

Extracted from:

# A Scrum Book

The Spirit of the Game

This PDF file contains pages extracted from *A Scrum Book*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2019 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

A SCRUM  
BOOK

THE SPIRIT  
OF THE GAME

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

# A Scrum Book

The Spirit of the Game

Jeff Sutherland

James O. Coplien

Lachlan Heasman

Mark den Hollander

Cesário Oliveira 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



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt  
VP of Operations: Janet Furlow  
Managing Editor: Susan Conant  
Development Editor: Adaobi Obi Tulton  
Copy Editor: Sean Dennis  
Indexing: Potomac Indexing, LLC  
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact [support@pragprog.com](mailto:support@pragprog.com).

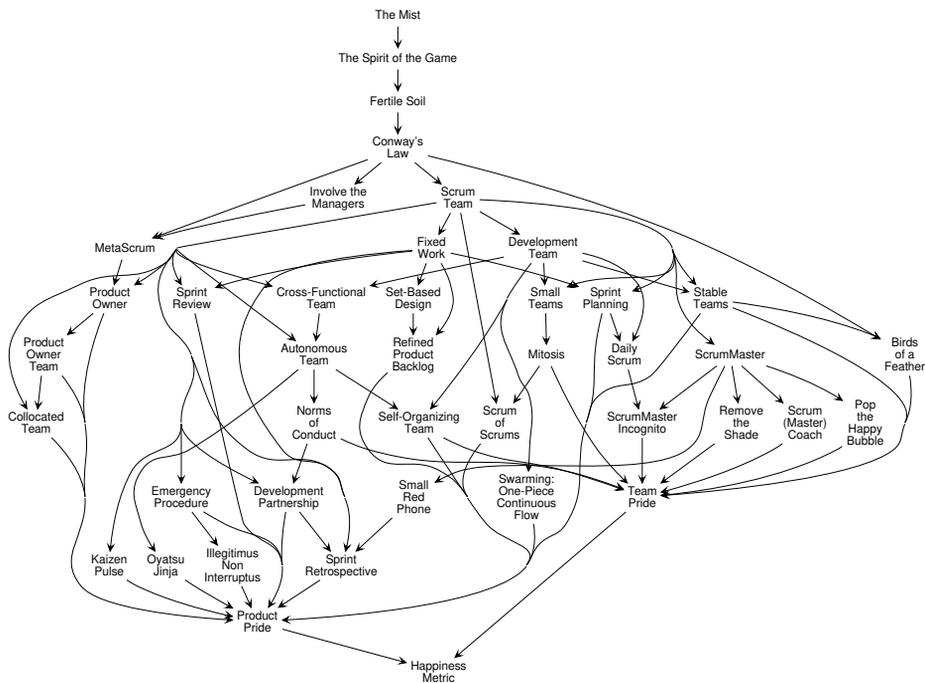
For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2019 The Pragmatic Programmers, LLC.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-671-6  
Book version: P1.0—August 2019

# Product Organization Pattern Language



This chapter describes a set of patterns you can use to build your [17 Scrum Team](#) and the other associations of people and loci of relationships that are common to Scrum practice. The Product Organization is one major Whole to which you must attend when introducing Scrum; the other one is the [Value Stream Pattern Language on page ?](#). You build your organization and your process in parallel, each a little at a time, weaving together patterns from both chapters.

The preceding figure shows the relationships between the patterns in this chapter. We apply patterns one at a time. The figure is a guide for the order in

which we should apply them. This structure is called a *pattern language*, because just as a language like English describes how we can sequence words in meaningful ways, this graph describes meaningful pattern sequences. More precisely, it is called a *generative pattern language* because it *generates* “sentences” that are combinations of patterns that powerfully reinforce each other.

Each node of the graph is a pattern, and the lines between patterns depict ordering dependencies between them. The graph therefore shows, for each pattern, the patterns above it, which you should already have considered before applying any given pattern—and the ones below it, which are candidates for next steps once you have put the current one in place. Of course, instead of just forging ahead, you can jump to another place in the graph and add a pattern that refines some other pattern you had applied earlier. One pattern is said to *refine* another (or to refine several other patterns), or the other way around, one pattern is said to *contain* one or more other patterns. So *The Spirit of the Game* refines [12 The Mist](#), and [14 Conway’s Law](#) contains *Scrum Team*.

