

The background of the slide is a grayscale aerial photograph of the University of Porto's campus, showing numerous modern and traditional buildings, green spaces, and infrastructure.

M.EIC, 2022-23

# Large Scale Software Development

Filipe Correia, Daniel Pinho, João Pedro Dias



# Project Information

Projecto

# Next two weeks

- Progressive refinement of the project descriptions into user stories.
  - > Identifying great groups of functionalities
  - > Finding a domain model
  - > Writing them as user stories (focus on the most important ones first)
- Kickoff meetings with POs
  - > For most, in the next class
  - > Classes *IMEIC03* and *IMEIC04* do it in the week after
- Experimenting with technology
- ...

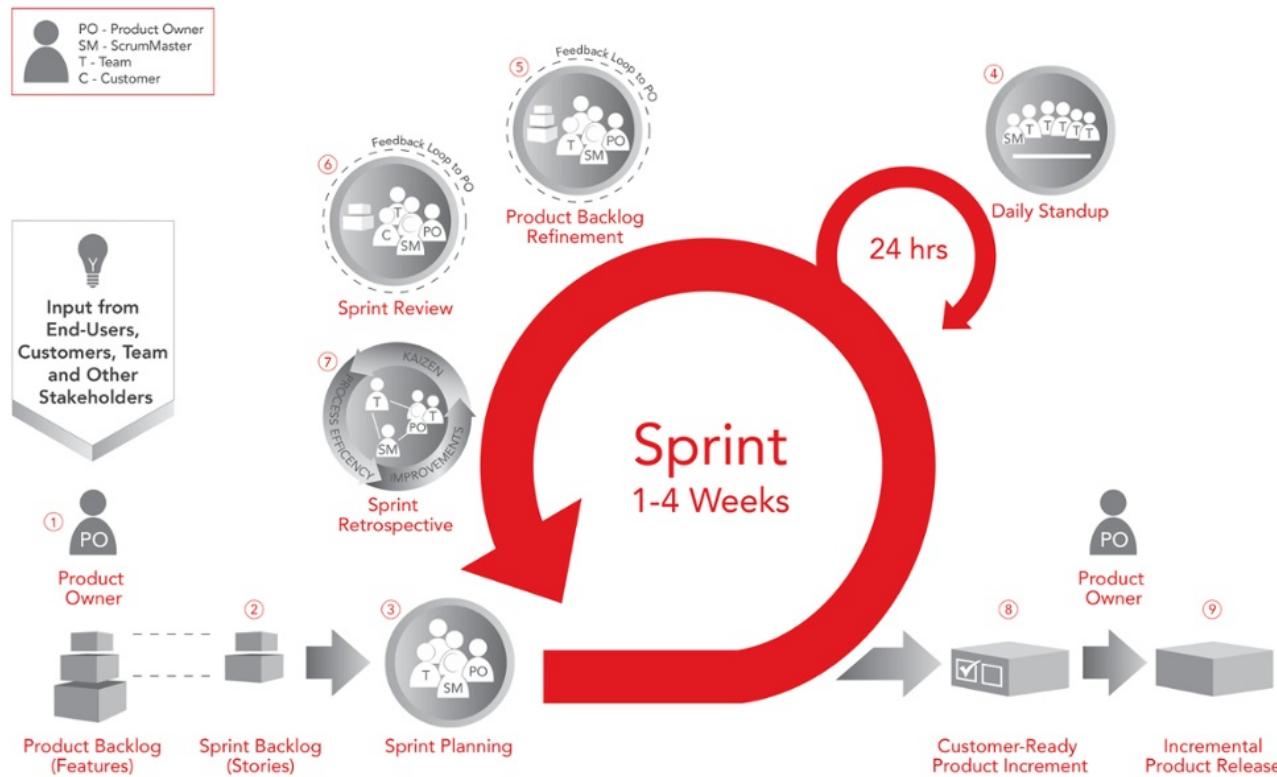
# Sprint 0

Sprint 0 (4 weeks)		15%
<b>Setup</b>	Git repository and board are created (GitHub). Project docs created and available on the repository(ies) according to templates. README.md on the root of the repository. CHANGELOG.md on the root of the repository. The SPO and SM roles are assigned to team members. There is a "factsheets/teamX.md" for each team. There is a "factsheets/firstname_lastname.md" for each team member. Team collaboration tools setup (e.g., slack, discord).	5%
<b>Initial Product Vision</b>	Kickoff meeting with PO. Product vision refined. Market research (survey of similar projects and comparative analysis). Domain analysis (includes high-level class diagram with key domain concepts).	25%
<b>Product Backlog</b>	Scrum Board has draft of Product Backlog Items (PBIs) ready for Sprint 1. PBIs have mockups (when applicable). PBIs have acceptance tests (not automated yet) covering key usage scenarios. PBIs validated by the PO (including stories, mockups, and acceptance tests). PBIs ordered by the PO.	25%
<b>Initial Technical Vision</b>	First architectural definition (component/deployment diagrams & using packages if needed). Identify main risks and justify choices showing the soundness of the technical vision. Choose development tools/environments/languages/frameworks. Learn and practice the technologies to be used; use what you learn to justify your choices. Development of a base implementation and guidelines to be used by the different teams.	45%

# Scrum Patterns

A set of small, semi-transparent navigation icons located in the bottom right corner of the slide. They include symbols for back, forward, search, and other presentation controls.

# Scrum





Jeff Sutherland  
James O. Coplien  
The Scrum Patterns Group  
*edited by Adaobi Obi Tulton*

<https://pragprog.com/book/jcscrum/a-scrum-book>

**A Scrum Book**  
The Spirit of the Game

Jeff Sutherland  
James O. Coplien  
Lachlan Heasman  
Mark den Hollander  
Cesário Ramos

and The Scrum Patterns Group:

Esther Vervloed, Neil Harrison, Kiro Harada, Joseph Yoder,  
June Kim, Alan O'Callaghan, Mike Beedle, Gertrud Bjørnvig,  
Dina Friis, Ville Reijonen, Gabrielle Benefield, Jens Østergaard,  
Veli-Pekka Eloranta, Evan Leonard, and Ademar Aguiar

