Nonfatal Transformation

Peter Pirogov



How to change our organizations without destroying them

August • 2021

Nonfatal Transformation

How to change our organizations without destroying them

Peter Pirogov

Part I The Recipe for Agility

Motives for change and a brief summary of the modern management techniques and frameworks

Part II Walking Around the Traps

Benefits and challenges of the autonomous product team concept

Part III Skeletons in the Closet

Why transformation emerges, what it needs to succeed, and how it may become fatal

Part IV Adventures of One Animation

Real cases of how organizational models emerge, evolve, applied, and mixed together

I want to thank all my great colleagues and friends I was lucky to work with in many excellent teams. The experiences we went through — hours of thinking together, sparkles of contradictory ideas, the magic of balancing views that seemed radically different — are a tremendous source of enthusiasm and learning. Thank you to Vladimir Glavnov, whose experiments and tireless attention to methodologies inspired this article. And special thank you to Fernando Hernández, Timofey Pirogov, Alexander Yaroshevich, and Daniel Mariño for reviewing the early drafts and contributing many excellent thoughts and suggestions.

DRAFT FOR COMMENTS | August 2021 | Rn119P-20210831 | Printer-friendly version

On the cover: drawing by Henrik Kniberg, photo by Andrey Maximenko (via iStock)

Fonts: *Gudea* by Agustina Mingote and *Source Serif Pro* by Frank Grießhammer via Google Fonts

All later text in national alphabets (excluding text within illustrations) was either translated to English or transliterated into Latin characters to ensure readability. Images were retouched to be viewed or printed in high resolution.

© 2021 Peter Pirogov

Contact Peter at his LinkedIn page: linkedin.com/in/peter-pirogov

Contents

Part I: The Recipe for Agility **4**
Intro and disclaimer 4
Why to transform? 4
The ingredients 6
How true is the recipe? 12
Key takeaways from Part I 13
Part II: Walking Around the Traps **14**
Core benefits 14
Challenges and drawbacks 17
Contradictions 24
Key takeaways from Part II 27
Part III: Skeletons in the Closet **29**
Intel story: Only the paranoid survive 29
Opposition to change 31
Transformation essentials 33
Boeing story: Fatal disconnection 39
Key takeaways from Part III 43
Part IV: Adventures of One Animation **45**
Evolution of Spotify 45
Revolution at ING 50
Transformation of Sberbank 55
Forming your change 60
Key takeaways from Part IV 64
Closing thoughts 66

Part I: The Recipe for Agility

Motives for change and a brief summary of the modern management techniques and frameworks

Intro and disclaimer

If you are a manager, what do you usually do at work? Most of us "fight in trenches" and "hold the fort" at any given moment. Within this routine, we never have time to stop and think about big questions "What is next? Where are we heading?" As I was lucky to get a long and peaceful sabbatical, I had a chance to take a "helicopter view" on our "battlefield." In this article, I tried to put together some information and ideas resulting from my little research. Please note that the text below, unless it has an explicit link to a source, reflects my personal and possibly biased opinion backed up only with my professional experience.

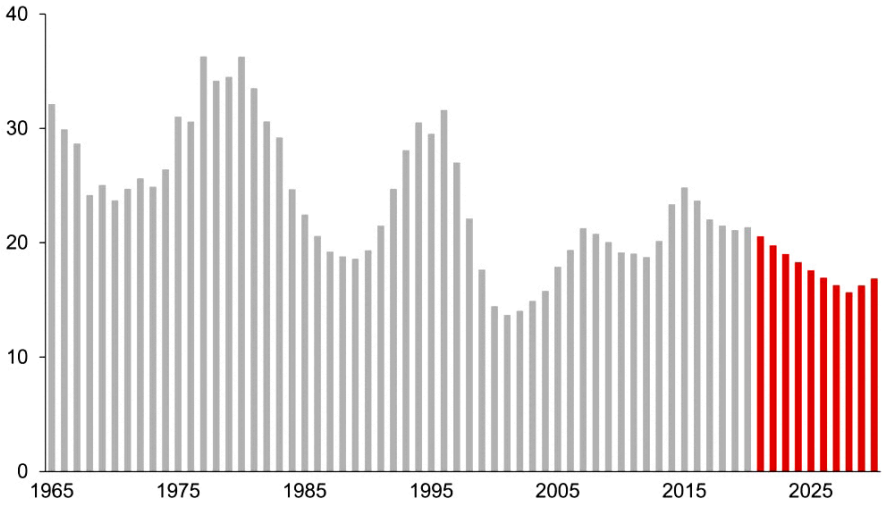
Why to transform?

Today it is hard not to notice that the world is changing at an extreme pace. It feels like riding a bicycle down a steep hill: exciting but a bit frightening. For many businesses "disrupt or be disrupted" narrative became both, biggest fear and desired management strategy.

Businesses feel threatened by an unknown startup or a Big Tech Giant cooking a product or a service that would destroy the current operational environment. A well-known example is the case of Uber, Lyft, DiDi, Grab, Yandex.Taxi and several other players successfully disrupting (and sometimes destroying) the business of legacy taxi companies.

A good reflection of the disruption process at a bigger scale, we see in the lifespan of companies included in the S&P 500 list. The corporate longevity research

of Innosight shows that newer giant corporations outgrow (or acquire) older giant corporations at an increasing pace. If in late 1970s a corporation could hold its place in the list for 30- to 35-year average, the forecast for 2028 is just 12 years. Tesla, for example, has entered S&P 500 at the end of December 2020. Would Tesla survive and stay on the list for more (or less) than 12 years — we will see.



Source: illustration for the briefing by S. Patrick Viguerie, Ned Calder, and Brian Hindo, *2021 Corporate Longevity Forecast*, Innosight (2021)

Average company lifespan (in years) on S&P 500 Index (each point is rolling 7-year average; data in red is a forecast), the illustration from Innosight's 2021 Corporate Longevity Forecast

If "too big to fail" giants are so short-lived nowadays, "regular" big and medium companies should be very concerned about their survival. Many business owners, companies' boards of directors, and top corporate executives consider the situation critical or, at minimum, very serious. This often pushes leaders of organizations to initiate internal transformations. Notably, this urge for change is equally strong in state-dominated economies like Russia, where Big Tech Companies have a hard time doing business. The only difference is that in such economies, your business space is more likely to be taken over by a state-controlled corporation, not an international tech giant.

Transformations are usually prepared (unfortunately, not always well), directed, and supported by internal stakeholders, external consultants, and other actors, which altogether are further called *change agents*. A transformation campaign can come under different names:

- For old-school "digital" businesses (for example, a software vendor or a bank), this change would usually be called *Agile transformation*. In most cases, such transformation would start at the division(s) responsible for developing new or existing products and services. This is what this article is about.

- For old-school "analog" businesses, such as pulp mills or steel plants, a similar change process is usually called *digital transformation*.

The ingredients

As arguable as it sounds, there seems to be a consensus among the professional community on the features of today's successful organization. Depending on a consultant or practitioner you listen to or a book you read, the picture may vary considerably. Nevertheless, each recipe would likely include most of the following seven ingredients — in different proportions and with different flavors:

- Various Agile frameworks
- Various Agile at scale frameworks
- Various Lean practices
- Concept of the autonomous product team
- Concept of flattened self-managed organizations
- Goals & actions alignment frameworks
- New product management and marketing techniques

Let's take a closer look at each ingredient.

INGREDIENT 1: Various Agile frameworks

Essence: Originated from software development, Agile is a way to create a product or feature in small, regular, and (preferably) functional increments, ideally verified by end-user. This approach is expected to minimize time-to-market and improve the correlation between product features and customer needs. Agile techniques are designed to support effective teamwork. Teams in Agile are usually small, from 3 to 9-10 people.

Flavors and sources:

- Scrum. Most popular, very strict, and "religious" framework. It has an official guide and is equipped with a number of well-settled ceremonies, roles, artifacts, and certification institutions. Many dogmatic evangelists state that not exact adoption of Scrum is not Scrum. Fortunately, when you need more flexibility, you can easily resolve this contradiction by giving your team's real-life best practice some other name (see stories in Part IV).
- Kanban. Another popular framework, inspired by Lean practices (see below). It can be combined with Scrum (together, often called Scrumban) and other techniques.

- Extreme Programming (XP). XP is not a single framework but rather a collection of helpful programming practices. Often used in combination with other frameworks.
- Hybrids and custom variations of frameworks listed above plus the number of less common methodologies.

To get familiar with Scrum (without extra technicalities), I would recommend "Scrum: The Art of Doing Twice the Work…" by Jeff Sutherland. Jeff (along with Ken Schwaber) is the original creator of Scrum. A similar source, but for the Kanban, is "Kanban: Successful Evolutionary Change…" by David J. Anderson. For a pragmatic perspective from a good practitioner, see "Scrum and XP from the Trenches," "Kanban and Scrum — Making the Most of Both" by Henrik Kniberg.

INGREDIENT 2: Various Agile at scale frameworks

Essence: If Agile transformation has succeeded on the team level, legacy organization structure usually becomes the next bottleneck (using TOC terminology). Agile at Scale Frameworks were designed to answer that challenge by providing new structures and processes on the level of a large department or whole organization.

Flavors and sources:

- Scaled Agile Framework (SAFe). Most popular organization-level framework (see the link to KPMG Survey below). Many critics argue that SAFe is overcomplicated and trying to combine legacy management techniques with Agile. On the bright side, with a flexible mind, you can select a few of the many SAFe elements that suit your environment and discard the rest (see Lego case in Part IV).
- So-called Spotify Model. The only popular framework that does not officially exist and is not promoted by any institution (i.e., nobody can legally make money on courses and certifications). Spotify Model, in a nutshell, is a snapshot of Spotify org structure and processes, as they existed in 2012-2014. More about this rare animal see in Part IV.
- Large Scale Scrum (LeSS). A strict, Scrum-based framework in which several universal cross-functional teams work on one product while none of those teams carry full responsibility for the product or its components.
- Scrum of Scrums. Simple and popular coordination technique providing limited scalability.
- Scrum@Scale. A relatively recent framework created by Jeff Sutherland, it has some similarities with Spotify Model.
- Nexus. One more framework based on the pure Scrum, created by Ken Schwaber, featuring a special virtual superstructure called "The Nexus Integration Team."

- Disciplined agile delivery (DAD). A moderately popular framework that was acquired by PMI in 2019 to "expand offerings".
- Several other frameworks, including custom variations and hybrids.

