Kanban Remastered

Agile Lessons from Strategy Games

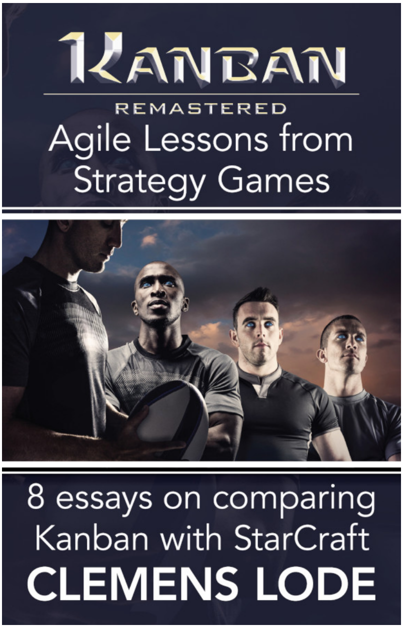
8 Essays on Comparing Kanban with StarCraft

KANBAN REMASTERED

Agile Lessons from Strategy Games

8 Essays on Comparing Kanban with StarCraft

Published by Clemens Lode Verlag e.K., Düsseldorf


KANBAN
REMASTERED
Agile Lessons from Strategy Games
8 essays on comparing Kanban with StarCraft
CLEMENS LODE

2019, Second Edition

ISBN 978-3-945586-66-2

Edited by: *Conna Craig*
Cover design: *Jessica Keatting Graphic Design*
Image sources: *Shutterstock*, *iStockphoto*
Icons made by http://www.freepik.com from
http://www.flaticon.com is licensed by CC 3.0 BY
(http://creativecommons.org/licenses/by/3.0/)
Subscribe to our newsletter. Simply write to newsletter@lode.de or
visit our website https://www.lode.de.

Printed on acid-free, unbleached paper. MANAGEMENT
PSYCHOLOGY

Introduction

神の一手—*Kami no Itte*—Japanese, roughly meaning "move of God" or "godly move"—describes an entirely new insight concerning a move during the game called "Go." Such a move is a goal taught to students to be more attentive toward less obvious maneuvers, leading the students to focus on alternatives. Likewise in management, first put aside the conspicuous answers (like adding more people to an already late project) to make way for an objective mind and attention to alternatives. In management, the challenge is to discover the potential of the team and your organization and to build upon that.

This book shows some of the 神の一手—*Kami no Itte*—of management, focused on the Kanban methodology and the computer game StarCraft. *Kanban Remastered* provides ideas for a Kanban coach to teach a team about Agile principles by building upon the team members' knowledge of games like StarCraft.

Contents

Publisher's Note

Thank you for keeping up the tradition of reading books. You and your fellow readers have created a market for this book. I hope that I can meet your expectations and I am looking forward to feedback from your side, no matter whether it is positive or negative. To send general feedback, mention the book title in the subject of your message and simply send it to feedback@lode.de. You can also contact us at https://www.lode.de/contact if you are having a problem with any aspect of the book, and we will do our best to address it. Also, we cordially invite you to join our network at https://www.lode.de.

Although I have taken every care to ensure the accuracy of the content, mistakes do happen. If you find an error in one of my books, I would be grateful if you would report this to me. By doing so, you can help me to improve subsequent versions of this book and maybe save future readers from frustration. If you find any errata, please report them by visiting https://www.lode.de/errata, selecting the book title, and entering the details. Once verified, the errata will be uploaded to our website. You will, of course, be credited if you wish.

Best regards,

Clemens Lode
CEO, Lode Publishing
Düsseldorf, Germany, July 1st, 2019

Preface

It has been 20 years since the release of StarCraft, with a "remastered" version coming out this summer (2017). Having been an avid player, I only now realize that a surprising amount of what we know as "Agile project management" is actually implicitly part of a regular game of StarCraft.

Being an Agile coach, I realized that especially in industries with teams with a gaming background, coaching lessons might benefit if I could show the relationship between games and Agile methodology. I was able to explain management methodology in a language that the tech-savvy audience understands.