[The Pragmatic Bookshelf](#)  
Raleigh, North Carolina

# ScrumPLoP, since 2010



# 94 patterns within 2 pattern languages

## Product Organization Pattern Language & Value Stream Pattern Language

- 1 The Spirit of the Game
- 2 The Mist
- 3 Fertile Soil
- 4 Conway's Law
- 5 Birds of a Feather
- 6 Involve the Managers
- 7 Scrum Team
- 8 Collocated Team
- 9 Small Teams
- 10 Cross-Functional Team
- 11 Product Owner**
- 12 Product Owner Team
- 13 Development Partnership**
- 14 Development Team
- 15 Stable Teams
- 16 Autonomous Team
- 17 Self-Organizing Team
- 18 Mitosis**
- 19 Scrum Master**
- 20 Oyatsu Jinja (おやつ神社)
- 21 Small Red Phone
- 22 Scrum (Master) Coach
- 23 Fixed Work
- 24 Sprint Planning**
- 25 Swarming: One-Piece Continuous Flow**
- 26 Kaizen Pulse
- 27 Remove the Shade
- 28 Pop the Happy Bubble
- 29 Daily Scrum**
- 30 Scrum Master Incognito
- 31 Norms of Conduct
- 32 Emergency Procedure

- 33 Illegitimus Non Interruptus
- 34 Scrum of Scrums**
- 35 Sprint Review
- 36 Sprint Retrospective**
- 37 MetaScrum
- 38 Product Pride
- 39 Vision**
- 40 Impediment List
- 41 Value Stream**
- 42 Set-Based Design
- 43 Sprint Burndown Chart
- 44 Scrum Board
- 45 Product Roadmap**
- 46 Sprint**
- 47 Organizational Sprint Pulse
- 48 Release Plan
- 49 Release Range
- 50 ROI-Ordered Backlog
- 51 High Value First
- 52 Change for Free
- 53 Money for Nothing
- 54 Product Backlog**
- 55 Product Backlog Item
- 56 Information Radiator**
- 57 Pigs Estimate
- 58 Small Items
- 59 Granularity Gradient
- 60 Estimation Points
- 61 Fixed-Date PBI
- 62 Vacation PBI
- 63 Enabling Specification
- 64 Refined Product Backlog

- 65 Definition of Ready**
- 66 Yesterday's Weather
- 67 Running Average Velocity
- 68 Aggregate Velocity
- 69 Specialized Velocities
- 70 Updated Velocity
- 71 Sprint Goal**
- 72 Sprint Backlog**
- 73 Sprint Backlog Item
- 74 Teams that Finish Early Accelerate Faster
- 75 Production Episode
- 76 Developer-Ordered Work Plan
- 77 Follow the Moon
- 78 Visible Status
- 79 Dependencies First
- 80 Good Housekeeping
- 81 Whack the Mole
- 82 Definition of Done
- 83 Team Sprint
- 84 Responsive Deployment
- 85 Regular Product Increment**
- 86 Release Staging Layers
- 87 Testable Improvements
- 88 One Step at a Time
- 89 Value Areas
- 90 Value Stream Fork
- 91 Happiness Metric
- 92 Scrumming the Scrum
- 93 Greatest Value**
- 94 Product Wake

# Why we need Scrum Patterns?

- Finding **where to start** implementing Scrum can be a *mystery*, and finding where to improve can be a *challenge*.
- The book offers **proven solutions** called *patterns* that the authors have **distilled while observing** many Scrum Teams — both their successes and failures.
- These solutions will help you implement and improve your use of Scrum whether you are a *beginner* or an *experienced* practitioner.
- The book **stands on Scrum's deepest foundations** and **reflects contributions** from many early shapers of Scrum, including one of its inventors.

*adapted from "Introduction" at scrumlop.org*

**name**

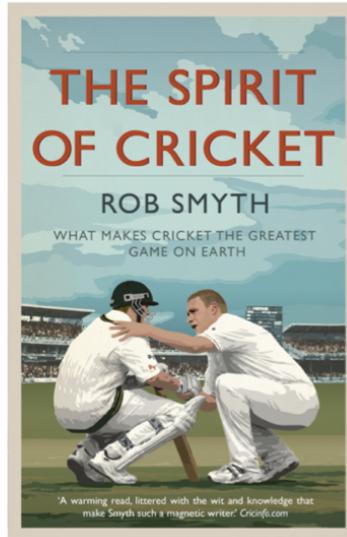
¶1 The Spirit of the Game • 3

**context**

**problem**

## ¶1 The Spirit of the Game

Confidence stars: \*\*



... the Scrum framework does not have all the answers, which means that the team cannot look to Scrum for direction when it does not give a final answer.

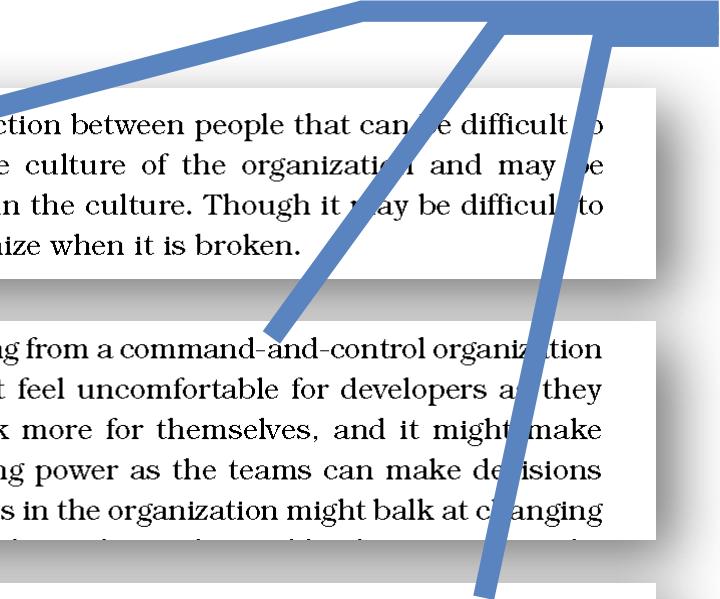


Written rules might give concrete guidance for how to work together, but *spirit* is part of culture that guides interactions and may be discerned only when ignored or violated.

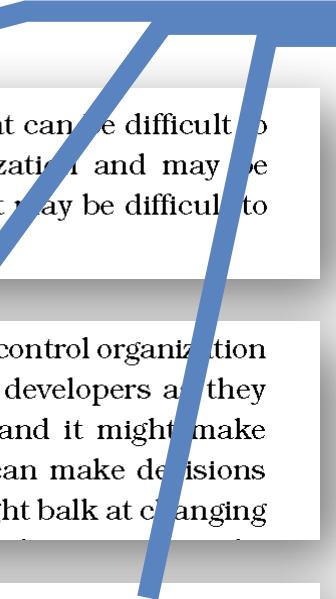
Cricket is a game that owes much of its unique appeal to the fact that it

## discussion

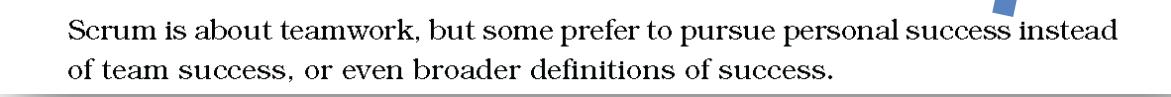
## forces



Scrum requires a spirit of interaction between people that can be difficult to define. This spirit is part of the culture of the organization and may be indiscernible for the people within the culture. Though it may be difficult to define the spirit is easy to recognize when it is broken.