Importantly, the scaling frameworks distinguish two types of Agile teams:

- functional or component teams, which include people with similar area of expertise (functional specialists)
- cross-functional (XFN) or feature teams, which include people with all different specializations needed to cover the whole task

The main benefit of a cross-functional team is its autonomy — the ability to work independently across a complete technology stack of R&D organization and solve full-cycle tasks with no or minimum handovers. Such teams are preferably formed by so-called T-shaped professionals. If both product architecture and organization's structure are built to minimize teams' dependencies, a cross-functional team can evolve into an autonomous product team (covered below). See more on cross-functional teams in Part II.

Some scaling frameworks are suited for both component and cross-functional teams (SAFe), while others require cross-functional teams (LeSS).

For an informative overview of the Agile at Scale Frameworks and statistical estimations, see KPMG Global Agile Survey 2019. Despite the data being a bit old and the regional coverage limited, the survey provides valuable insights on the Agile transformation.

Ingredient 3: Various Lean practices

Essence: Lean is a mature and universal approach to processes optimization. Among others, base practices of Lean include:

- increasing productivity by removing "waste" (here "waste" is any segment or feature of any production and business process or supporting org structure that does not directly create value for the customer)
- non-stop improvement of processes and products in iterative cycles

Lean served as a foundation for many techniques and frameworks, including Kanban (see above).

Flavors and sources: Lean principles are originated from Toyota Production System (TPS) developed at Toyota mainly after WW2 (in the 1950s). TPS has later evolved into Lean Manufacturing. Lean ideas successfully survived through the years giving special flavor to many new management theories (for example, Lean Six Sigma, which was popular some time ago).

Lean Manufacturing much influenced Agile frameworks too. For that reason, Agile and Lean have many similarities, including:

- work is split into minimal rational increments
- processes use "pull" methods instead of "push"

At the same time, the combination of Lean and Agile might seem counterintuitive. Original Lean Manufacturing targeted repetitive mass production processes, while Agile is to improve the creation of new, innovative products and services. Nevertheless, Lean principals served well when experts tried to decompose creative work into formalized and repetitive steps, forming the Lean Startup methodology. Within this methodology, products are developed in short iterative cycles. Each cycle, we check one or several ideas or hypotheses by making lightweight prototypes and putting them through instant field testing, preferably involving customers.

To get familiar with these practices, see the book "The Lean Startup" written by Eric Ries and influenced by Steve Blank. Do not be confused by the name, as methodology equally applies to both startups and big corporations when creating new products or services. For the practical application of Lean in software development, see "Lean from the Trenches" by Henrik Kniberg. For an in-depth and transformation-oriented view, see "Lean Enterprise" by Jez Humble and other authors.

Ingredient 4: Concept of the autonomous product team

Essence: *The small autonomous self-managed cross-functional product team* (or simply *product team* further on) is often considered a central element of the new organizational structure. A cross-functional product team can be viewed as an evolved version of *the feature team* mentioned above. By its role, such a team is similar to a separate business unit:

- Product team works exclusively on a specific product (or a discrete part of a product)
- Product team is given specific business tasks to solve
- Product team is empowered to find and execute the right solutions (all decisions on how to solve a business task are made inside the team)
- Product team is accountable for business results

Flavors and sources: Many experts mentioned in this article, such as Eric Ries, Henrik Kniberg, Jez Humble, Marty Cagan, Aaron Dignan, and also many others, describe an autonomous product team in various ways and under different names. The most common term is straightforward — "product team." One of the exotic variations is the SLAM team (self-managed, lean, autonomous, and multidisciplinary).

Marty Cagan, a big advocate of the concept, came up with the term empowered team, which he popularized in the recent book "Empowered". This book is a good place to learn about such teams and the specifics of the organizational environment they need to thrive.

The benefits and challenges of the autonomous product team concept are also discussed in Part II of this article.

INGREDIENT 5: Concept of flattened self-managed organizations

Essence: In theory, self-managed organizations should be the next logical step in the evolution of self-managed teams. The way they are seen today, these organizations should implement some radical practices, such as:

- Removing most of the levels of management
- Replacing management positions with a set of roles assigned to some team members (one person can carry multiple roles)
- Maintaining complete transparency, which includes disclosure of everybody's compensation and corporate financials to all the staff

In reality, these practices appear too extreme for the vast majority of today's organizations (see more on this in Part II). Nevertheless, a major shift in this direction looks inevitable. Any organization that wants to base its transformation on autonomous product teams will be forced to make significant changes in its org structure and process, including (at minimum):

- Migration to distributed decision-making. Shifting the considerable part of decision-making from the top of the hierarchy to the team level is necessary for autonomous teams to function.
- Establishing a considerable level of corporate information transparency. Broad context (including much of financial data) is essential to support distributed decision-making.

Flavors and sources: There is a selection of more or less developed theories of self-managed organizations:

- Teal Organizations (see "Reinventing Organizations" by Frederic Laloux)
- Holacracy (see "Holacracy: The New Management System…" by Brian J. Robertson)
- Adhocracy
- Sociocracy
- several others

For helpful insights and stories of self-managed organizations, see articles "Beyond the Holacracy Hype" (by Ethan Bernstein and others, published at HBR) and "How to Choose a Model of Self-Organization..." (by Aaron Dignan, published at Medium).

If you consider implementing self-management principles and practices in your organization but want to find a way that suits your organization's culture and environment, a good place to start is the book "Brave New Work" by Aaron Dignan.

Ingredient 6: Goals & actions alignment frameworks

Essence: When an organization starts moving from "legacy" command & control management to distributed decision making and simplified org structure, it needs a new way of coordination and alignment of teams' efforts. Today, the most popular and practical way of solving this challenge is using an alignment framework, similar to OKR (Objectives and Key Results).

OKR framework provides a structured way:

- to decompose and cascade core goals (Objectives) from company leaders through all layers of management down to every team on the ground
- to define core actions that should bring explicit, quantifiable results (Key Results) which are necessary to reach Objectives

Usually, OKR assumes that Objectives go from top to bottom, although upstream movement is also possible. Key Results are defined by bidirectional communication between a team (or individual) responsible for the task and their supervising manager.

Flavors and sources: OKRs were first created by Andrew Grove, top manager (COO, CEO, Chairman) of Intel Corporation. Later the framework was widely popularized by venture investor John Doerr, who learned it directly from Grove while working at Intel. The list of OKRs adopters includes Google, Amazon, Netflix, Uber, LinkedIn, and many others.

OKR methodology can come under different names in different organizations. For example, at Adobe, it is called "Goals and Expectations." In most cases, OKRs are implemented in a customized, organization-specific way.

For the history of OKRs and implementation cases, see John Doerr's book "Measure What Matters" and site whatmatters.com. For the origin of the OKR concept and other management insights, see Andy Grove's "High Output Management."

INGREDIENT 7: New product management and marketing techniques

Essence and sources: Modern product management and marketing practices include the techniques for close customer interactions, methods for controlling and adjusting product lifecycle using products' and processes' metrics, plus frameworks for business-model design and correction. Although these are essential elements of the Agile way of working, they are not covered in this article to keep its size reasonable. Nevertheless, if you are unfamiliar with the concepts, see "The Lean Startup" by Eric Ries already mentioned above, "Working Backwards" by Colin Bryar and Bill Carr, and "Inspired" by Marty Cagan.

How true is the recipe?

Why has the industry consensus formed around particular frameworks and practices listed above? And, at the end of the day, why should we use them? The more or less sensible answer to these questions can be summarized as "That is how Big Tech Giants and other young disrupters are built. If you want to survive, you need to be built competitively." For example, the summary of Marty Cagan's "Empowered" starts with the question "What is it about the top tech product companies such as Amazon, Apple, Google, Netflix and Tesla that enables their record of consistent innovation?" For Marty's consulting team, the answer is clear — for the modern organization to thrive, it must be built based on empowered product teams.



Source: *"Ordinary People Extraordinary Results" by Marty Cagan at Lean Product Meetup* video, Dan Olsen's Channel, YouTube (2019)

Marty Cagan introduces the ideas of his new book at Lean Product Meetup

While “see how they do it and do the same” might sound like an attractive and possibly rational approach, in reality, every successful company is inevitably built in its own unique way. Nevertheless, we can learn much from all industry’s best practices and failures. This is what we would look at in the following three parts of this article.

Key takeaways from Part I

- In the modern world, there is a distinguished trend of disruption. Old-school companies lose business to new tech companies or even start-ups as some emerging technology radically changes a specific business area.
- Trying to save their organizations, leaders push for internal Agile transformation campaigns. When such a campaign is planned, the ultimate target is to match the way new tech companies (and especially Big Tech) are built.
- Article summarizes the list of concrete techniques, frameworks, and methodologies that define how new tech companies are supposedly built. In part, those techniques come from Agile methods of software development, in another part from Lean manufacturing methods, plus there are some relatively new techniques for alignment and coordination.

Part II: Walking Around the Traps

Benefits and challenges of the autonomous product team concept

As was mentioned in Part I, the concept of *small autonomous self-managed cross-functional product team* seems to be a core ingredient of almost every recipe of modern organizational transformation (see details in Part I). It is safe to say that such teams exist in many IT companies for a long time already. Nevertheless, we know that any promising enhancement usually comes at a certain cost. Let's try to determine the main benefits and drawbacks of the organizational structures and processes based on autonomous product teams compared to functional structures that some call "legacy" today.

Core benefits

Flexibility and agility

Flexibility and agility are probably the most significant benefits of granular organizational structures based on autonomous product teams. Such structure allows you to rearrange your forces very fast, on a small or big scale, whenever you need to invest more resources in a specific product (project, service, functionality) or go for a rapid business model pivot.

Why is flexibility so important? Because we see an increasing number of cases demonstrating that the organizations slow to adapt do not survive in a rapidly changing environment.

One colorful example opens a remarkably honest book, "No Rules Rules," written by Reed Hastings, CEO of Netflix and co-authored by Erin Meyer. We see a frightening story of Netflix's biggest competitor, Blockbuster. Blockbuster refuses to buy Netflix in 2000 for $50 million. Later, at the time of Netflix's IPO in 2002,

Blockbuster is still literally a hundred times larger ($5 billion in oppose to $50 million). Nevertheless, in 2010 Blockbuster went bankrupt, as it was "unable to adapt from DVD rental to streaming." Further on, Hastings continues:

"Blockbuster's story is not an anomaly. The vast majority of firms fail when their industry shifts. Kodak failed to adapt from paper photos to digital. Nokia failed to adapt from flip phones to smartphones. AOL failed to adapt from dial-up internet to broadband. My own first business, Pure Software, could not adapt to changes in its industry either because our company culture wasn't optimized for innovation or flexibility."