StarCraft, the most popular real-time strategy game of all time, is all about producing and deploying as many game units as possible at just the right time. This book is about the relationship between StarCraft and Kanban. When your team knows StarCraft but not Kanban, this book will provide you with a series of analogies to allow a better and easier understanding of Agile principles. It is written in a light-hearted tone, similar to how you might chat with a fellow coach about your Agile experiences implementing Kanban, taking for granted that you have experience with StarCraft.

Despite the subject of my previous book, *Scrum Your Jira!* and despite Scrum being the most popular Agile methodology (at least on paper), I decided to compare StarCraft with *Kanban* instead. The reason is that StarCraft knows no separate roles, planning phases, or sprints and is all about managing your limited time to give commands on the battlefield.

Currently, Kanban is one of the most popular Agile management techniques besides Scrum and has also found its way away from pure logistics to software development. It can be used in other industries or work fields, but most of the examples I will bring up will be about software.

Ultimately, though, my personal stake in this subject is my interest in how the world works. Twenty years ago, I had a dream: to learn from nature. Reading about evolutionary algorithms that solved complex problems, step by step, fascinated me. It was the reason why I got into computers, and it was the reason I later embraced Agile development principles. During my computer science studies, I combined this interest with my hobby of playing StarCraft, which is why I chose StarCraft as an example to draw parallels between games and management. The result of this mixture was Evolution Forge, which I will discuss in the last chapter of this book.

Clemens Lode
Author, *Kanban Remastered*
Düsseldorf, Germany, July 4th, 2017

Chapter 1

An Introduction…to StarCraft

There is a saying that if you want a successful software developer team, hire StarCraft players. Why? StarCraft not only trains you to react quickly, but it also teaches you valuable knowledge about management. For the players, this is not necessarily obvious, and management will often be viewed as something abstract and not related to their work. For an Agile coach, having a team with such background knowledge can help to speed up the learning process, and most importantly, increase the chances that the process will be accepted, becoming part of the team's culture.

For those unfamiliar with the game, let us first look at what StarCraft is. Then, in the following chapters, we will compare StarCraft with Kanban.

The computer game StarCraft puts you, the player, into a management position. Your task, in the beginning, is to manage three resources: minerals, Vespene gas, and supplies. The goal is to provide the "general" (also played by you) with military and support units. The specific management goal depends on the military scenario, but it usually revolves around defeating an enemy general.

Combat is based on a sophisticated version of a "rock, paper, and scissors" model (air, close-combat, and ranged units), so it is important to gather intelligence about what type of units the enemy produces. There are only a few effective ways of gathering intelligence, mainly moving a unit into the enemy's territory. The general decides on the strategy and provides the manager with the priorities based on the "market," the plan of your enemies: which units are most needed. The manager implements these priorities by using a specific build order, assigning worker units to produce factories, and factories to provide the military units or more worker units ("SCVs") in the "Command Center."

> **SCV** · In StarCraft, the *SCV* is the all-purpose worker unit who gathers resources and constructs buildings. While it can defend itself, it is the weakest unit in the game. While it seems wise to produce new workers non-stop, you might want to temporarily sacrifice long-term growth with short-term gains: finding the right window of opportunity to build specific units, instead of investing into economic growth, might lead to you winning the game or at least getting ahead. Both in the game as in business, it does not matter how good your army or your product is, but how it is relative to your competition. Do not try to build a perfect army or perfect product. Check the market, check what consumers want: they want a better product than what the competitors are offering.

> **COMMAND CENTER** · In StarCraft, the *Command Center* is the first building you own. It can produce additional workers to mine resources. Building a second Command Center might speed up your worker production and resource gathering speed, but it is also costly. In a company, it could be compared with the basic structure of hiring new people, training, people management, etc. Both in StarCraft and in business, you are faced with the question of how many people you will hire for a project. Will you employ just a few and save money, or do you stop any other current investments and hire more people now to have them finish the project more quickly?

As in business, you will not win the game if you take forever to produce your "perfect product"—an army, in this case. Units produced as early as possible (or just at the right moment) can give you the upper hand in the game.