Changing habits is difficult. Moving from a command-and-control organization to [I16 Autonomous Teams](#) might feel uncomfortable for developers as they are faced with the need to think more for themselves, and it might make managers feel that they are losing power as the teams can make decisions without their approval. Individuals in the organization might balk at changing



Scrum is about teamwork, but some prefer to pursue personal success instead of team success, or even broader definitions of success.

## solution

*Therefore:*

When using Scrum the product community must focus on explicitly creating a culture in the organization where people know and follow the spirit of Scrum.

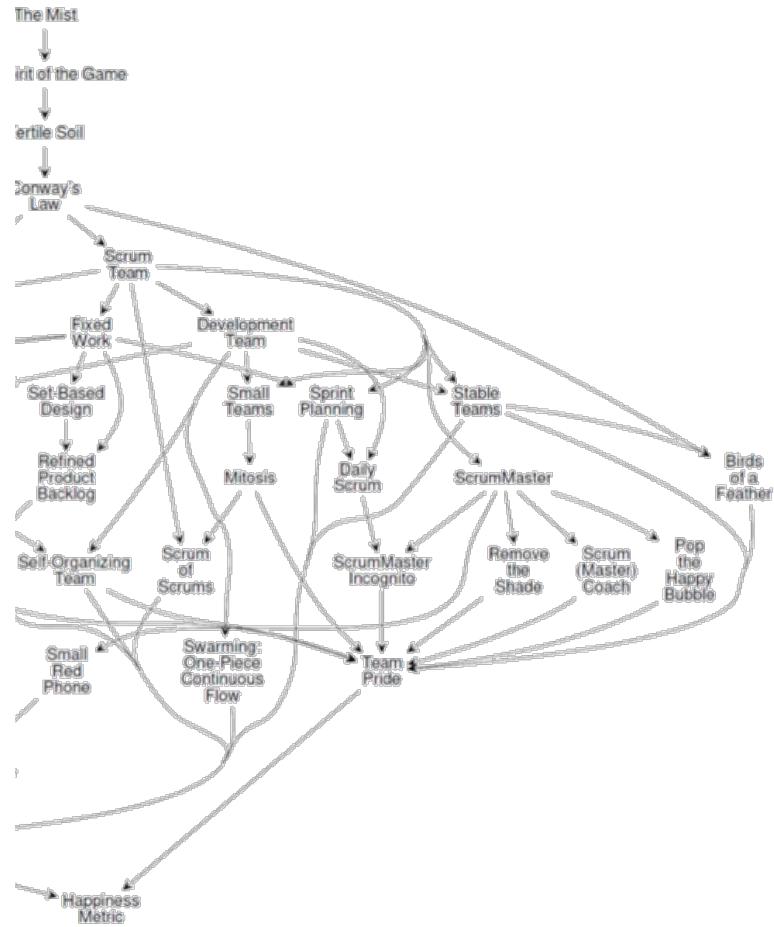
## resulting context

within the spirit of Scrum. When starting to use the Scrum framework the team will find it challenging to work within the spirit. It will feel uncomfortable for people, and will be arduous — again, it's about habits. To overcome this challenge it is essential to start with good [#19 ScrumMasters](#) and [#11 Product Owners](#), and it is necessary for everyone on the team to support each other to work within the spirit. A new culture emerges, where the spirit will be inherent in the ways of working and interacting.

In Scrum and Cricket there are clear rules for the game; in both, it is essential that the spirit is a guide for the people using these rules.

# Product Organization Pattern Language

37 patterns + 2 sequences to build your Scrum Team and other associations of people that are common to Scrum practice.



# Product Organization Sequence

¶1 THE SPIRIT OF THE GAME

¶2 THE MIST

¶95 COMMUNITY OF TRUST

¶3 FERTILE SOIL

¶4 CONWAY'S LAW

¶5 BIRDS OF A FEATHER

¶6 INVOLVE THE MANAGERS

¶7 SCRUM TEAM

¶11 PRODUCT OWNER

¶13 DEVELOPMENT PARTNERSHIP

¶14 DEVELOPMENT TEAM

¶19 SCRUM MASTER

¶24 SPRINT PLANNING

¶25 SWARMING: ONE-PIECE  
CONTINUOUS FLOW

¶29 DAILY SCRUM

¶32 EMERGENCY PROCEDURE

¶33 ILLEGITIMUS NON INTERRUPTUS

¶34 SCRUM OF SCRUMS \*

¶35 SPRINT REVIEW

¶36 SPRINT RETROSPECTIVE

¶37 METASCRUM

¶38 PRODUCT PRIDE

¶118 TEAM PRIDE

# ¶3 Fertile Soil



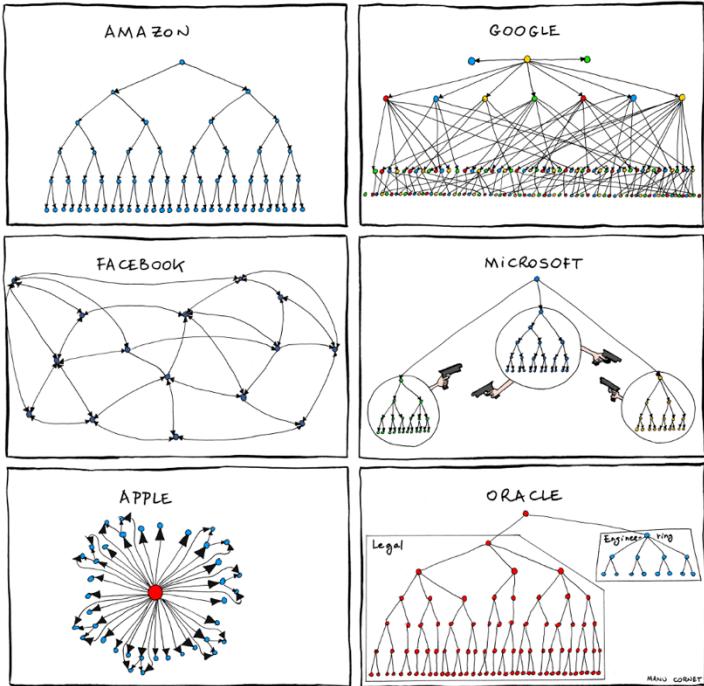
## ¶3 Fertile Soil

It is the moment by moment interactions of people working together on a product that build and sustain product organizations. Interaction qualities both reflect and define organization qualities.