That is probably enough reasoning to consider flexibility as a critical factor. We will also look more carefully at the industry shifts — in Part III.

Rapid scalability

Rapid scalability is another benefit you might gain with the autonomous product teams approach. Today, it is critical not just to bring an unmatched product (or service) to the market but to bring it a bit earlier than your competitors can. Having a sufficient amount of resources and granular structure, a company can allocate a large number of product teams to work on a specific product or feature. Still, there is the difficulty of decomposing this product or feature into smaller components with minimum dependencies. But if that is solved, the organization can put a bet, invest massively and deliver a product very fast, potentially winning a race. Functional structures, even if fine-tuned, can hardly scale fast enough due to overall organizational complexity.

A remarkable case of such rapid scaling, we find in the recent book on Amazon's way of working — "Working Backwards" by Colin Bryar and Bill Carr. Bill Carr was one of the leaders who started Amazon's new Digital Media and Devices business in 2014. The company was betting big on digital, so the business unit needed to grow fast at certain moments. Carr uses the example of the video client application development to illustrate how this growth was happening:

"In 2004, video client application development was handled by one two-pizza team. It then became three teams, one for web, one for mobile devices, one for TV devices. Then the mobile team became four teams (iPhone, Android phone, iPad, Android tablet) and the TV team became five-plus teams (Xbox, PlayStation, TiVo, Sony Bravia, Samsung, etc.)…"

Amazon is known as one of the innovators of the small autonomous product team concept, and the "two-pizza team" is Amazon's name for such a team. The idea behind the name is that there should be no more than ten people in the team so that you could feed them with two large pizzas.

Unleashed creativity and performance

The shift of management approach from command & control to distributed decision making by self-managed teams assumes the involvement of all team members in product design and customer relations. If you manage to go through such a shift in a true way, your people will gain:

- strategic context and knowledge of business specifics
- feeling of ownership for the product and emotional involvement in the customer's needs
- sufficient information on how the product progresses, based on the product's metrics
- flexibility on how challenges can be resolved

All these factors together can considerably improve people's motivation and, in turn, lead to a boost in productivity and innovation.

Minimized coordination efforts

A distinct difficulty of a sizable organization is the amount of coordination necessary to get anything done. At a certain scale, a company is often forced to add a project management office or other coordination superstructures to support new undertakings and even everyday work. With the autonomous product team approach, coordination of that type is in big part unnecessary. When supported with the right technological and organizational environment, teams do most of their work independently, not relying on other teams or other parts of the organization. Thus, teams can move faster, especially when developing new products or adding new functionality.

Good fit for new product management techniques

Autonomous cross-functional product teams are a native part of many new product management techniques and processes, especially when it comes to new products and services development. Since, as mentioned earlier, product management topics are not covered in this article, please see the sources referenced in Part I for more information.

Challenges and drawbacks

No cost reduction

Any organizational change requires considerable short-term investment. To compensate, previous generations of transformation processes promised us a reduction of costs at the end. Many evangelists would disagree, but after the Agile transformation, the investments in the product development process will likely increase, not decrease. In other words, outgoing functional structure (if fine-tuned) could perform more cost-efficiently, benefiting from "economies of scale" effects. For example, in software development, you could support a large codebase with minimum number of engineers. Granular Agile structure inevitably brings some duplication of work and other inefficiencies "by design." Think about a T-shaped (having some skills in related fields) specialist working at his secondary area of expertise compared to a functional specialist. Simply put, with cross-functional teams, we optimize the system for speed and agility, not for overall productivity.

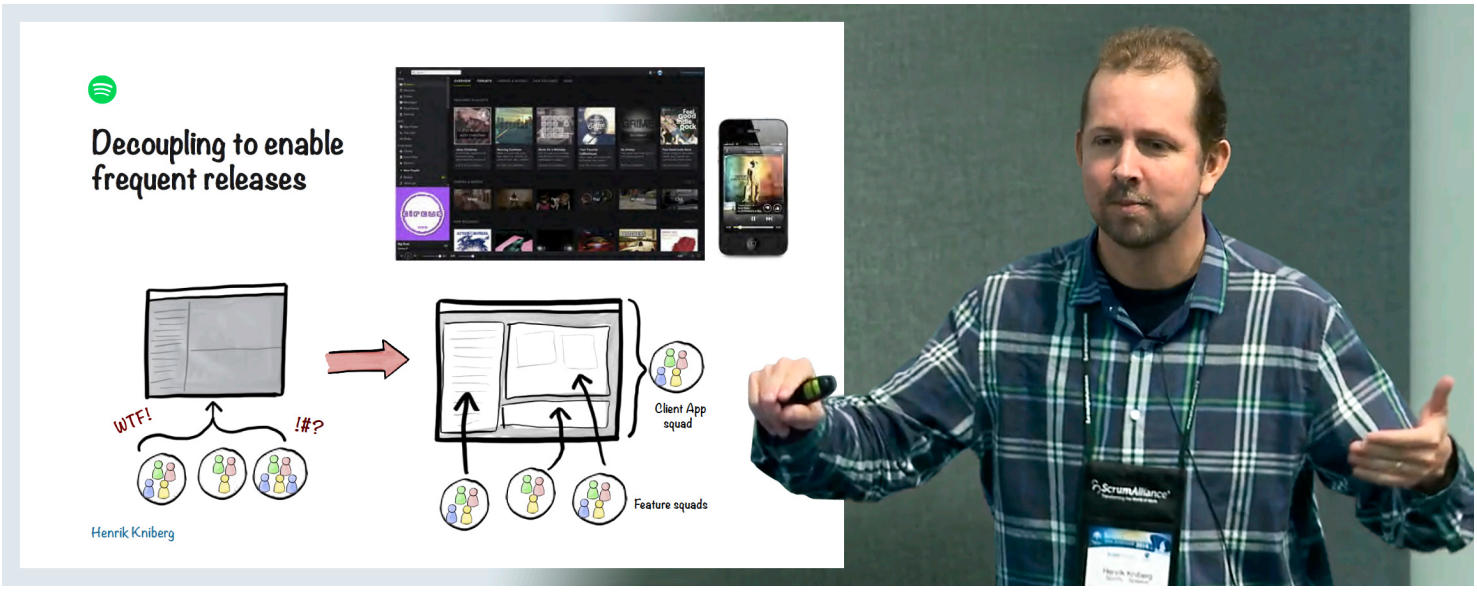
Nevertheless, under certain circumstances, cost reduction is still possible:

- If you are going through the first transformation steps, implementing some Agile techniques on a team level only. At this stage, you should see the rise of productivity even within a functional structure. Unfortunately, without expanding the change process to the level of the whole organization, you have very little chance to reach all benefits described in the sections above.
- If your organization was very "spongy" before the transformation. For example, it had lots of bureaucracy, redundant structures, or inefficient processes. In this case, Agile transformation can make the organization more nimble and more cost-efficient simultaneously.
- If you are putting old-school "analog" business through the digital transformation. If done well, such change should bring you to better efficiency of all types as well.

Rebuilding product to match org model

One of the main factors required for autonomous cross-functional teams to work effectively is a technological base of the product(s), eliminating or, at least, minimizing the dependencies across the teams. For example, in software development, when you have several teams working on the same product, this product must be divided into virtually independent components. For the server-side software, this is often accomplished by using microservices or service-oriented architecture (SOA). For the client-side software, similar results can be achieved when creating a shell application with different functions implemented as modules "hosted" in that shell.

In Part I, we have met Henrik Kniberg, a strong practitioner who worked as an Agile coach and change agent in Spotify, Lego, and several other organizations. During his talk at a conference, Henrik explained how Spotify changed the architecture of its desktop client application to match autonomous teams' areas of responsibilities.



Source: *Scaling Agile @ Spotify with Henrik Kniberg* video, Scrum Australia Channel, YouTube (2015)

Henrik Kniberg describes changes implemented in Spotify desktop client software during his talk Scaling Agile @ Spotify

Depending on the nature and history of your product, investments in changing product's architecture might be very considerable, often comparable with rewriting the whole codebase from scratch.

A remarkable example of such massive investment is the story of Amazon rebuilding all core segments of its IT platform from monolith to service-oriented architecture (while running its day-to-day business in parallel). Slightly different variations of that story we see in books "Working Backwards," mentioned earlier and "Lean Enterprise" by Jez Humble and others, referenced in Part I. Interesting technical details of the process we can also find in the ACM Queue interview with Amazon's CTO, Werner Vogels. For Amazon, the transition to SOA had started in 2001 as a part of the implementation of "two-pizza team" and later "separable, single-threaded teams" (explained below) concepts. Colin Bryar and Bill Carr remember: "The path <...> took almost a decade, in large part because it required that we first untangle our monolithic software architecture and the organizational structures that had grown alongside it, then replace both, step by step, with systems designed to support rapid innovation".