Beyond combat units, you also have carriers ("Dropships") and buildings. The former can be seen as a means of "deploying" units by transporting them to the battlefield. The latter can be used to block enemy units, or—wrongly-placed—to prevent your own movements, making deployment more difficult.

> **Dropship** · In StarCraft, the *Dropship* is basically a spaceship that can transport your units from one place to another. They can be seen as a means of "deploying" units. This element of the game is similar to what you have to do in a business. It is not enough to produce items, you also need to bring them to a place where your customers can see and buy them, be it a physical location or software on a web server. If appropriately coordinated, Dropships can be used for surprise attacks. Again, a concept you find in the business world: you need to plan your release of the product according to the external market forces, be it special dates (like Christmas) or competitor products. If you catch your competitor unprepared, the competitor might need time to adapt to your new product.

Resource management involves sending worker units to mineral fields and Vespene geysers in order to mine minerals and Vespene gas, which is used for production. These fields are depleted after a while, so you need to look out for and secure new locations that have resources. Also, you need to build supply depots to support your standing army. If you do not have enough supplies, your production stops, and you can no longer produce new units, so you have to plan for that in advance.

Besides minerals, gas, and supplies, time is another resource, as combat usually requires you to spend time micromanaging your units to move army groups and attack specific targets. As StarCraft is a real-time strategy game, any time used tending to combat cannot be used for managing to your production and resources. While you can queue movement commands at no cost, queuing production commands in your factories binds in advance resources that might be better used elsewhere (but allows for production without any interruption).

Even more complexity is added by the fact that there are three different factions, the option to research better weapons and armor, and tech-trees that are unlocked when constructing certain buildings. Last but not least, there are islands, cliffs, ramps, narrow pathways, and hills that add additional strategical and tactical challenges.

Now, this is all interesting in theory. But what does it look like? I recommend watching a few StarCraft games together (team building!) and discussing how elements of the game apply to the situation in your company. It is an excellent way of opening people up. By seeing it "in action," your team can more easily understand the management terms and concepts in Kanban. In fact, I encourage you to watch a few minutes of StarCraft replays right now or download the game (it's free) and play. If your manager asks, say that it is for research.

Done? Now on to the next chapter, comparing StarCraft to Kanban.

Chapter 2

Kanban and StarCraft

> When you choose to use Kanban as a method to drive change in your organization, you are subscribing to the view that it is better to optimize what already exists, because that is easier and faster and will meet with less resistance than running a managed, engineered, named-change initiative. Introducing a radical change is more complicated than incrementally improving an existing one.

—David J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*

In an organization, work can be distributed by two different approaches: "push" or "pull." You either get new work put on your desk and have to manage what to prioritize, or you ask for new tasks once you are finished with your old ones.

For example, "Gosplan," the agency responsible for central economic planning in the Soviet Union, used a push-based logistical system. At Gosplan, mathematical models predicted consumer behavior. Based on these models, the government created plans for the entire supply chain from the factory to the shops—with catastrophically bad results. For certain goods, there was always either too much or too little available. By contrast, "Kanban" is different:

> **Kanban** · *Kanban* is Japanese and literally means "signboard." In the context of project management, the term is interpreted as "queue limitation." Kanban is a method designed to reduce unfinished work and wasteful inventory levels; it was originally developed at Toyota in the late 1940s. Back then, marketers at Toyota studied consumer behavior and supermarket stocking strategies and applied the ideas to logistics in industrial production. At Toyota, they had previously produced as much as possible, regardless of the demand from the market. In contrast, in supermarkets, customers take only what they need, expecting that the supermarket will be stocked up the next time they visit. The customer "pulls" an item from the shelves, and the supermarket makes sure to refill the shelves. This new Kanban method applied to production provided just as much as what was needed, just in time.

In addition to improving the production flow within a company, the significant advantage of Kanban is that it can be applied to the production phases of any existing organizational structure without having to change business processes. With the production flow being made transparent, you can detect where work piles up at one place and then introduce a limit of the amount of work that is in progress. As opposed to a "relay race" model where work is *pushed* to the next department and not followed through, in Kanban, work items are *pulled*, and work is stopped when the work limit of the following department hits its limit. This ensures that no work piles up or gets lost.