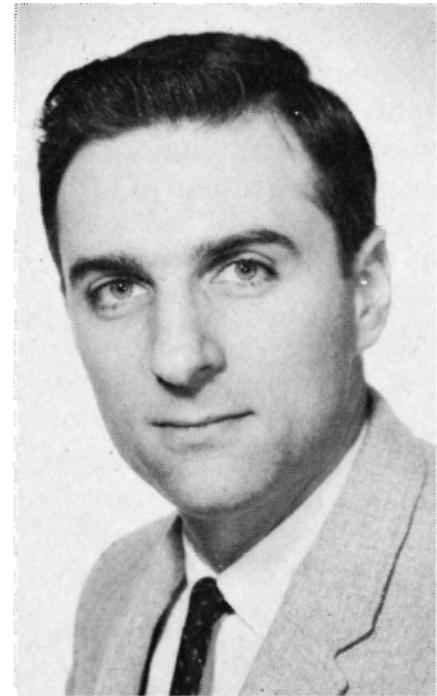
*Therefore:*

Demonstrate the values of **Commitment, Focus, Openness, Respect** and **Courage** in your day to day behaviors and interactions [...]. This helps create a virtuous circle that supports **transparency**, and that makes it possible to build on the **inspection and adaptation** at the core of effective Scrum efforts.

# ¶4 Conway's Law



<https://bonkersworld.net/organizational-charts>



## ¶4 Conway's Law

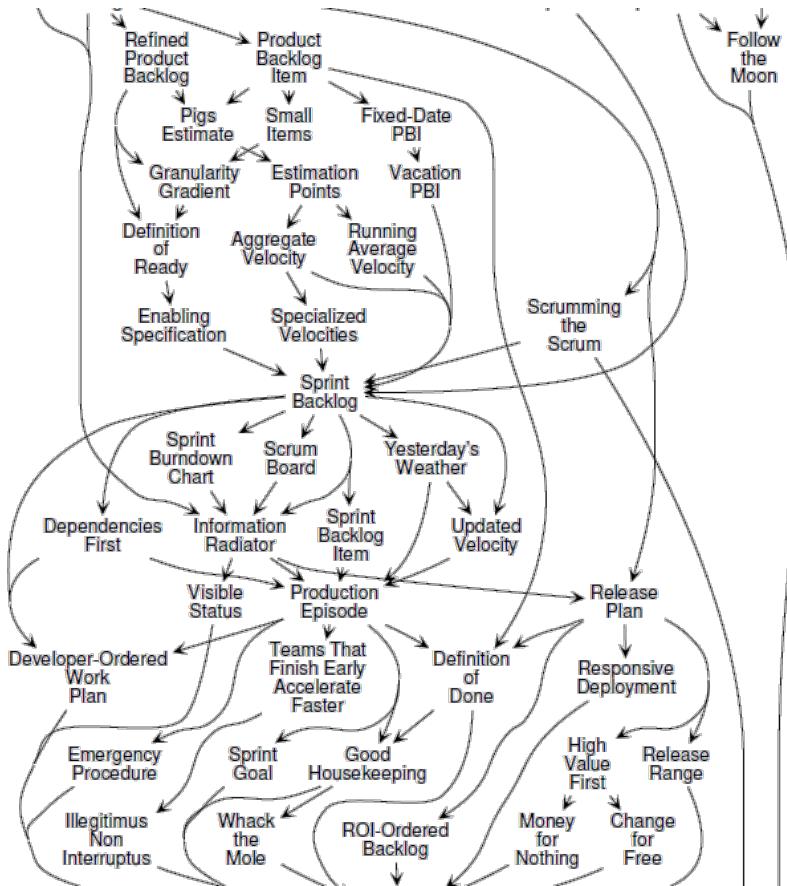
Effective communication and feedback are at the heart of effective complex system development, and **the organization structure should be optimized for the most crucial paths of communication**. Communication and feedback, together with self-organization, are the agile heart.

*Therefore:*

**Organize** the work force into ***Small Teams*** of more or less **five people**, **partitioned** according to the most important **concerns** for the creation of **value** by the enterprise. Supplement this structure with a small number of crosscutting structures for secondary but important concerns, never forgetting that these structures are only optimizations in what is otherwise an open environment of unconstrained cooperation.

# Value Stream Pattern Language

56 patterns + 2 sequences to build relationships between steps of product construction and the artifacts that represent parts of the process.



# Value Stream Sequence

¶2 THE MIST

¶39 VISION

¶41 VALUE STREAM

¶45 PRODUCT ROADMAP

¶46 SPRINT

¶54 PRODUCT BACKLOG

¶55 PRODUCT BACKLOG ITEM

¶71 SPRINT GOAL

¶72 SPRINT BACKLOG

¶75 PRODUCTION EPISODE

¶84 RESPONSIVE DEPLOYMENT

¶35 SPRINT REVIEW

¶36 SPRINT RETROSPECTIVE \*

¶85 REGULAR PRODUCT INCREMENT

¶86 RELEASE STAGING LAYERS

¶89 VALUE AREAS

¶90 VALUE STREAM FORK

¶93 GREATEST VALUE

¶94 PRODUCT WAKE

¶39 Vision



# ¶39 Vision

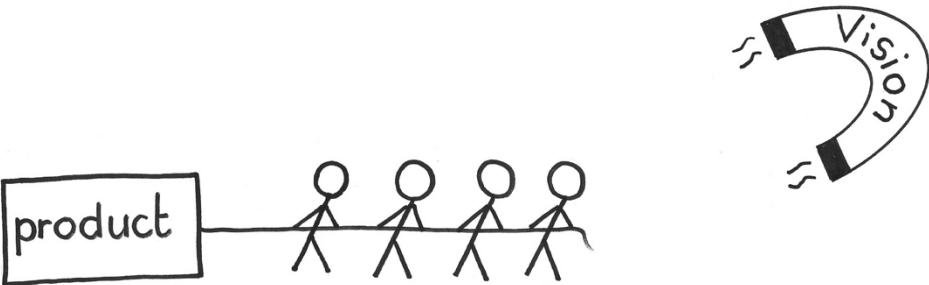
People **thrive within constraints** that channel their creativity and work towards a common good, but it's also true that **overly specific constraints** can turn contributors into subservient **robots who blindly follow orders** rather than following with their heart.

