team always autonomous?<sup>1</sup>**

**Henny Portman**

#### **Introduction**

Autonomous teams are more agile than dependent teams. And many organizations are currently striving for agility in order to quickly respond to new developments. But is an agile team always autonomous? Or can an agile team always be made autonomous?

#### **What is autonomy?**

What does 'autonomous' mean? It means that you are not dependent on anyone or anything. 'Dependent' can have several meanings:

1. Dependent on someone, a team or something
2. Needing support, care or assistance from someone, a team or something
3. Being in subordinate relationship to someone, a team or something.

#### **Intersection versus roundabout**

But why are autonomous teams more agile than dependent teams? To illustrate this proposition and to clarify the concepts of 'autonomous' and 'dependent', you can make a comparison with an intersection versus a roundabout. The intersection is a metaphor for dependent teams. And a roundabout is the metaphor for autonomous teams. In the table below you will find important differences in characteristics.

<b>Intersections as a metaphor for dependent teams</b>	<b>Roundabout as a metaphor for autonomous teams</b>
Traffic lights are controlled by a central command and control system. Traffic lights directly influence drivers' activities (driving, stopping).	Roundabouts create the conditions for self-regulation by drivers
Intersections with traffic lights represent hard controls	Roundabouts without traffic lights represent soft controls
Control & compliance	Trust & autonomy
32 points of potential collision	8 points of potential collision

<sup>1</sup> How to cite this article: Portman, H. (2022). Is an agile team always autonomous? Sensemaking in the Agile Forest series, PM World Journal, Vol. XI, Issue VI, June.

Slower traffic flow (inflexible)	Faster traffic flow (more agility)
More accidents	Fewer accidents

PORTMAN  
**PMO**  
CONSULTANCY

Autonomous or not?

Source: Blue Striped Frog magazine + Basile Lemaire 8

## Decision-making

An important aspect in which autonomy is expressed is decision-making: can you, as a team, make decisions yourself or do you have to escalate? The importance of autonomy is also reflected in the research of The Standish Group. This organization has conducted research into the causes of project failure. One of the most important reasons is the excessively long lead time required to make decisions. This is also referred to as 'decision latency'. It is therefore important to have decisive decision-making and therefore sufficient autonomy.

## Five critical skills to build an autonomous team

In the Project Development magazine<sup>2</sup> Kate Dames explained five critical skills to build an autonomous team.

Critical skills	Description
Trust and respect	When you have people that trust and respect each other, it's possible to allow them to make their own decisions, knowing that if it falls outside their comfort levels, they will ask for help.

<sup>2</sup> <https://peopledevelopmentmagazine.com/2021/09/18/autonomous-teams/>

Good communication skills	When teams don't communicate effectively, it's very hard to trust them, or for them to trust each other. Autonomy requires people to be able to express their needs, explain and defend their viewpoint, and give as well as receive feedback from colleagues.
Strong relationships	For a team or organization to be autonomous, they need clearly defined relationships, with defined go-to people for different types of issues. That means that you need to know each team member's strengths and weaknesses — both technically and personally.
Clear vision and requirements	Without a target or goal to work towards, the relationships and communication don't really help you get closer to it. An autonomous team needs to be very clear on what they are working towards. No ambiguity or disagreement.
High confidence, low arrogance	Confidence is the key to autonomy. Each team member needs to know what to do, who to ask for help or support, and how to do their work. When you don't know all the answers, you need the confidence to ask for help, and the humility to offer help. When you've made a mistake, you need the confidence to admit it and take ownership of the consequences. A functional team admits when they're wrong and apologize.

In essence, when you have clarity of roles, respect, and trusting relationships, you have autonomy. Autonomous teams do not mean everyone decides what they want to do, when. It also doesn't mean that you don't need a leader.

### **Decentralized decision-making is faster**

An autonomous team will be able to make decisions on its own much more often than a dependent team, resulting in shorter decision-making time. This is called 'decentralized decision-making' and is a starting point of the agile culture and mindset. You can find this in many Agile frameworks such as SAFe, LeSS and AgilePgM.

For an individual team that can create and maintain a product or service on its own, it is quite possible to achieve a far-reaching degree of autonomy. However, in order to keep it workable and not get bogged down in the many lines of communication between them, the team should not become too big. A team size of six to nine people is recommended<sup>3</sup>. But what if more people are needed to get the job done?

---

<sup>3</sup> See also Jeff Bezos' rule: every internal team should be small enough that it can be fed with two pizzas.

## A team of teams

If the product or service is too big for one team, you can work with a 'team of teams'. For example, think of five teams of six to nine team members each who work together on the product or service. Such a 'team of teams' works together on the product or service. However, this set-up has a negative impact on the autonomy of the individual teams: working with more people and teams means making more appointments. And making more agreements means less independence.

## Making agreements

But agreements are necessary. After all, in a 'team of teams', those involved need to make agreements on:

Agreements on, among other things	Explanation
The organization of product ownership	Is there one product owner for all teams as, for example, in LeSS? Or is there a hierarchy of product owners as, for example, in SAFe?
The product backlog	Is there one Product Backlog for all teams as, for example, in LeSS? Or is a hierarchy of backlogs used, as in SAFe for example? SAFe has a backlog with features for the team of teams and stories for the individual teams.
The planning and distribution of work	Which team takes care of which features? Do the teams make the planning together as in SAFe? Or is this done with representatives of the teams as in LeSS?
Enterprise architecture model	The business architecture model to be used as a precondition within which all teams must operate.
Sub-product dependencies between teams	A logical dependency, for example, is that team B can only deliver a functionality after team A has delivered something first. This is different from a logistical dependency. In the latter case, team A waits for a contribution from someone from team B, who also must carry out work within team A. This is not in line with the agile philosophy, which puts focus central; participating in two teams reduces this focus.

The sprint length and the moment of starting the sprint

These agreements are necessary to integrate and demonstrate the sub-products of the various teams at the same time.

## In short

In short, is an agile team always autonomous? After all, being able to work in an independent, autonomous team offers more agility than working in a 'team of teams'. But it is not always possible to work autonomously. There are, however, more and more organizations that create the preconditions for teams to work autonomously. They do this, for example, by redesigning the application architecture based on micro-services. These micro-services represent core functions that can be built and implemented independently. Each team develops and maintains its own micro-services. In this way, a team can act more self-sufficiently. In the analogy, you replace intersections with roundabouts. Although this can be complex and time-consuming, especially for organizations with outdated systems, it is worthwhile. In this way, more teams can work more autonomously and thus increase their agility.

## Sensemaking in the Agile Forest series

This article is part of a series of articles called *Sensemaking in the Agile Forest*. This series<sup>4</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](#)
- Is an agile team always autonomous?
- What do iterative and incremental mean in Agile?
- The Minimum Viable Product (MVP) unraveled
- Prioritizing in an agile team (MoSCoW, WSJF)
- Multitasking, task switching or monotasking
- Being predictable as an agile team (story points, velocity, t-shirt sizing, flying fingers)
- Self-managing or self-organizing agile teams
- Slicing user stories

---

<sup>4</sup> This series is based on a number of short blogs I made for Forsa Advies, a project management training organization in the Netherlands (<https://www.forsa-advies.nl>).

- 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.

---

## 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).