For example, cars need tires, a chassis, and a motor. Without a Kanban system, the three departments responsible for these parts will produce as much as possible: the company will always end up having either too many chassis, motors, or tires. If you instead have the departments check how many items are already in stock and stop producing when the stock is full, you save a lot of money. Sure, you might need these items later, but keeping a large inventory costs money that you could have invested elsewhere.

When presenting the situation in a business this way, it opens an objective discussion at the management level, ideally followed by incremental, evolutionary improvements. As opposed to Scrum which can be executed within a company "by name only," i.e., by following the ceremonies but never addressing core issues like multidisciplinary teams (see my book *Scrum Your Jira!*), Kanban is an organization-wide change management approach. Sure, Scrum involves creating lists of impediments, but it leaves it up to you how to deal with them.

> **SCRUM** · *Scrum* is a set of management tools that focuses a project back on the team level and uncovers internal and external impediments of the production process. By reducing communication paths through small, multidisciplinary teams, as well as frequent releases to the customer for review, the probability for project success can be improved even if the scope is not clear from the start. Also, work is divided into units of fixed lengths (sprints), which helps to plan future sprints with your team working at a sustainable speed.

Once Kanban is in place, the goal is to focus on the bottlenecks, and manage the flow. Discussions with management can best be lead by making things explicit. Identifying the bottlenecks themselves is just the start of your work. The real game-changer is to make explicit the current collaboration policies. This moves the discussion towards objectivity away from the abstract and maybe emotional or anecdotal arguments.

Usually, after an organization-wide introduction of Kanban, the first bottleneck is found at the top. If the organization is strongly hierarchical, work (decisions!) piles up at the desk of the organization's leader. In StarCraft, this is not much of a problem as playing in a team inherently includes decentralized control, but it helps to imagine how cumbersome a game would be if all decisions had to go over a third party's desk.

Even with a decentralized organization, a common argument against Kanban is that there will be idle time because one part of the organization might not be able to keep up with the rest. As a result, work piles up until it hits the limit. This is true, but the idea that you are doing a good job when everyone is working at 100 percent is not always accurate. This is comparable to, in StarCraft, trying to keep all building facilities active, whether you need the units at the moment or not—or maybe even refraining from building additional facilities fearing they would not be used all the time.

Ultimately, Kanban is about trying to improve the flow through an organization. It does not matter how many items you produce if your sales department cannot bring them to the customer. Likewise, in StarCraft, you must think about how you will use the units you produce, meaning how you will deploy them on the battlefield. Sure, creating as many units as possible is a viable strategy, just like you could throw your unsold products on the market at a lower price or keep them in storage "just in case" there is a sudden demand. But it is an inefficient strategy. *We can do better than mere local optimization.*

This rings especially true when looking at the market. While a company does not literally fight an enemy like you do in StarCraft, your competitors try to get your market share. Scouting your enemy and adjusting your strategy are central components in StarCraft, just as they are on the market. You need to have foresight and gain intelligence about them and maybe even think about making some risky investments before you are absolutely sure what others are doing. Why? Because waiting itself is a risk and you might miss the window of opportunity to be the first on the market with your product.

When all is in place, you can focus on communication. Using Kanban automatically leads to situations where a team is stuck insofar as it cannot pull new items to work on because the subsequent phases or departments have hit their work in progress limit. This encourages collaboration, where one team can help out the next and where teams sit together and think about how to prevent future bottlenecks. This element can be found in StarCraft, too, with very close team communication over audio during the game, as well as a review of the game and discussions about how to optimize team play afterward.

No matter the size or structure of your organization, take small steps. Kanban promotes this approach by starting with what you have in place and pointing out the bottlenecks. Where StarCraft falls short, by comparison, is the visualization of the "workflow." While tools eventually emerged that visualized some aspects of the replays, like your "actions per minute" symbolizing your workload, I know of no tool that does it the Kanban way and actually analyzes how much time you spend on each team or location and thus points out possible paths for optimization.