*Therefore:*

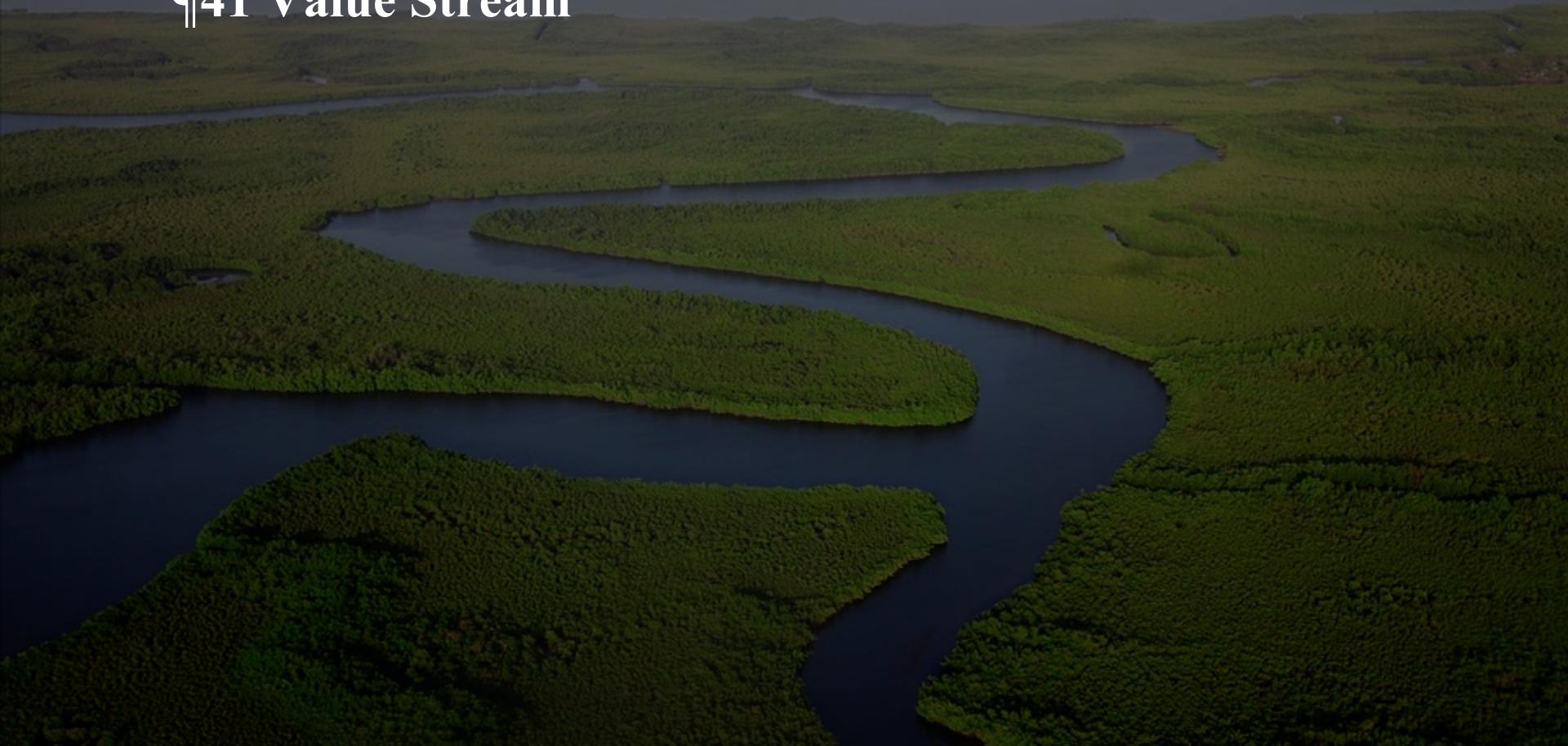
The **individual** who embodies the **passion for this new product** effort takes on the **role of ¶11 Product Owner**, around whom stakeholders and potential future coworkers rally to articulate and together to define and refine a *Vision*.

# ¶39 Vision

The *Vision* is a description of how the product supports a desired future towards which an envisioned future Product Organization Pattern Language advances.



# ¶41 Value Stream



# ¶41 Value Stream

The **development process** and the **path from conception to market** are as important to product success as the product idea itself.

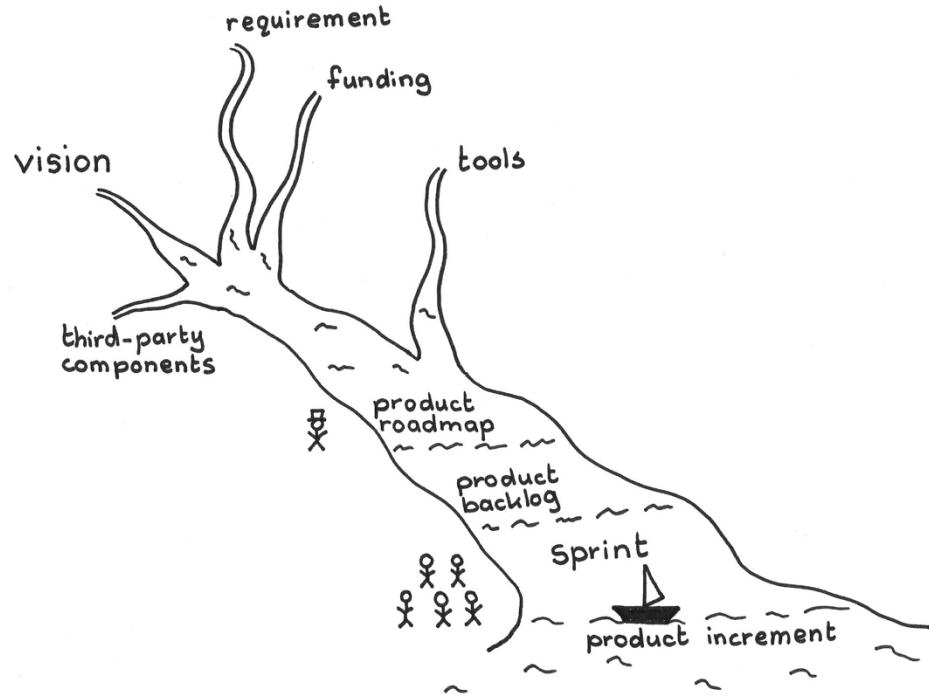
*Therefore:*

The *Product Owner* creates an ecosystem whose elements build on each other to deliver ever-increasing value in an evolving product. At the center of this ecosystem, there is a **process to deliver ongoing and evolving streams of product increments** to stakeholders: the *Value Stream*.