Limited team size and specialists' shortage

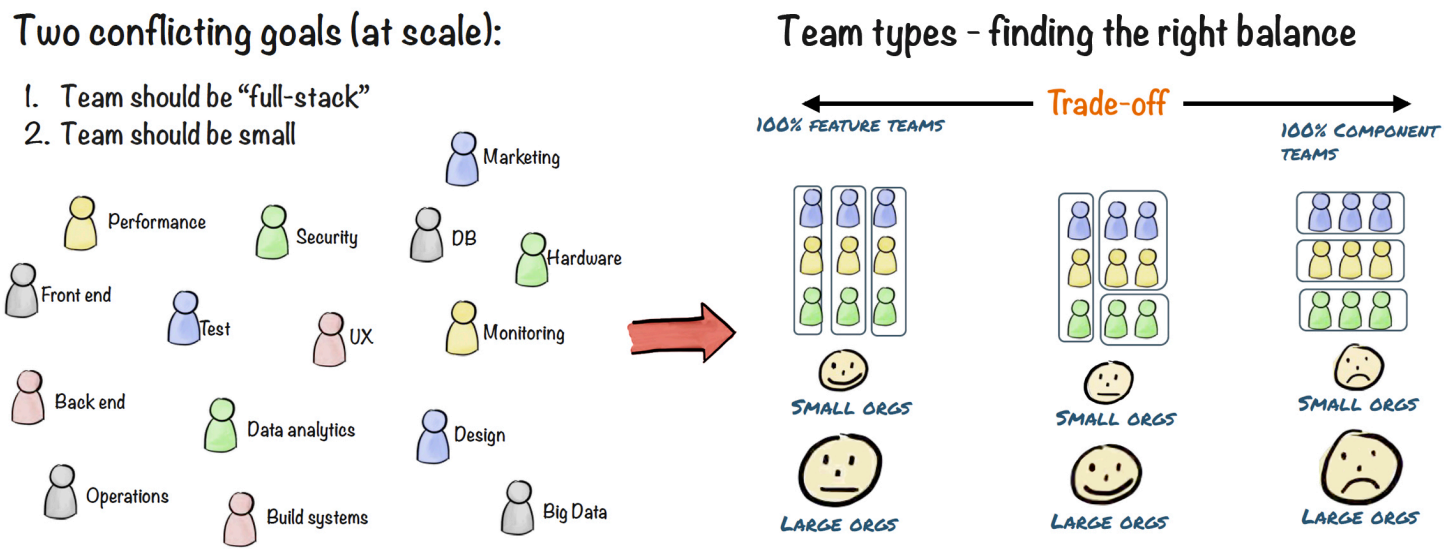
As we saw in Part I, the assumed evolution of "legacy" organization first goes from functional division structure to cross-functional structure based on autonomous teams. At this phase, such teams are often called *feature teams*. The next upgrade (which today should probably be done in parallel) is empowering feature teams and converting them into true *product teams*.

The first challenge we meet as we go through this transition is an obvious side-effect of the limited team size (no more than 9-10 people). A proper cross-functional team should include at least one specialist of every area of expertise. To start with, you might simply lack enough people of particular specialization. But even when you hire more people, the variety of specialists needed might not allow you to fit them all in a small team.

Vincent van den Boogert, one of the key change leaders in ING Bank in the Netherlands, describes this challenge in the following way (detailed story of ING transformation see in Part IV):

"People come here and bring their expertise. But if the norm is multidisciplinary teams, how do we guarantee their craftsmanship in three years from now? For example, they will all be customer journey experts, not marketers. In the old days it was relatively easy because we could have a department with six social media campaign experts. But now I have 300 squads working autonomously — I cannot have a social media expert per squad. And agile means that we cannot make a pipeline that requires people to pass through social media experts. This question is puzzling us a bit."

One reasonable but compromise way of solving this particular challenge (and some similar ones, as described later) is forming a hybrid structure. Both the team size limitation and compromise of the hybrid structures are well demonstrated by two more of Henrik Kniberg's clever illustrations — "Two conflicting goals (at scale)" and "Team types — finding the right balance."



Source: Henrik Kniberg, illustrations for the slides displayed in *"Real-life Agile Scaling" by Henrik Kniberg* video, Agile66 Channel, YouTube (2019)

Finding the right balance between cross-functional and functional teams — two slides from Henrik Kniberg's keynote at Agile Tour Bangkok

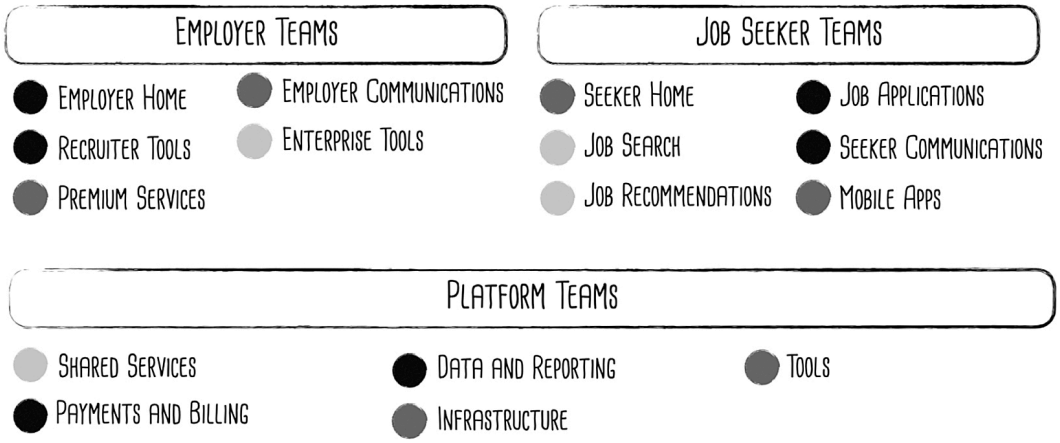
Functions you do not want to distribute

There is also another factor that can force us to create a hybrid structure. Any organization has particular functions that become ineffective if split between cross-functional teams (e.g., accounting). In software development specifically, you will likely see the need for dedicated functional teams responsible (at least) for infrastructure and base software layer that is often called platform. Those teams' job is to create, maintain and improve different parts of technical infrastructure, shared tools, and highly specialized components.

A good example of real-life balance between teams of different types, we can see in "Empowered" by Marty Cagan and Chris Jones (in the book, look for the "Case Study" section). Authors describe an online jobs marketplace company. The company's name is not disclosed, but Marty ensures us that this is a real-life example. This organization has sixteen product-related teams in total. Eleven teams are responsible for different parts of customer experience and are divided into two groups by customer type — *Employer Teams* and *Job Seeker Teams*. In particular, one of those eleven teams, the *Recruiter Tools Team*, is supporting "advanced capabilities for loading and managing large numbers of open jobs and managing the flow of applications, interviews, and decisions."

The other five teams make the group of *Platform Teams*. These teams are not directly involved in forming customer experience but rather help customer-facing teams to do their job best. For example, one of the Platform Teams — *Payments and Billing* "handles all financial transactions including recurring

payments, discounts, promotions, and the various payment methods." This functionality is kept in a dedicated team because it is complex and highly specialized. In this way, "other teams don't have to understand this complexity."



Source: Illustration for the book of Marty Cagan and Chris Jones, *EMPOWERED (Silicon Valley Product Group)*, Wiley, Kindle Edition (2020)

Teams topology for an online jobs marketplace company — the example from the book "Empowered" by Marty Cagan and Chris Jones

Interestingly, several challenges we just looked at (team size limitations, specialists' shortage, dealing with functions we do not want to distribute) lead to the same hybrid org structure, combining functional and cross-functional teams. Although effective, this solution comes with its own challenges we need to consider carefully:

- Functional teams are there to enable product teams to create value for a customer. Unfortunately, this makes functional teams less involved in the final result. To compensate, we need to find a good way to motivate functional team specialists and the right metrics to evaluate their work. For example, we can use the "internal customer" approach, when a product team acts as a customer of a platform team while goals and key results are shared among teams. In parallel, this platform team might have its own goals, such as creating or refactoring a specific platform component.
- Functional platform teams are the natural source of extra dependencies, as they work to support many teams simultaneously. Here, unfortunately, the more functional teams we have, the more flexibility we potentially lose.

Challenges of product or opportunity scale

Let's also look at the product and team balance from the product size perspective.

- **Cannot make product small enough.** When we begin a transition to the structure based on the autonomous product teams, we need to define what segments of the company's offering can be separated and shaped into

independent products. The most puzzling part starts when we realize that our products are still too big for 9-10 people teams — after all the efforts to cut small. This is often the case for banks or telecom companies, for instance. Such a contradiction often forces organizations to depart from the "one product — one team" principle, losing much of the expected flexibility. Some Agile at Scale frameworks (see the list in Part I) were designed to minimize the negative effects of such a scenario. See examples in LeSS implementation stories in Part IV.

- **Do not have enough leaders for every small team.** That bottleneck is well illustrated by the story of Amazon's conversion to the two-pizza team concept. At some point, the company realized that it does not have enough leaders with the right capabilities. Or, in other words, it became clear that a senior leader who can act as a "CEO of a product" can develop a much more significant business area than one small team can cover. Here is how the book "Working Backwards" describes Amazon changing its focus from the team size to leader's role:

"But we later came to realize that the biggest predictor of a team's success was not whether it was small but whether it had a leader with the appropriate skills, authority, and experience to staff and manage a team whose sole focus was to get the job done. Now free of its initial size limits, the two-pizza team clearly needed a new name. <...> Thus, "single-threaded leaders" and "separable, single-threaded teams" were born. <...> The [single-threaded leader] extends the basic model of separable teams to deliver their key benefits at any scale the project demands. Today, despite their initial success, few people at Amazon still talk about two-pizza teams".

New silos drawbacks

One of the core reasons for improved productivity and agility of the org structures based on autonomous cross-functional teams is the lack of "legacy" functional silos. Simply put, when people of different specializations constantly communicate as a part of a single team, complex processes like product design move much faster. At the same time, when a team gets self-sufficient, it loses (at least partially) motivation to communicate with other parts of the organization. This new isolation factor has to be compensated with changes in the processes, structure, and culture. Otherwise, you might experience major problems with organization's overall technical expertise and product quality.

- **Problems with technology sharing and expertise collection.** Generally, an autonomous product team is free to choose which "tools" to use. If we take software development example, each team might want to use a different programming language, set of components, or database for similar tasks. Although this approach increases flexibility and encourages

experiments with new technologies, it can damage the organization's expertise level. Such expertise is mainly built by intensive knowledge sharing among functional specialists on a particular subject (for example, programming language specifics). That is the way best practices are naturally formed. If engineers use a large variety of technologies or do not share knowledge effectively, the organization as a whole acts inefficiently, learning many technologies on a poor level instead of mastering a few selected ones. Also, fragmented technology stack makes it difficult to move engineers from team to team and split work among teams.

To some degree, these challenges can be solved by additional voluntary structures, like *Communities of Practice* with are part of LeSS and SAFe frameworks. However, in Spotify Model, we will later see the combination of professional communities (named Guilds) and unconventional matrix structure joining product teams (functional specialists from different teams are forming Chapters). This combination looks like a more viable approach. The details on Spotify Model see in Part IV.

- **Problems with product usability and quality standards.** When several autonomous teams develop the same product (or product suite), this might lead to significant differences in product quality standards and user experience specifics. Taking the Spotify desktop application example we saw earlier, each team might be responsible for a specific product component, while each component has its own segment of the user interface. In this case, if teams have different ideas on how a user should interact with the product's interface, this user would feel very awkward while using this interface.

  To address those issues, you would need supervising and coordination forces, such as engineering or product design leaders, or, at least, expert committees. Although the supervision will decrease teams' flexibility, it is essential for uniform user experience and consistent product quality.