There are, of course, almost countless ways to sequence the patterns, each sequence giving rise to a Scrum organization with a slightly different character. Though you should challenge yourself by considering most of the patterns in this book over time, you don’t have to use all of them. You will usually pick the patterns that catch your eye and then organize them into a sequence using the graph as a guide. But the dependencies aren’t strict and we encourage you to engage your insight and intuition as you apply the patterns. Of course, it is O.K. to change your mind along the way.

In the next section, we present a kind of canonical or default sequence for the major patterns in the Product Organization Pattern Language. But, again, use your insight and intuition rather than following the sequence as a mandate. You know more about your organization and your situation than we do.

## Product Organization Sequence

A *pattern* is an instruction to shape something we build to increase the Wholeness of the Whole. *Organizational patterns*—such as those in this section—are networks of people that an organization builds, restructures, or incrementally refines with an eye to overall improvement (see [Kaizen and Kaikaku on page ?](#)). The organization applies these patterns according to a *sequence*, trying only one pattern at a time to assess its contribution to organizational quality. Occasionally a pattern doesn’t work as expected, and it makes sense to try another pattern or to try patterns in a different order. Many different sequences can work, with as many or as few patterns as the organization

sees fit, applied in what the organization feels is the best order. If a pattern is like a word, the sequences are like sentences.

A *pattern language* is the set of rules for combining the patterns in meaningful orders; as a “language” it has a grammar that can generate all sequences that are meaningful “sentences.” While the number of combinations of patterns is large, not all orderings make sense. In the pattern language for a house one usually builds the foundations before the walls, and a pattern language for a house would probably *constrain* that the foundation be built before the walls.

Here we offer an archetypical sequence for the Product Organization Pattern Language. It starts with the “largest” pattern, i.e., that which metaphorically covers the largest area. Subsequent patterns subdivide that space or add small increments to it according to the precedence given in the sequence. Each pattern describes some configuration of people in relationship who have come together to pursue some interest of the organization. Some, like [¶17 Scrum Team](#), are long-term configurations of membership while others like the [¶129 Daily Scrum](#) are short-lived in any particular incarnation. You will not be able to find all the groups (configurations of people) on the organizational chart. Some, like [¶124 Sprint Planning](#), form periodically, and while there is some stability to this group’s constituency, it also has some ad hoc membership. Some may happen only occasionally or perhaps not at all, such as the association of folks to enact an [¶132 Emergency Procedure](#). Some, like [¶11 The Spirit of the Game](#) or [¶138 Product Pride](#), describe qualities taken on by some of those groups of people.

It is useful to think of the patterns as building the spaces, or the identities, for groups of people who gather occasionally or periodically to exchange ideas, build things, solve problems, and to grow together. Any single individual may have a place in many of these configurations of group interaction, though usually engaging in only one at a time, and perhaps several over the course of a day. The Product Organization Pattern Language builds the “organization chart” for these sometimes tacit and usually dynamic configurations of human interaction that each come together around some purpose.

The sequence presented below doesn’t catalog all the book’s Product Organization patterns, but rather features a core subset of patterns that give an organization its most salient features. The ensuing pages present all the patterns we have chosen to include in this pattern language. We present them more or less in the order of a meaningful sequence of unfolding.

A sequence is more structural than temporal, and it may take a small bit of mental training to think of it in other than the terms we normally use for

development processes. For example, the [¶11 Product Owner](#) is probably the first on the scene—the seed around which the organization will grow. After he or she does some up-front work, a [¶14 Development Team](#) may join, and then a [¶19 ScrumMaster](#). Thus assembled, we now have a *Scrum Team*. Yet if the initial intent is to create a *Scrum Team* and that's what we set out to do, maybe in fact it is instead seeded with members of a *Development Team* to which we later add the other roles. So if the *Product Owner* is the first on the scene, we kind of pretend that he or she is the entire, primordial *Scrum Team*, which we later differentiate by adding *ScrumMaster* and *Development Team* roles. The same slight fiction works if the effort grows out of a *Development Team*.