# ¶41 Value Stream

The building blocks include the artifacts (backlogs, product components) and the processes that guide and coordinate the creation of these artifacts (through events and joint work).

The people build the processes that guide the creation of the artifacts, themselves enact them, and are instrumental in evolving them.



# ¶54 Product Backlog



# ¶54 Product Backlog

At any given time, it is important that the whole team **is aligned about what they need to deliver next**, and that the direction be transparent. The *Development Team* can't do everything at once — in fact, you can't even do two things well at the same time. It's important to maintain focus.

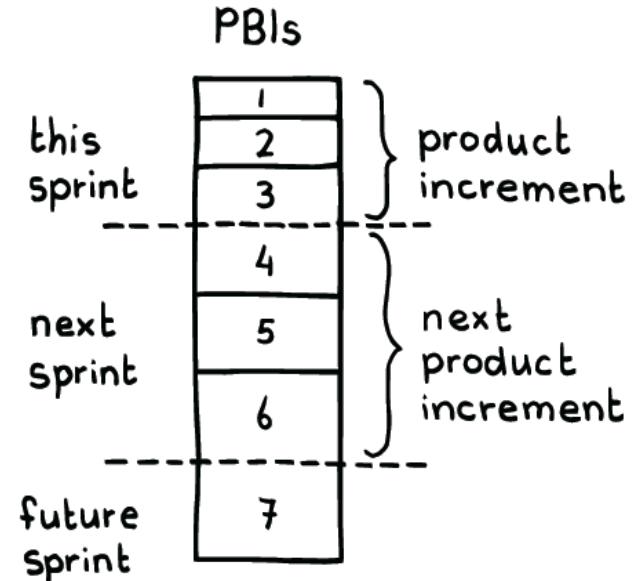
*Therefore:*

For each product, **create a single ordered list** called the *Product Backlog* — a list of ¶85 *Product Increment* contributions called ¶55 *Product Backlog Items (PBIs)*, arranged in delivery order.

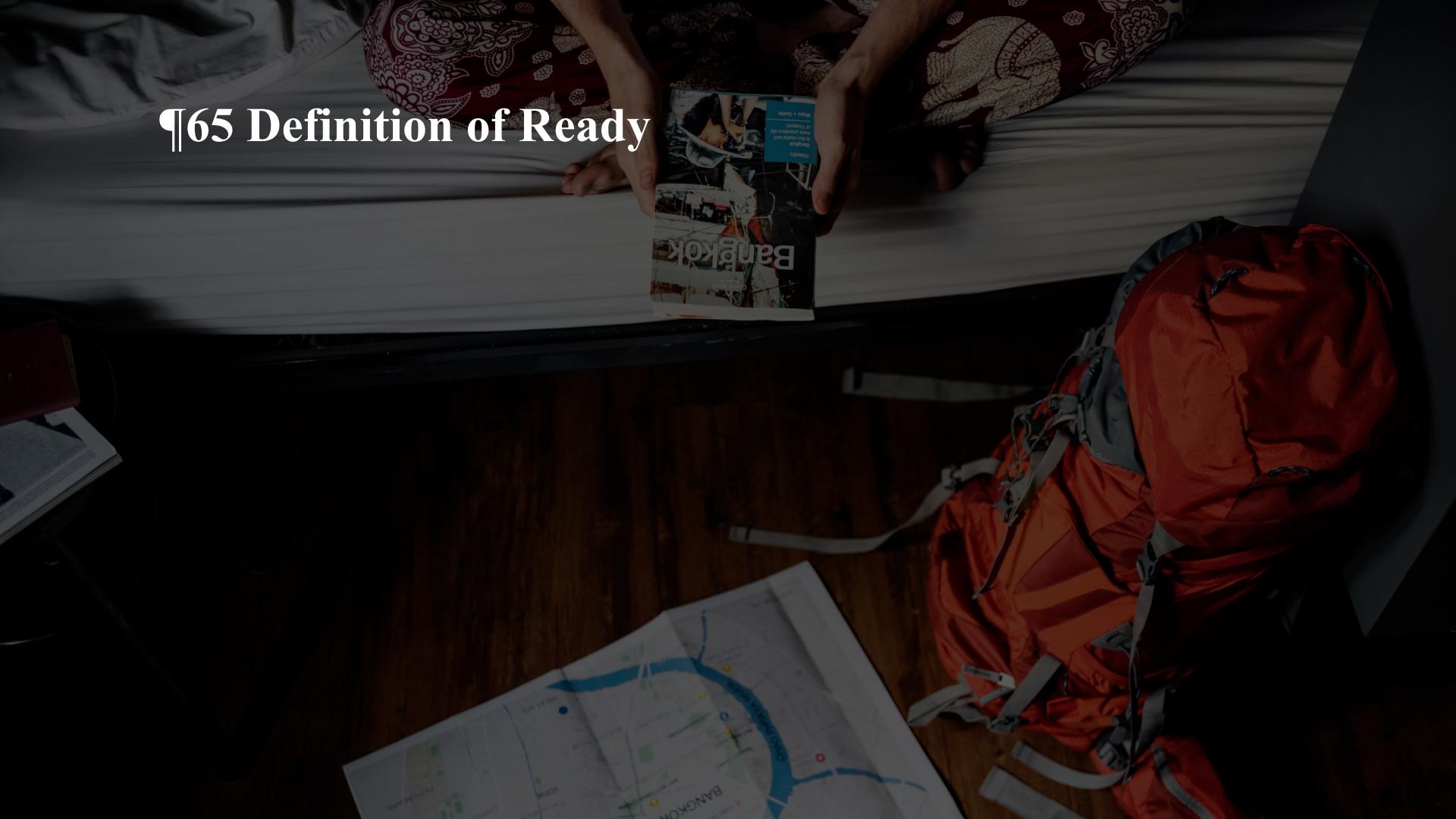
# ¶54 Product Backlog

The *Product Backlog* details the *Product Owner* vision for the product according to the expectations of all stakeholders, with each *PBI* describing a contribution to a deliverable ¶85 *Product Increment*.