- **Difficulties with career path and professional growth.** Conventional functional structure (where programmers are teaming with programmers and designers are teaming with designers) can provide an efficient environment for professional growth. You start as a junior or an intern and then, with enough effort, grow into an expert. In contrast, a cross-functional team expects you to be a well-experienced professional from the very start, better T-shaped.

  Professional development issues can also be resolved, at least partially, with the help of structures like Chapters and Communities of Practice. But overall, the ability of an Agile organization to hire people for potential and let them grow from the entry-level positions looks somewhat limited.

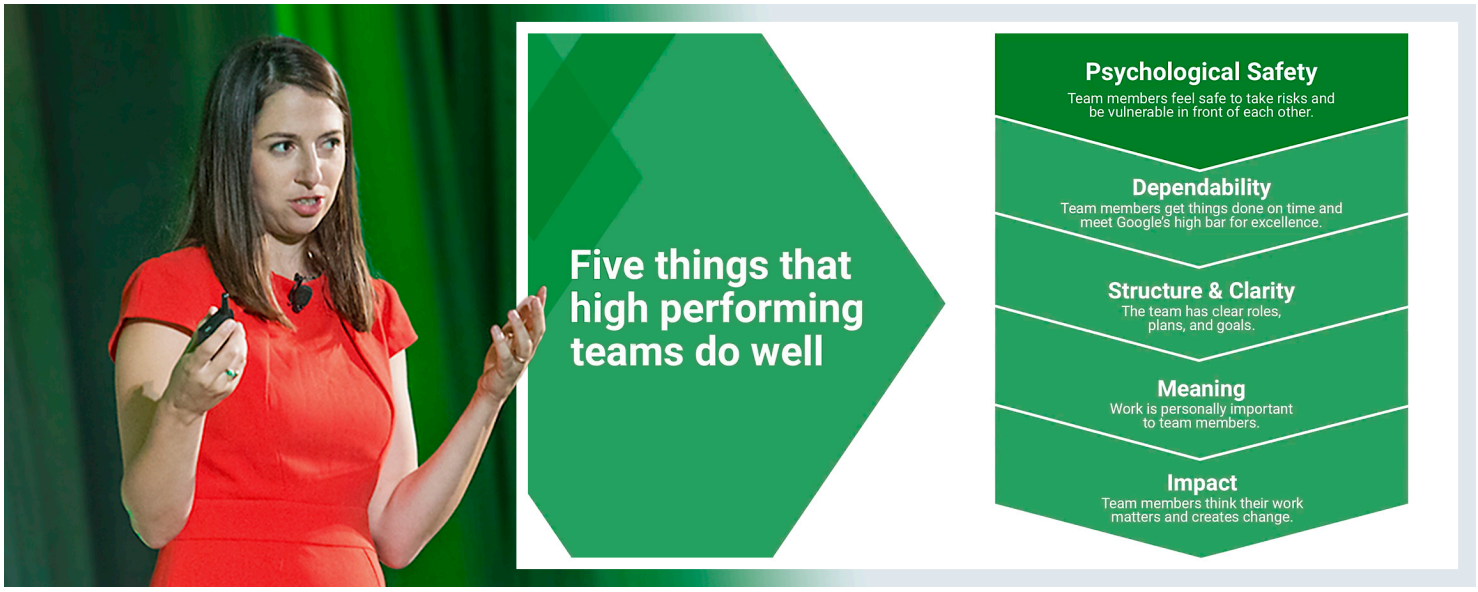
Contradictions

As noted in Part I, there seems to be a consensus among experts on the core characteristics of the modern developing organization. However, the views on some critical details might be radically different.

Ordinary or extraordinary people

The signature subtitle of Marty Cagan's "Empowered" is "Ordinary People, Extraordinary Products." The ideas behind this subtitle can be summarized as follows: you can put ordinary specialists together to form a good team; give this team enough authority; provide it with the proper training and support; organize and maintain a productive environment in your organization; then (if you succeeded with all the steps) your ordinary people will absolutely shine and show extraordinary results. I am a big supporter of this idea, as I was lucky to see this happening in reality. There is also "Project Aristotle," extensive research conducted by Google on what makes its teams most effective. Two important facts were among this research's findings:

- Psychological safety appeared to be the most important factor of team's effectiveness
- There were no direct correlation found between the number of extraordinary "rock star" individuals in a team and the team's effectiveness

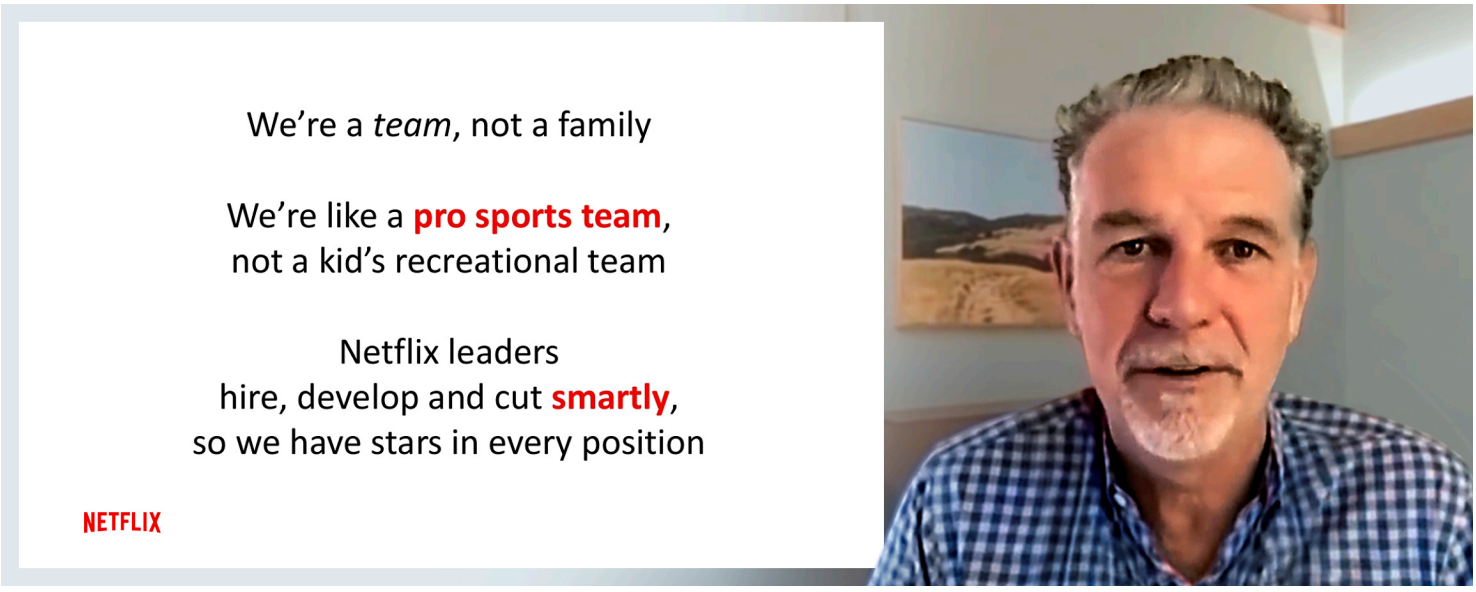


Source: *Julia Rozovsky How to Build a Great Team* video, re:Work with Google Channel, YouTube (2017)

Julia Rozovsky, a researcher participated in Project Aristotle, explains the project results and answers the question on "rock star" individuals

At the same time, the culture code of Netflix is based on a directly opposite concept. From "No Rules Rules," we learn that one of the cornerstones of Netflix's success is a concept of "talent density." But, how is the necessary level of talent density reached?

- Netflix hires exclusively best-of-breed extraordinary specialists and "pay [them] at the top of the market."
- Netflix practices the Darwin-style selection process. Suppose, at any given point of your career, your boss does not believe you are performing on an extraordinary level. If so, he or she should fire you as soon as possible "with a generous severance package."



Source: *Netflix CEO Reed Hastings: Marriage counseling showed me I was a 'systematic liar'* video, CNN Channel, YouTube (2020); Reed Hastings, *Culture* slide deck, Version 2, SlideShare (2009)

Reed Hastings, in an interview to CNN, argues that Netflix culture is not equal to "Hunger Games"; a slide from the original Netflix Culture Deck

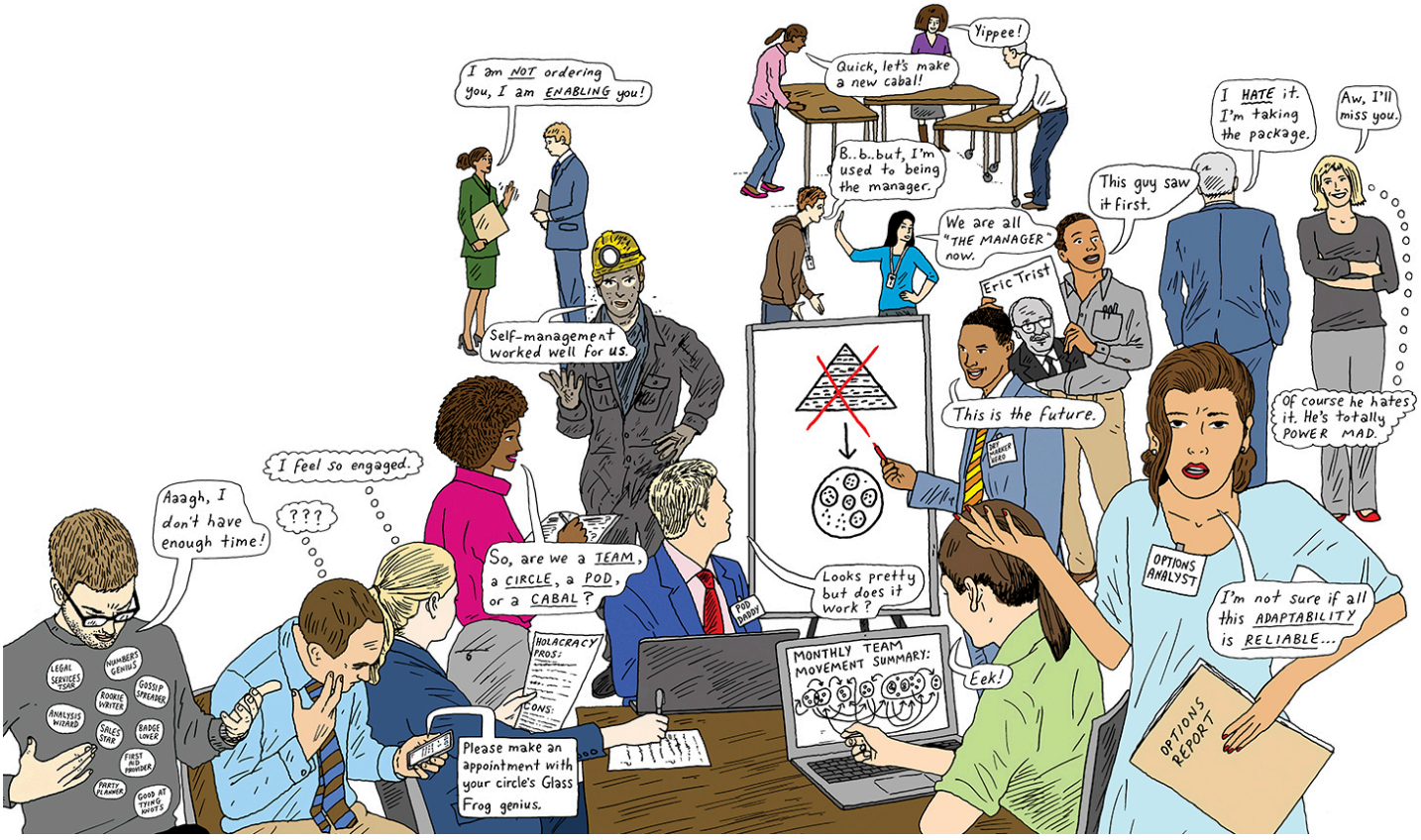
Apparently, Netflix, which was (and is) a source of inspiration for many, does not believe that ordinary people can create extraordinary products. At the same time, other leading organizations believe otherwise. Every transforming organization should probably remember this contradiction and form a custom HR strategy based on its own cultural values.