The closest software that does a similar job is Evolution Forge, which I will discuss in the last chapter of this book. It helps you to optimize your basic build order in small steps. Behind both approaches (Kanban and Evolution Forge) is the grand idea inherent in nature to leave things as they are and move forward without ever taking a step backward. Every change you make should improve the situation, and more substantial changes come into place as the sum of a whole number of smaller ones. In that regard, if you want to improve your StarCraft play the Kanban way, manual observation, maybe together with a critical friend, might be the best choice. On the other hand, if you want to learn the idea of Kanban with StarCraft as an easy-to-understand reference, you have come to the right place!

Chapter 3

The General

> “It is the sovereign’s function to give broad instructions, but to decide on battle it is the function of the general.
>
> —Sun Tsu, *The Art of War*

When implementing Agile methodology, some things are often overlooked: the organizational structure, the culture of the organization, and the people involved. Management looks at the market, hears “Agile” as the new buzzword, and thinks that this is the solution to the problems they (management) figure exist *within* the development department.

Introducing Agile is a change process, and change is hard. Why is it hard? Because over time, no matter how effectively an organization is structured, people fine-tune themselves to their organizational environment to at least have locally optimal work conditions. Changing any aspect of the organization then leads to those local optimizations becoming worthless, which leads to lower productivity and stronger resistance to change.

Because of this, when adopting Agile methods, an organization leader needs to take extra care to not only change one part, but also to change the whole organization (over time). Every part of the organization is involved because the idea of Agile is to combine different phases into one.

> **Agile** · *Agile*, in the context of project management, is a method to reduce waste and delays by anticipating that plans will change. It is a set of methods which are most effective when applied to projects that are complex and chaotic, especially in product development. It also has its place in production, given that customer demand as well as productivity fluctuate.

> If words of command are not clear and distinct, if orders are not thoroughly understood, then the general is to blame. But, if orders are clear and the soldiers nevertheless disobey, then it is the fault of their officers.
>
> —Sun Tsu, *The Art of War*

Kanban itself does not require specific roles within the team. What it does, though, is uncover problems in the organization that could be solved by assigning a single "product owner" to the team in order to give the team a vision and help with prioritizing tasks. Most often, the opposite is actually true; namely that you have *too many* product owners and overlapping responsibilities.

Imagine you have a software team and an architecture team, each with a separate supervising product owner. The software team needs machines for production deployment, but the product owner of the architecture team prioritized another project. In theory, in true Kanban fashion, the software team could move in to help out the architecture team and speed things up. In reality, this could fail for many reasons, such as lack of knowledge, access rights, or merely a difference of opinion about how the IT infrastructure should look. Also, the software team may get conflicting messages from both product owners about how to proceed. Focusing both development and IT architecture by putting them under the command of a single product owner can prevent these problems, and deployment could be streamlined.

Looking at StarCraft, you will encounter both scenarios in multiplayer mode. There is no separate "general" role that oversees how a team should concentrate its forces. Instead, players individually decide where and how to attack. They are very opportunistic, quickly joining an attack by other members of the team with their available forces, and with little to no need for a manager at the top. Also, the most suitable (in terms of experience, size of force, proximity to the base, etc.) player automatically becomes a temporary leader in a specific situation.

In games like StarCraft, the general is in full control of land and air forces, from production to deployment. However, looking at military history, we see a totally different approach by some leaders. Sun Tsu, a Chinese general and military strategist in the "Spring and Autumn" period of Chinese history (722 BC to 470 BC), most famous for his work *The Art of War*, pointed out that it is essential for victory that generals are unconstrained by their superiors. A general must be free to wage war without interference. In World War II, the Allied command structure gave total authority to General Eisenhower, with four commanders representing the US Navy Group, United States Army Air Forces, US Army Group, and British Army Group. Each commander had clear responsibilities and was able to plan and execute operations independent of the other commanders. Land, air, and sea operations could be prepared by the individual commanders, and only joint operations that required a combination of groups had to involve Eisenhower directly.