The *Product Owner* has final authority over the content of the *Product Backlog*; however, he or she usually develops the *Product Backlog* in a joint effort with the *Development Team* during regular events convened to maintain a ¶64 *Refined Product Backlog*, as well as during ¶24 *Sprint Planning*.



# ¶65 Definition of Ready



# ¶65 Definition of Ready

If the Development Team does not precisely understand ¶55 *Product Backlog Items (PBI)*, development effort (and time) tend to balloon, which in turn cause the Sprint to miss the Sprint Goal or to not deliver what stakeholders expect.

*Therefore:*

Each ***Product Backlog Item*** must meet at least the following criteria before the Development Team can take it as a candidate for the work on the Sprint Backlog during ¶24 *Sprint Planning*:

1. The **work** is immediately **actionable** by the team.
2. The planned **deliverable has value**.
3. The ¶11 **Product Owner** and the **Development Team** have discussed it.
4. The Development Team has **estimated** it.
5. It is testable, and the **Product Owner has defined tests** for it.
6. The ¶7 **Scrum Team** has **sized the pieces appropriately** (see ¶58 *Small Items*).

# ¶85 Regular Product Increment



# ¶85 Regular Product Increment

It is often very difficult to validate if the team has created value in every ¶46 Sprint. However, the **Product Owner wants to be sure that the product increases value, Sprint after Sprint.**

*Therefore:*

Every Sprint the *Scrum Team creates a **Product Increment** that is **Done*** (see ¶82 *Definition of Done*), **usable** and potentially **releasable**. The team uses the *Product Increment* to validate if they have increased the value of the product, and to understand how the product actually performs in the market. In the long term the end users will be happier, and current use can hone foresight that can help the team head off many future risks.

# ¶46 Sprint



# ¶46 Sprint

The most fundamental human processes build on cadence.

Human culture often realizes cycles in rituals and other visible events of its calendar.

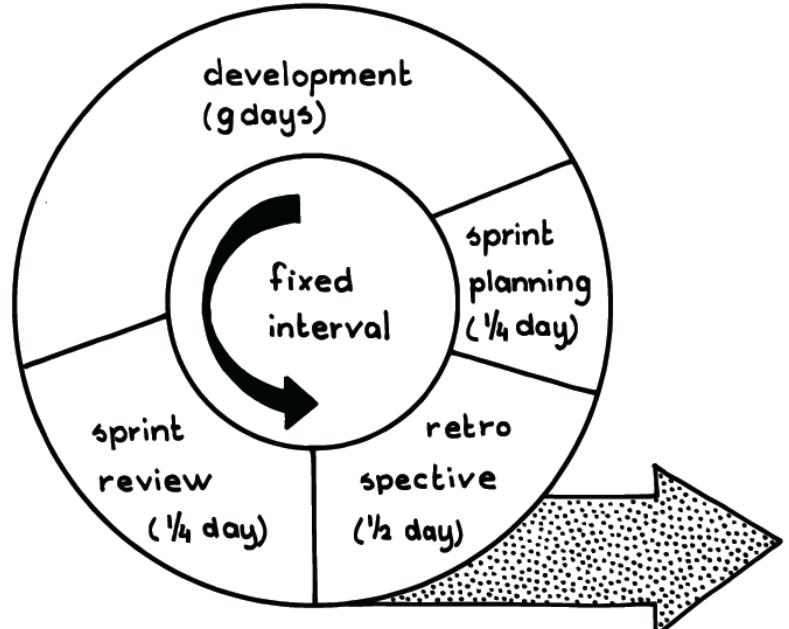
*Therefore:*

**Organize development around recurring, frequent, fixed-length time boxed intervals** called *Sprints*. The *Sprint* is both a single time boxed period of product *delivery effort* (duration) as well as a unit demarcating *delivery interval* on the release calendar (cadence).

# ¶46 Sprint

We can regard a *Sprint* in three ways: as a unit of time, as a unit of *Regular Product Increment*, and as a unit of learning.

It is first a unit of *time*: the implementation of regular delivery that we find in *Regular Product Increment*.



# ¶71 Sprint Backlog



# ¶71 Sprint Goal



# ¶71 Sprint Goal

The objective of a *Sprint* is to deliver value to the stakeholders. However, simply following a list of ¶73 *Sprint Backlog Items (SBIs*; e.g., tasks) does not necessarily result in the creation of the greatest value possible.

*Therefore:*

The ¶7 *Scrum Team* commits to a **short statement** of the **value** that they intend to **create during** the *Sprint*. This becomes the focus of all work in the *Sprint*.

# ¶57 Pigs Estimate



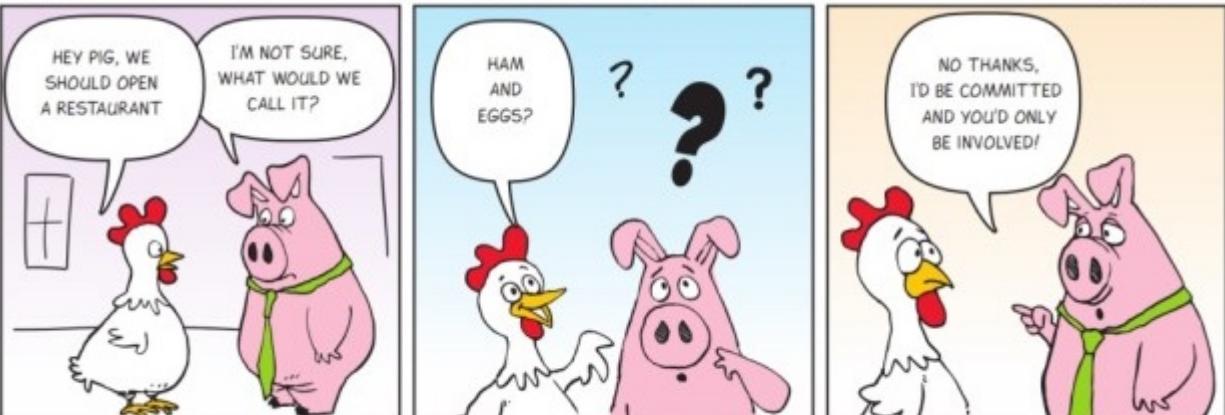
Gelukkig Nieuwjaar

# ¶57 Pigs Estimate

The team should ground its estimates in reality rather than assumptions or wishful thinking.

*Therefore:*

**Let the people who are committed to do the actual work do the estimation.** In the Scrum sense, it is pigs that estimate—not chickens ([1], p. 31; [2], p. 51; and [3], p. 123).