Hierarchical or decentralized structures

Any big organization going through modern transformation (or a new organization overgrowing a particular size) faces the challenge of scaling the formula of autonomous product teams. As already noted in Part I, this challenge does not seem to have any common resolution yet.

On one side of the spectrum, we have "good old" centralized hierarchical structures. As Aaron Dignan emphasizes in the opening part of his book "Brave New Work," such structures did not change much for at least a hundred years. So, it is highly unlikely that hierarchies can fit today's dynamically evolving reality without considerable changes.

On the other side, we have a variety of flattened, decentralized, self-organizing management concepts, from *Holacracy* to *Evolutionary Organizations* (see some details in Part I). The problem with those great concepts is that just a handful of companies apply them in the real world. Most authors keep talking about the same iconic examples — Valve, Zappos, Morning Star, and few others. Holacracy has a somewhat longer list of official adopters. But even that list has just 25 organizations from all over the world with a size of over 200 people (as of August 2021).



Source: Peter Arkle, illustration for the article by Ethan Bernstein, John Bunch, Niko Canner, and Michael Lee, *Beyond the Holacracy Hype*, Harvard Business Review, Harvard Business School Publishing (2016)

HBR article "Beyond the Holacracy Hype" argues that most organizations should adopt only a limited amount of self-management and in a way it suits them

None of Big Tech Giants and other new champions is known to use much decentralization on a corporate level. It is safe to assume that they actually delegate considerable authority to the product teams. But, other than that,

each of Big Tech Company likely has a unique combination of structures and practices. For example, both Netflix and Spotify practice radical transparency. But if Spotify is known for minimizing the number of management layers (see more in Part IV), Netflix, as well as Amazon, has an organizational structure closer to the "classic" multilayer hierarchy. If we take Google, it has probably gone even further from decentralization and self-management. Company is often criticized for becoming another "big evil corporation." Some of Google employees even formed a trade union in January 2021.

Nevertheless, if a transforming organization tries to preserve its existing hierarchy (and bureaucracy) from significant changes, it will most probably fail the transformation. As described earlier, the autonomous product team concept requires teams to be empowered and aligned to reach agility and boost innovation. Here again, every organization would have to find the right style of leadership and individual balance of the authority distribution — depending on the organization's culture, capabilities, and business context.

Key takeaways from Part II

- Small autonomous cross-functional product team concept is probably a central element for an Agile organization. Potentially powerful, this concept brings along substantial benefits, significant challenges, and puzzling contradictions.
- Main benefits of the concept include:
    - Flexibility, as the ability to shift and concentrate resources on a specific product or business area
    - Scalability that allows adding more teams promptly when rapid growth is needed
    - Increased creativity and performance, since people are better motivated, more involved with product development and customer relationships, plus use metrics to measure the progress
    - Good fit for new product management techniques
- Among potential challenges of the concept, we saw:
    - With a few exceptions, Agile transformation does not promise any reduction of costs (rather opposite).
    - Major investments are required in products' refactoring to support new org structures and processes
    - Limitations of team size enforce making products granular or go for alternative org structures when products cannot be made granular enough

    - New silos (that can potentially emerge) threaten an organization's integrity, technology sharing, best practice collection, product's consistency, and people's career paths
- Possible solutions for the challenges could be matrix org structures, professional communities, supervising and coordination forces, and viable hybrids between cross-functional and functional constructions.
- Two contradictions related to the concept were noted:
    - The contrast in acceptance of the idea that ordinary people can show extraordinary results
    - The lack of industry consensus on hierarchical versus decentralized organizational structures

Part III: Skeletons in the Closet

Why transformation emerges, what it needs to succeed, and how it may become fatal

Although the Agile transformation is a relatively recent invention, organizations tried to reform themselves, hoping to become more efficient (or simply survive) all the time they exist. Perhaps, diving into ancient history is not entirely relevant here. However, looking at the time span of several decades, we find the dramatic stories of the change process emerging, succeeding, or failing, plus we get a chance to indicate the core reasons for such successes and failures.

Intel story: Only the paranoid survive

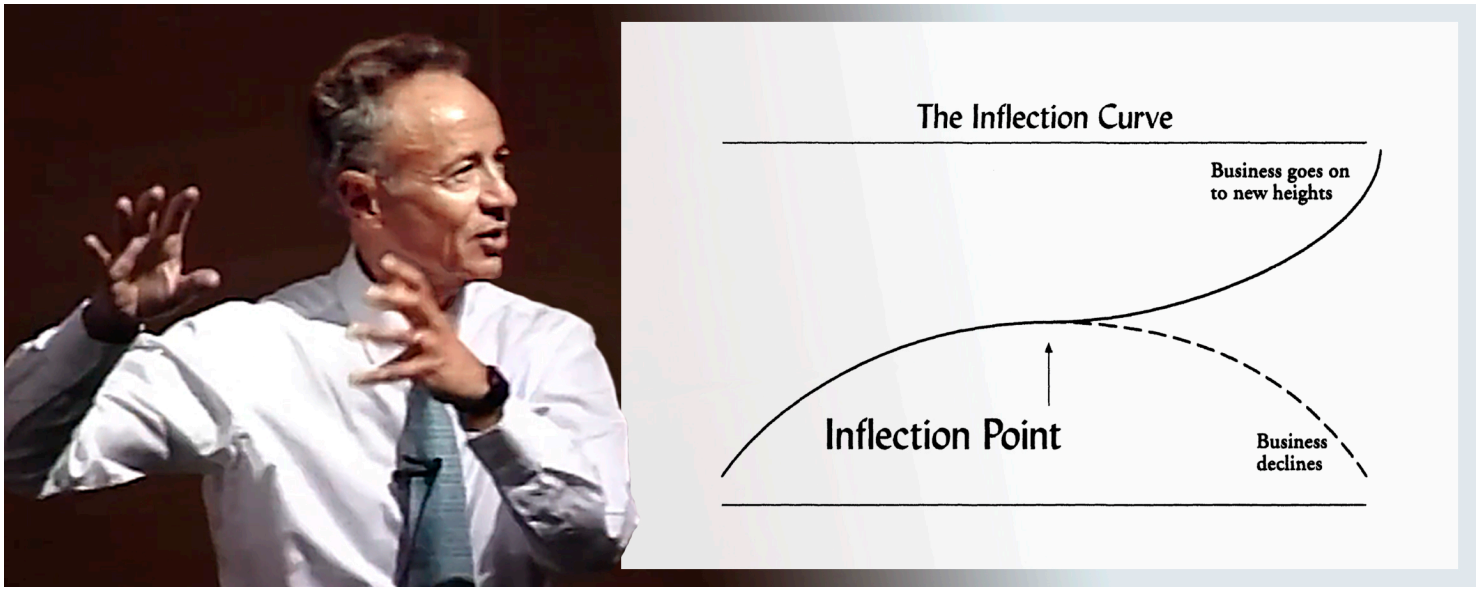
We tend to believe that our time is the most revolutionary. But the fear of disruption is anything but new. The phenomenon, very similar to what we now broadly call "disruptive technology" or "disruptive innovation" (making original authors of the term somewhat unhappy), was first popularized in 1942 by Joseph Schumpeter as "creative destruction."

As noted earlier, this phenomenon often makes business owners and top managers deeply concerned. They see a significant threat of some radical innovation that another player would bring, dramatically changing the way their industry operates.

In the book "Empowered" that was already mentioned quite a few times in this article, Marty Cagan describes how he convinces organization leaders of the importance of technological innovation:

"Often, when I am having dinner with one of the CEOs from a company that doesn't get this [the role of technology] they'll tell me how they're not a technology company—they're an insurance company, or a health care company, or an agricultural company. I'll say, 'Let me tell you what I would do if I was a product leader at Amazon or Apple, and we've decided to go after your market because we believe it is large and underserved, and that technology is available that enables dramatically better solutions for your customers.'"

You could argue that such ideas might make the organization leader paranoid, and you would be right. Not all leaders can handle that type of pressure decently. But here we should remember that legendary Andrew Grove, Intel's leader for many years and father or OKRs (as described in Part I), wrote the book named "Only the Paranoid Survive." This book, first published in 1996, introduces the idea of "strategic inflection point" — the point in time when your industry environment is fundamentally changing and company must either reinvent itself or lose the business.



Source: *Andrew Grove, "Strategic Inflection Points" — 1996 MIT Industry Leaders Program Lecture* video, MIT Video Productions Channel, YouTube (2019); illustration for the book of Andrew S. Grove, *Only the Paranoid Survive*, Crown, Kindle Edition (2010)

Andrew Grove explains the concept of strategic inflection point during his lecture at MIT; illustration from "Only the Paranoid Survive"

One of the cases Grove described back in 1996 was related to fundamental industry change that he nearly missed himself, and that could take Intel out of business in 1984-1985. At the time, Intel's main business was in memory chips production, but the company was inevitably losing its market share to Japanese competitors. By the mid-1986, Intel was rescued by a strategic pivot, as it switched painfully from memory chips to microprocessors production.