One would expect an even better-organized hierarchy on the German side. However, there, you had a whole array of overlapping authorities with, for example, tank divisions being distributed among different army groups. The idea behind Hitler's organizational setup was primarily not about efficiency on the battlefield, but to retain control over the various factions within the German army and to have the final word on all decisions (i.e., to be the dictator). Before any of his commanders could execute an operation, it had to go over Hitler's desk because only he had all the necessary information required to make a strategic decision.

Imagine playing a game like StarCraft if team players had to share half their units with another player from the team—with the only possibility of talking to that player being via a third person who kept (informational) control this way. This allows the project to be micromanaged and controlled by a third person (basically a "dictator"); but you will lose the creativity and flexibility emerging from decentralized management.

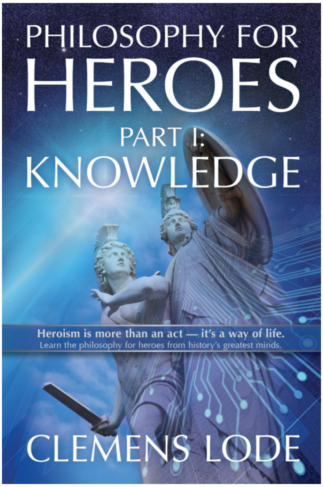
In business, it is relatively easy to recognize the type of organization in which you are working. How are budgets allocated? Can individual departments work closely with other departments without the involvement of the CEO? Do the departments have the authority or can decisions be enforced only by mentioning the CEO (or adding him or her to the CC line in email messages)?

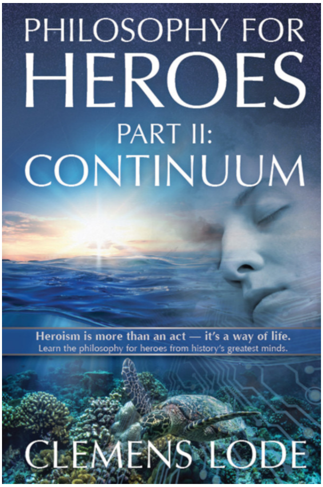
Sure, organizations ultimately have to remain unified. A CEO cannot have one department running wild. And there is usually a reason that an organization is the way it is, so merely changing the leadership style might actually result in even more chaos. Ultimately, it depends on the people involved. What type of people are they? Do they prefer a clean organizational chart with clear responsibilities, or can they operate only based on personal bonds and informational control? Do they share and work toward a shared vision, or do they prefer having a strong leader directing the way from case to case?

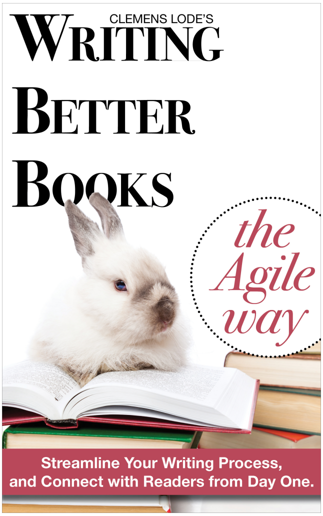
While every organization is different, you will find the latter type of organization more often in the field of sales-focused industries, while the former is usually to be found in production-focused industries. I think that both kinds of organizations have their place, but when you want to produce and run projects, you might want to think twice about wanting your organization to be controlled from the top.

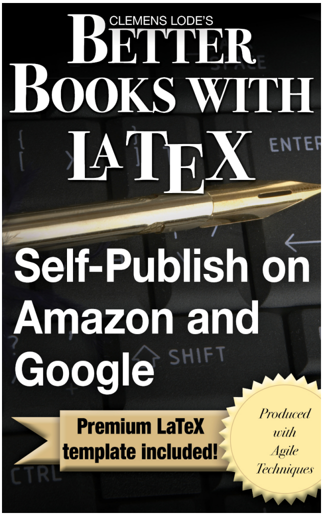
What type of structure do you have in the organization where you work? Is it led by someone who is focused on production or on sales? How does that affect your experience with change in the organization?

Other Books (available in paperback and e-book!)