© Jake Calabrese at Helping Improve

(Inspired in [5], p. 42)

HelpingImprove.com/agile-commitment-scrum-pig-chicken-part-1

- [1] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum* (Series in Agile Software Development). London: Pearson, Oct 21, 2001, p. 35.
- [2] Mike Cohn. *Agile Estimating and Planning*, 1 edition. Englewood Cliffs, NJ: Prentice-Hall, 2005, p. 51.
- [3] Kenneth S. Rubin. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Reading, MA: Addison-Wesley Signature Series (Cohn), Aug. 5, 2012, p. 123.
- [4] Magne Jørgensen. ‘What we Know about Software Development Effort Estimation.’ In *IEEE Software* 31(2), March/April 2014, pp. 37–40.
- [5] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum* (Series in Agile Software Development). London: Pearson, Oct. 2001, p. 42.

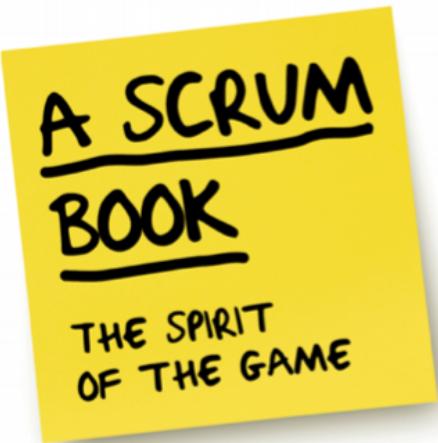
# 94 patterns within 2 pattern languages

## Product Organization Pattern Language & Value Stream Pattern Language

- 1 The Spirit of the Game
- 2 The Mist
- 3 Fertile Soil
- 4 Conway's Law
- 5 Birds of a Feather
- 6 Involve the Managers
- 7 Scrum Team
- 8 Collocated Team
- 9 Small Teams
- 10 Cross-Functional Team
- 11 Product Owner**
- 12 Product Owner Team
- 13 Development Partnership**
- 14 Development Team
- 15 Stable Teams
- 16 Autonomous Team
- 17 Self-Organizing Team
- 18 Mitosis**
- 19 Scrum Master**
- 20 Oyatsu Jinja (おやつ神社)
- 21 Small Red Phone
- 22 Scrum (Master) Coach
- 23 Fixed Work
- 24 Sprint Planning**
- 25 Swarming: One-Piece Continuous Flow**
- 26 Kaizen Pulse
- 27 Remove the Shade
- 28 Pop the Happy Bubble
- 29 Daily Scrum**
- 30 Scrum Master Incognito
- 31 Norms of Conduct
- 32 Emergency Procedure

- 33 Illegitimus Non Interruptus
- 34 Scrum of Scrums**
- 35 Sprint Review
- 36 Sprint Retrospective**
- 37 MetaScrum
- 38 Product Pride
- 39 Vision**
- 40 Impediment List
- 41 Value Stream**
- 42 Set-Based Design
- 43 Sprint Burndown Chart
- 44 Scrum Board
- 45 Product Roadmap**
- 46 Sprint**
- 47 Organizational Sprint Pulse
- 48 Release Plan
- 49 Release Range
- 50 ROI-Ordered Backlog
- 51 High Value First
- 52 Change for Free
- 53 Money for Nothing
- 54 Product Backlog**
- 55 Product Backlog Item
- 56 Information Radiator**
- 57 Pig Estimate
- 58 Small Items
- 59 Granularity Gradient
- 60 Estimation Points
- 61 Fixed-Date PBI
- 62 Vacation PBI
- 63 Enabling Specification
- 64 Refined Product Backlog

- 65 Definition of Ready**
- 66 Yesterday's Weather
- 67 Running Average Velocity
- 68 Aggregate Velocity
- 69 Specialized Velocities
- 70 Updated Velocity
- 71 Sprint Goal**
- 72 Sprint Backlog**
- 73 Sprint Backlog Item
- 74 Teams that Finish Early Accelerate Faster
- 75 Production Episode
- 76 Developer-Ordered Work Plan
- 77 Follow the Moon
- 78 Visible Status
- 79 Dependencies First
- 80 Good Housekeeping
- 81 Whack the Mole
- 82 Definition of Done
- 83 Team Sprint
- 84 Responsive Deployment
- 85 Regular Product Increment**
- 86 Release Staging Layers
- 87 Testable Improvements
- 88 One Step at a Time
- 89 Value Areas
- 90 Value Stream Fork
- 91 Happiness Metric
- 92 Scrumming the Scrum
- 93 Greatest Value**
- 94 Product Wake



Jeff Sutherland  
James O. Coplien  
The Scrum Patterns Group  
*edited by Adaobi Obi Tulton*

<https://pragprog.com/book/jcscrum/a-scrum-book>

**A Scrum Book**  
The Spirit of the Game

Jeff Sutherland  
James O. Coplien  
Lachlan Heasman  
Mark den Hollander  
Cesário Ramos

and The Scrum Patterns Group:

Esther Vervloed, Neil Harrison, Kiro Harada, Joseph Yoder,  
June Kim, Alan O'Callaghan, Mike Beedle, Gertrud Bjørnvig,  
Dina Friis, Ville Reijonen, Gabrielle Benefield, Jens Østergaard,  
Veli-Pekka Eloranta, Evan Leonard, and Ademar Aguiar

[The Pragmatic Bookshelf](#)  
Raleigh, North Carolina

# References

- The ScrumPLoP site, The Scrum Patterns Group, <http://www.scrumplop.org>, 2010-2017.
- The Scrum Guide, Jeff Sutherland and Ken Schwaber, <http://scrumguides.org>, 2017.
- Scrum: The Art of Doing Twice the Work in Half the Time, Jeff Sutherland, 2014.
- Secrets of Scrum, Gertrud Bjørnvig & James Coplien, slideshare, 2014.
- Alexander, A Pattern Language, Oxford, 1977.

# FEUP.DEI



DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA  
[dei.fe.up.pt](http://dei.fe.up.pt) | [secdei@fe.up.pt](mailto:secdei@fe.up.pt) | +351 225 082 134  
Director **João MP Cardoso** | Full Professor  
[jmpc@fe.up.pt](mailto:jmpc@fe.up.pt) | +351 220 413 284

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO  
Rua Dr. Roberto Frias | 4200-465 Porto | PORTUGAL  
[www.fe.up.pt](http://www.fe.up.pt) | [feup@fe.up.pt](mailto:feup@fe.up.pt) | +351 225 081 400  
 @DEI\_FEUP