In January 2021, Intel turned a new page in its history — the board of directors appointed Pat Gelsinger as new CEO. Gelsinger spent the first three decades of his career in Intel and was mentored by Andrew Grove. No surprise, Bloomberg's reporter named the article describing challenges for the new CEO "Only the Paranoid Survive: The Sequel." This time Intel is facing the major threat of losing the leadership positions in the microprocessors industry, which the company dominated for decades. By the end of March 2021, Gelsinger's team already had a new action plan that many experts and investors found convincing. Intel is a great technology company, and hopefully, it would manage to get through this new inflection point.



Source: *Intel Unleashed: Engineering the Future (Event Replay)* video, Intel Newsroom Channel, YouTube (2021)

Pat Gelsinger presents the strategic updates on a webcast and makes a statement that "Intel is back"

Adding Intel's story to other cases we saw in Part I and Part II, it becomes clear that leaders today have enough reasons to be deeply concerned (or even paranoid) about their business disruption. Those concerns often push leaders to start the Agile Transformation, hoping to make the organization more adaptive and flexible. However, as with any other significant change process, this transformation can face considerable internal opposition.

Opposition to change

The resistance to the change process can appear on different levels of an organization (top and middle management, key experts, teams, and individual contributors) and in various forms. Notably, people leading the transformation

and those who are opposing it might have very similar motivations. Individuals on both sides might be truly concerned about the organization's wellbeing and, at the same time, worry about losing their jobs, assets, or power. Understanding the reasons behind the resistance can help Change Agents to work with the opposition. Let's highlight few types of usual negative feedback, taking into account Agile Transformation specifics.

Let's just leave it as it is

This base objection can come from people at any level of organization and take forms "do not fix what works" or "that is above my paygrade." Also, you would often hear, "just let me do my job." For example, a software engineer could say, "I just want to code, but those endless meetings and discussions only disrupt my work." If, after looking at such a critic's motivation, you understand that all he or she wants is to keep the current status quo, it is not much you can do at the initial stage of the process. Whenever it is possible, try to keep such people aside, preferably as neutral. Some of them would come on board as the change process start moving for real. The same applies to other neutral individuals who neither support nor oppose the transformation at the beginning.

That framework is good for nothing

"This damn Scrum/LeSS/SAFe/Spotify (or any other methodology) is a curse. It would never work in real life." This reasoning sounds very similar to "let's leave it as it is." But as a matter of fact, that type of opposition can be potentially converted into productive cooperation — if we pay attention to the context.

Some of the Agile frameworks we saw in Part I exist for two decades already. Others are a bit younger, but they also have their history. As we would see shortly, any organizational change statistically has low chances to succeed, primarily because of poor management. So, many companies that failed their transformation efforts within the last decade possibly failed while enforcing one of the Agile frameworks. There is also the "bad news travels fast" effect. When, for example, some SAFe implementation fails, not just those who suffered but also a large number of people in their professional networks, conclude that SAFe is evil. No surprise that Scrum, as the most popular framework, seems to have an army of haters as big as the army of adopters.

So, how do we involve people who have a strong bias against a framework or methodology we plan to use? We should start from two-way communication and try to understand what stands behind our colleagues' skepticism. If skeptics rely on their or someone's real-life experience, this can be quite valuable input. As we communicate, we can customize our change plan or even methodology itself when necessary. Plus, within this process, we might convert our opponents to change supporters or even change agents.

We will shoot ourselves in the foot

There is a particular type of high-level objections you can hear precisely from managers, lead experts, or opinion leaders:

- "Scrum/LeSS/SAFe/Nexus (or any other) might be a decent framework, but if we implement it in our current situation, we will kill several processes we have and create several other significant problems." The concrete examples of such problems see in Part II.
- "How do we deliver what we promised to our customers on schedule if there are no deadlines in the Agile world?"
- "Whom do we hold accountable for the business results if all decisions are distributed in autonomous teams or communities?"

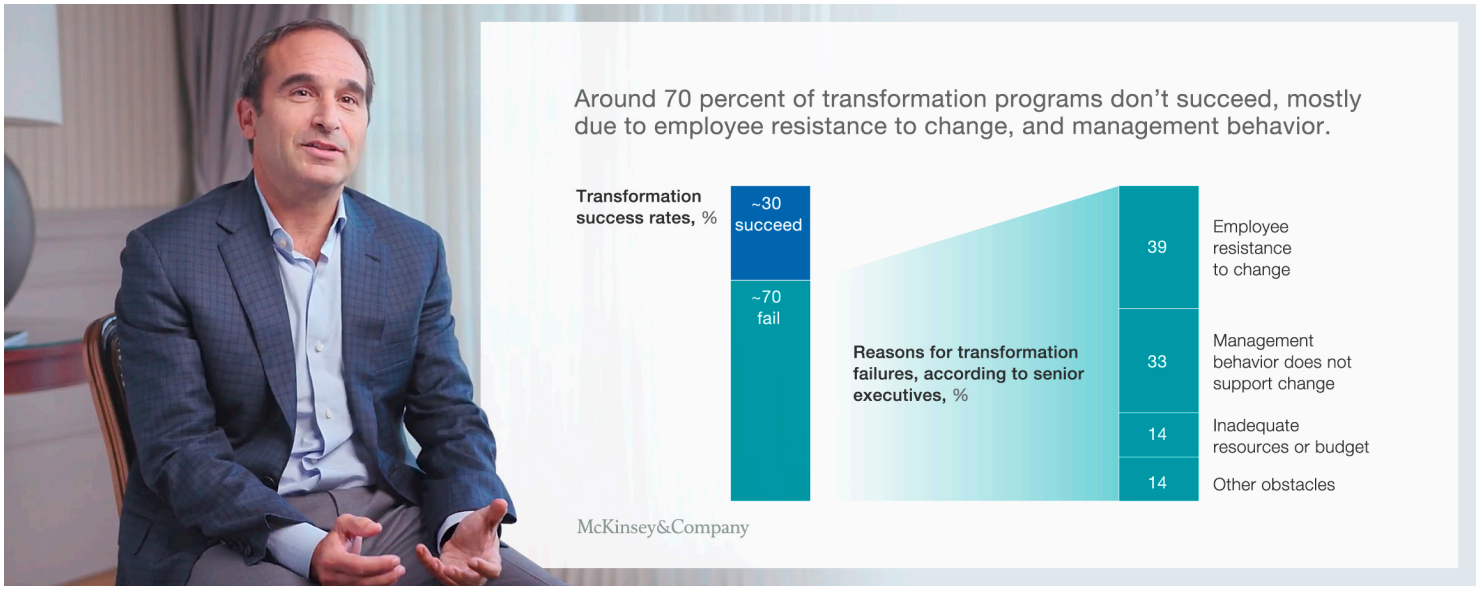
These are valid concerns that must be addressed. If we can involve such opponents in change process management, this will help us find the right solutions and, as in the previous topic, convert opposition leaders into engaged supporters, improving chances for transformation's success.

To sum up the "how to manage the opposition" ideas. Any transformation is pushed and powered by people who care — change agents. These people are emotionally and mentally engaged (see more on enthusiasm below). When change agents meet serious opposition, they should recognize that their opponents might also care about the same important things. If we manage to factor opponents' criticism into transformation planning and execution, we get a safer process and newly engaged supporters. Of course, there are also true antagonists trying to keep the status quo at all costs. There is no point in wasting much of our energy trying to persuade these people. It is better to keep them neutral or mitigate their influence when necessary.

Transformation essentials

For many years McKinsey runs analyses and research on transformation success. The research results match assumptions from many other sources: on average, at least 70% of large-scale change initiatives either fail entirely or bring unsuccessful results. Harry Robinson, who leads McKinsey's Transformation Practice (RTS) globally, in his interview names the root causes of transformation failures, all related to the leadership mistakes:

- Lack of aspiration
- Lack of shared vision
- Lack of engagement
- Low investment in capability building
- Insufficient structure



Source: *Why do most transformations fail? A conversation with Harry Robinson* video, McKinsey & Company Channel, YouTube (2019); illustration for the article by Tony D'Emidio and David Malfara *Build a case, build a following: Laying the groundwork to transform customer experience in government*, McKinsey Insights, McKinsey & Company

Harry Robinson explains the reasons for organization transformation failures; the illustration from the related article on mckinsey.com

As Mr. Robinson mentions closing this interview, McKinsey has created certain recipes for overcoming main failure causes. There are also tons of other valuable materials and even educational courses on the subject. Nevertheless, let's try to highlight some of the factors that are critical for transformation success but often missed or undervalued.

Set a compelling goal

Organization's leaders must define and communicate the attractive and motivating goal. The goal must be exciting enough to motivate people, clear enough to synchronize all core decisions, significant enough to answer the questions, "Why are we doing this? What is the purpose?"

- **Goal cannot be negative.** The fear of disruption might motivate leaders for some time but will not motivate the whole organization for a long time. You need a positive vision of the future to build new structures and processes.
- **Goal cannot be too abstract.** "Let's make the world a better place" might be a compelling goal only if individuals within the organization can map their everyday activities to world well-being. For most organizations, this is not the case.
- **Goal should be** either **quantifiable** itself, or you should be able to decompose it to the high-level quantifiable objectives (see OKRs in Part I). Simply put, when we start implementing the goal, there should be a way to check if we are actually moving in the defined direction and how far did we

go. Any sensible set of metrics and milestones can help with this understanding. For example, "we have to grow in revenue by 60% every year" might make a practical goal. However, to use a financial metric, you need to have a proper compensation system in place (otherwise, revenue is perceived as another abstract parameter).

- **Goal should highlight your bottlenecks**. There is a common misconception that Agile Transformation is about adopting some combination of frameworks and methodologies. In reality, we want to rebuild and optimize the organization, solve its internal problems, and make it more adaptive and flexible. Eliyahu Goldratt's Theory of Constraints (TOC) shows that all optimization efforts must focus on a system's bottlenecks. Otherwise, we risk spending our time and energy on local optimums that do not "move the needle." Within the TOC approach, the right goal should first spot the most critical constraint we have at the moment. Next, when this one is resolved, the same goal should point at the following constraint, and so on. To get familiar with TOC, see Goldratt's book with a signature name — "The Goal."
- **Goal has to be well communicated** throughout the organization along with transformation plans and details. Only in this way can it be shared by the whole organization and become recruiting and alignment tool.