PHILOSOPHY FOR
HEROES
PART I:
KNOWLEDGE
Heroism is more than an act — it's a way of life.
Learn the philosophy for heroes from history's greatest minds.
CLEMENS LODE


PHILOSOPHY FOR
HEROES
PART II:
CONTINUUM
Heroism is more than an act — it's a way of life.
Learn the philosophy for heroes from history's greatest minds.
CLEMENS LODE


CLEMENS LODE'S
WRITING
BETTER
BOOKS
the Agile way
Streamline Your Writing Process,
and Connect with Readers from Day One.



SCRUM YOUR JIRA!
Your Waterfall Organization Transformed into Multidisciplinary Teams
10 Essays on Using Online Tools to Improve an Offline Technique
CLEMENS LODE

The Author

Clemens Lode works as an author as well as a coach for software teams throughout Europe. He lives in Düsseldorf (Germany). You can follow him on Facebook (https://fb.me/ClemensLode) or Twitter (https://www.twitter.com/ClemensLode), or just drop him a line (clemens@lode.de).

> “What I cannot create, I do not understand.
>
> —Richard Feynman

Glossary

A

Agile ● *Agile*, in the context of project management, is a method to reduce waste and delays by anticipating that plans will change. It is a set of methods which are most effective when applied to projects that are complex and chaotic, especially in product development. It also has its place in production, given that customer demand as well as productivity fluctuate.

B

Barracks ● In StarCraft, the *Barracks* are the first production facility to produce combat units (the Space Marines). Building the Barracks also allows building other, more technologically advanced, production and research facilities, making the Barracks a cornerstone of any build order. The Barracks can be compared with the very basic structure of an enterprise with sales and marketing: No matter how good your product is, you still need to market and sell it!

C

Command Center ● In StarCraft, the *Command Center* is the first building you own. It can produce additional workers to mine resources. Building a second Command Center might speed up your worker production and resource gathering speed, but it is also costly. In a company, it could be compared with the basic structure of hiring new people, training, people management, etc. Both in StarCraft and in business, you are faced with the question of how many people you will hire for a project. Will you employ just a few and save money, or do you stop any other current investments and hire more people now to have them finish the project more quickly?

D

Dropship ● In StarCraft, the *Dropship* is basically a spaceship that can transport your units from one place to another. They can be seen as a means of “deploying” units. This element of the game is similar to what you have to do in a business. It is not enough to produce items, you also need to bring them to a place where your customers can see and buy them, be it a physical location or software on a web server. If appropriately coordinated, Dropships can be used for surprise attacks. Again, a concept you find in the business world: you need to plan your release of the product according to the external market forces, be it special dates (like Christmas) or competitor products. If you catch your competitor unprepared, the competitor might need time to adapt to your new product.

K

Kanban • *Kanban* is Japanese and literally means "signboard." In the context of project management, the term is interpreted as "queue limitation." Kanban is a method designed to reduce unfinished work and wasteful inventory levels; it was originally developed at Toyota in the late 1940s. Back then, marketers at Toyota studied consumer behavior and supermarket stocking strategies and applied the ideas to logistics in industrial production. At Toyota, they had previously produced as much as possible, regardless of the demand from the market. In contrast, in supermarkets, customers take only what they need, expecting that the supermarket will be stocked up the next time they visit. The customer "pulls" an item from the shelves, and the supermarket makes sure to refill the shelves. This new Kanban method applied to production provided just as much as what was needed, just in time.

P

Pomodoro Technique • The *Pomodoro Technique* is a simple time management technique that allows focusing on one task, followed by a break, then moving on to the next task (Cirillo, *The Pomodoro Technique: The Acclaimed Time Management System That Has Transformed How We Work*).

S

Scrum • *Scrum* is a set of management tools that focuses a project back on the team level and uncovers internal and external impediments of the production process. By reducing communication paths through small, multidisciplinary teams, as well as frequent releases to the customer for review, the probability for project success can be improved even if the scope is not clear from the start. Also, work is divided into units of fixed lengths (sprints), which helps to plan future sprints with your team working at a sustainable speed.