[¶1 The Spirit of the Game](#) The environment is animated by an agile spirit that can handle novel and unusual situations as well as respond to the continual changes facing most organizations, while working within a framework of discipline for the orderly engagement of all players.

[¶2 The Mist](#) There is a vague longing across some community which sparks attempts by groups to solve great problems locally or individually, and while aspiring to do great things, stops at small, local benefits for want of effective cooperation.

[¶95 Community of Trust on page ?](#) Team members create an environment where they consider themselves as a team (a community), with sufficient trust to feel safe interacting with each other. They take explicit, obvious, and consistent actions to demonstrate trust. They set up working procedures by mutual agreement, rather than by unilateral edicts.

[¶3 Fertile Soil](#) Interaction qualities both reflect and define organization qualities. Create cultural foundations for the values of Commitment, Focus, Openness, Respect and Courage in the teams' day-to-day behaviors and interactions.

[¶4 Conway's Law](#) Product groups form around opportunities to serve some constituency by creating a product that serves some segment of society, as a precursor to aligning the work, the organization and business, and the market.

[¶5 Birds of a Feather](#) Additional cross-cutting groups form around internal areas of interest such as technology or other core competencies that support the work.

[¶6 Involve the Managers](#) The effort may grow to the point where it can benefit from business administration support, broad coordination, and a locus

of responsibility to initiate discontinuous change. The enterprise adds a small management staff as “über ScrumMasters” who can initiate radical change, remove impediments, provide administration and coordinate from a “big picture” position.

¶17 Scrum Team The Scrum Team emerges from the broader organization: a ¶18 Collocated Team and ¶10 Cross-Functional Team that operates as a small business within the context of the organization, making independent decisions to respond to stakeholders and the market. It is brought into existence and led by the...

¶11 Product Owner who leads the newly formed team to realize his or her ¶139 Vision for creating value. The Product Owner is the single person who takes accountability for realizing the Vision, for the value that emanates from the delivery of that Vision, and for the contents of the Product Backlog. The Product Owner is the voice of value to rest of the Scrum Team.

¶113 Development Partnership The Product Owner leads an effort to forge a partnership with the constituency that will be served by the organization’s product.

¶14 Development Team The Product Owner hires a team to implement the product as a series of product increments that realize the Vision. (As previously described, by “hire” we mean to associate a committed group to the effort, whether through formal employment or some other form of organizational commitment.)

¶19 ScrumMaster The new team hires a ScrumMaster as their “servant leader” to guide the team in using the Scrum process, and in continuous improvement.

¶24 Sprint Planning The Scrum Team—the Product Owner, Development Team, and the ScrumMaster—assembles to plan a Production Episode and build a potentially shippable product increment.

¶25 Swarming: One-Piece Continuous Flow During production, the team works as one mind, minimizing inventory and work in progress by taking each Product Increment continuously through its development tasks from beginning to end.

¶29 Daily Scrum The Development Team assembles every day to replan their work and to optimize their chances of meeting the Sprint Goal.

¶32 Emergency Procedure If the team faces serious trouble that threatens the team’s ability to meet the Sprint Goal, the team assembles to discuss

escalation strategies that may entail replanning, getting outside help, decreasing the scope of the deliverable, or aborting the current Sprint and doing more fundamental replanning.

¶133 *Illegitimus Non Interruptus* Scrum Teams develop the discipline to notice when product churn falls outside statistical norms, such that they abort the Sprint, take corrective action, and replan.

¶134 *Scrum of Scrums* Each day, all the Development Teams for a given product synchronize with each other and work together to solve problems and move the teams towards the Sprint Goal.

¶135 *Sprint Review* At the end of the Sprint, the Scrum Team evaluates progress on the product; the Product Owner decides what changes to incorporate into the imminent release.

¶136 *Sprint Retrospective* As the last gathering of the Sprint, the Scrum Team assembles to contemplate how best to make incremental process improvements, and commits to one such improvement during the next Sprint.

¶137 *MetaScrum* If management is interfering with the teams, you should create a regular encounter between management and the Product Owners where management has a chance to influence product direction.

¶138 *Product Pride* The team grows to identify with the product they build and are proud to be in the business of producing it.

¶118 *Team Pride on page ?* The team has a “team spirit” and is proud of who they are.