SCV • In StarCraft, the *SCV* is the all-purpose worker unit who gathers resources and constructs buildings. While it can defend itself, it is the weakest unit in the game. While it seems wise to produce new workers non-stop, you might want to temporarily sacrifice long-term growth with short-term gains: finding the right window of opportunity to build specific units, instead of investing into economic growth, might lead to you winning the game or at least getting ahead. Both in the game as in business, it does not matter how good your army or your product is, but how it is relative to your competition. Do not try to build a perfect army or perfect product. Check the market, check what consumers want: they want a better product than what the competitors are offering.

Service Level Agreement • Thinking about Kanban leads to thinking about how different departments or teams work together. Written or not, there is always some sort of contract between the parties involved. A *Service Level Agreement* is such a contract and usually denotes the time between the initial request (e.g., for a software fix) until the first response by the team. This contract is established implicitly when people meet the first time or have the first telephone call. "Do you have a minute?" "Sure!" is a commonplace contract which translates into

“Drop everything you are doing right now and focus on my problem.” For the person requesting the service, this means that the supporting person is available on very short notice and ongoing work can be interrupted. In Kanban, with time-boxing, the answer is “Sure, but let me first finish what I am doing right now, and check if there is other, even more important work.” If the current tasks are generally small, the request will still be worked on within a short time, but any overhead related to stopping and restarting ongoing tasks is prevented.

Space Marine ● The *Space Marine* is the basic combat unit in StarCraft. It is the dominant unit in the early part of the game, the only protection between your own base and the enemy. Space Marines are produced in the Barracks and will be used as a basic example, in this book, for build orders. Just like a product needs several parts to be able to make a single sale, a lone Space Marine is the weakest unit of the game. Their real power shows when they act in a group, with other units supporting them.

S

Tech-Debt ● In software terms, “tech debt” usually refers to code that later needs to be rewritten or systems that later have to be scrapped and rebuilt and reconfigured. The term can apply to anything that saves you time now but has to be paid back later in the form of additional work.

Time-boxing ● *Time-boxing* means to work on similar tasks during a specific time period. For example, instead of answering emails throughout the day, reserve half an hour each day to work through any unanswered emails. This eliminates the overhead of having to refocus between different tasks.

W

Waterfall ● *Waterfall* is a project management method where a product moves through a number of phases before a final version is finished for release. Compared to Agile, the problem with this method is that it requires additional communication channels between the individual phases and the time until a team or company gets feedback from a customer is generally much longer with Waterfall than Agile.

Work In Progress ● The general idea in Agile project management is to limit the number of things you work on at the same time. In Scrum, you limit it by setting a fixed time frame (sprint) for a work package. In Kanban, you directly limit the number of tasks or projects worked on. This reduces overhead and automatically will lead to more complete tasks. If you can focus on but a few tasks and bring them to a finish, they are no longer dragged along half-finished without any value.

Bibliography

Cirillo, Francesco. *The Pomodoro Technique: The Acclaimed Time Management System That Has Transformed How We Work*. Crown Business, 2017. ISBN: 978-1524760700.

Lode, Clemens. *Kanban Remastered*. Clemens Lode Verlag e.K., 2017. ISBN: 978-3-945586-68-6.

— *Scrum Your Jira!* Clemens Lode Verlag e.K., 2017. ISBN: 978-3-945586-69-3.

Index

Note that the indexes that resemble a file structure (e.g., /back/amazon.tex) refer to a paragraph in the book where the contents of the file from the Overleaf template are discussed.

An Important Final Note

Writers are not performance artists. While there are book signings and public readings, most writers (and readers) follow their passion alone in their writing spaces at home, in a café, in a library, at the beach, or at a mountain retreat.

What applause is for the musician, ***reviews*** *are for the writer.*

Books create a community among readers; you can share your thoughts among all those who will or have read this book.

Please leave a thoughtful, honest review and help me to create such a community on the platform on which you have acquired this book. What did you like, what can be improved? To whom would you recommend it?

Thank you, also in the name of all the other readers who will be better able to decide whether this book is right for them. A positive review will increase the reach of the book; a negative review will improve the quality of the next book. I welcome both!

“ Two things destroy businesses—mediocrity and making it about yourself.

—Halt and Catch Fire