Unfortunately, there is no universal template for the transformation goal. Every organization would need to create a unique one basing on the organization's ambitions, culture, and context. However, there is a simple way to test a prospective goal. Try to "sell" it to several opinion leaders in different parts of the organization. If you see enough excitement, you are on the right track.

Nurture enthusiasm

Imagine that there is an intangible distributed energy supply that powers any change process. Let's call this energy *enthusiasm*. Enthusiasm is generated and amplified by change leaders, change agents, and later on by other engaged individuals. Enthusiasm is spent on breaking all kinds of barriers and on building new structures and processes. Similar to a computer game, the amount of energy you have is always limited, although renewable. If you use up the energy too soon — the game is over (transformation has failed).

This imaginary model might look arguable and not entirely serious. Nevertheless, it explains the cases when all big decisions were made, considerable resources were invested, but transformation efforts had faded away shortly. To avoid such an unfortunate scenario, we need to manage the supply of enthusiasm carefully. There are ways for the change leaders to add energy to the change process:

- Gradually engage more people from all parts and levels of the organization
- Ensure consistent support from the organization leadership

- Celebrate all wins, even small ones
- Establish good communication between all parts of the organization so that people share ideas, successes, and failures
- Accept the fact that some parts of the change plan do not work for this organization and help the teams and individuals to find alternative solutions

There are also ways how limited supply of enthusiasm is wasted:

- By running change without a clear goal and understanding what we try to achieve
- In contrast, by investing too much effort in preliminary theoretical preparations
- By diluting the change process with endless talks and discussions
- By following transformation plan dogmatically, enforcing changes "by the book" (more on this right next)

Do not get stuck on books and theories

One wise observation of Prussian field marshal Helmuth von Moltke is usually simplified to a formula "No plan survives first contact with the enemy." Expanding this formula to the Agile Transformation frameworks and methodologies (see the list in Part I), we would have to conclude that none of them can be universally applied "out of the box". This does not make frameworks are fake or useless, of course. Most of them were based on best practices and actual cases collected in real life. However, as was noted earlier, every organization preparing for the transformation has its own unique prerequisites:

- set of challenges
- point in the lifecycle
- environment and business context
- culture and values

You simply cannot squeeze such a complex combination of factors into any predefined template.

Most likely, to make your transformation successful, you would need to iterate, modify plans, change course, innovate on your own, customize or even switch frameworks. At least, that is what all unedited case studies reveal. See the examples in Part IV.

Unfortunately, among the supporters of every framework or methodology, we might meet dogmatic consultants and coaches. Their guidance is usually quite strict and straightforward — "do it by the book." What can be the consequences if we follow such guidance?

- Leaders and change agents might dump the most valuable native Agile practices that were implemented internally prior "official start" of the change process (since those practices do not precisely match what the selected methodology demands).
- Teams involved in transformation might not be given enough space to experiment with new practices, customize them and share the results (since everything should be strictly "by the book").
- Teams and individuals pushing, managing, and participating in the change process might waste substantial energy arguing about insignificant details or enforcing practices in an exact manner.
- When anything goes wrong in the process (and something always goes wrong), consultants or coaches do not help our organization to find an alternative solution. Instead, they just give a universal explanation of the problem — change was not done "by the book."

At the same time, running the transformation entirely ad hoc might be equally dangerous. We risk running in circles, slipping to the old routines, wasting the whole energy trying to reinvent the wheel. So how do we find a healthy balance?

- We need to do our homework to understand our options and see different perspectives: reading the books and articles, going to the conferences and courses, talking to colleagues with positive and negative relevant experience. This way, we can make hypotheses about frameworks and methodologies that might be a decent match for our organization.
- When it is necessary, we should hire the right consultants and coaches. Transformation is quite a challenging process; adding some external forces might be very helpful. However, we have to be very selective and hire only specialists with proven relevant experience and expertise; that are ready to be flexible and experiment together with us.

Value your people and culture

In the book "Brave New Work," Aaron Dignan, who has quite extensive experience in various change processes, shares his understanding of organizations' nature:

"We need a new approach to organizational transformation. We can start by accepting that organizations are complex adaptive systems, not complicated mechanical ones. They are living systems, not machines. They are the sum total of the principles, practices, mindsets, assumptions, and behavior of hundreds or thousands of people."



Source: *How to Revolutionise The Way We Work With Aaron Dignan* video, Penguin Books UK Channel, YouTube (2019)

Aaron Dignan explains two root mindsets required to manage and develop any organization as a living system

From a "living system" perspective, it is clear that all significant factors shaping our organization depend on the people we work with. If so, we need to define the most valuable "assets" we initially have and preserve those while struggling through the transformation:

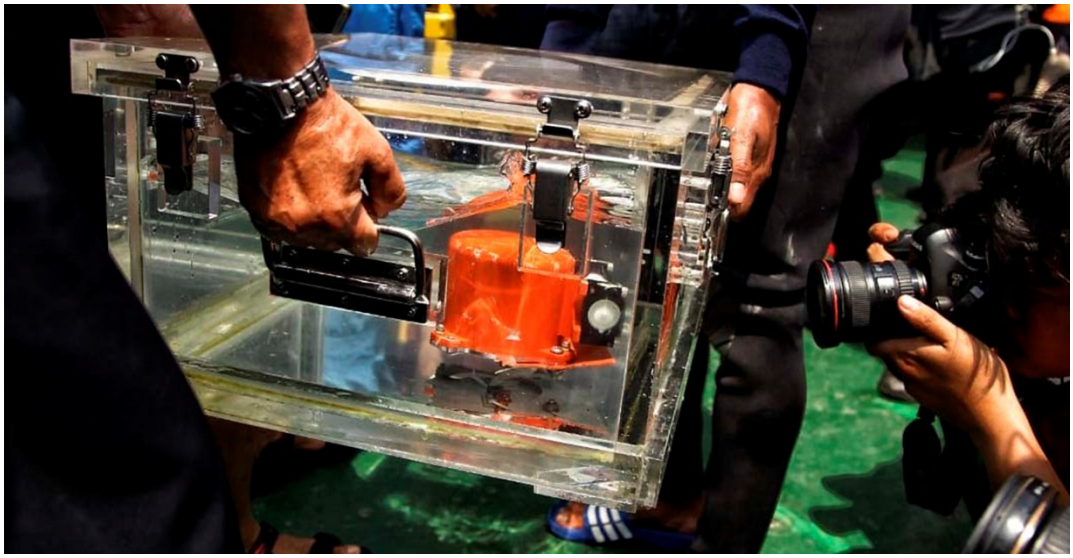
- **Key people and the sum of their expertise.** Every organization has a group of internal champions — managers, key specialists, and opinion leaders (often apart from the official org structure). Imagine that we lose a significant part of those champions in a short period of time. Suddenly, our organization might see a critical decline of competence and competitive power, leading to degradation and potential destruction. Since champions are often well-positioned prior to a transformation, many of them might see a threat in the new course and start looking for another job. You are on the safe side if potential Level 5 leaders (in Jim Collins's terms) thrive in your organization. Those leaders are defined by valuing the organization's success over their own careers. Alternatively, if you did not manage to grow or keep such people, the risks are more significant. In either case, see some ideas on dealing with constructive opposition earlier in this article. Plus, you definitely can find many other ways to motivate your core specialists. It is critical to understand which of your people you cannot afford to lose, involve them in the change process and constantly invest in keeping them positive.
- **Culture, core values, internal communications.** When professional change agents (consultants, coaches) talk about transformation at scale, you would often hear that it is mainly about cultural change. Agile Transformation is

no exception. At the same time, your existing corporate culture and values are what your organization is built on. Any healthy change process should keep the morale up and the culture gradually improving. Sadly, in many cases, transformation is thoughtlessly enforced by top management, causing people's demotivation, erosion of the culture, and disconnection between people and leadership. To avoid these devastating effects, change leaders must establish and continuously improve communication channels in all parts and layers of the organization. The feedback loop, an essential part of change communication, would allow leaders to correct course, resolve impediments, and understand the progress.

Unfortunately, the concepts of people's value, core engineering culture, and the critical role of two-way communication sometimes sound too abstract even for the leaders of best engineering companies. That is what likely happed at Boeing, leading to literally catastrophic results.

Boeing story: Fatal disconnection

Boeing, the iconic US airplane manufacturer, was going through a controversial and painful transformation for about two decades. On this journey, Boeing's leadership was able to achieve impressive financial results but at the expense of the company's innovative safety-centric culture and interconnection with the core engineering force. What happened next we all saw on the news.



Source: Pradita Utama/DETIKCOM/AFP via Getty Images, photo for the publication by Harry Suhartono, Alan Levin, and Julie Johnsson *Boeing Jet's Faulty Sensor Wasn't Fixed Before Lion Air Crash, Report Finds*, Bloomberg (2018)

The black box recovered from the first of two Boeing 737 Max crashes (Lion Air Flight 610 on October 29, 2018)

In October 2018 and in March 2019, less than five months apart, two new Boeing 737 MAX planes crashed, killing 346 people. The cause for both crashes appeared to be the same combination of malfunctioning sensors and a poorly designed software subsystem. After the first crash, Boeing did not fix the problem. Pilots only received instructions on how to turn the malfunctioning subsystem off. After the second crash, the reaction was more appropriate — all 737 MAX planes were grounded and stayed grounded for the next 20 months (until December 2020).

The investigations that followed revealed many additional technical issues and other embarrassing details. The history of this crisis was covered in the Bloomberg documentary "How Boeing Lost Its Way" and many other media reports.

Initially, Boeing was created and grew for about 80 years as an engineering organization. Jerry Useem, in his article for The Atlantic magazine, emphasizes, "Its executives held patents, designed wings, spoke the language of engineering and safety as a mother tongue." Big changes started in 1997, after the merger with McDonnell Douglas. Boeing received a group of executives who were measuring success exclusively by the stock price level. The introduced policy changes probably seemed an obvious choice for top managers with financial background — increasing profits by cutting costs and breaking the dependence on the engineering force.

New leadership was quite explicit about the direction of changes. Useem quotes Harry Stonecipher, who held several executive positions at Boeing from 1997 to 2005, saying, "When people say I changed the culture of Boeing, that was the intent, so that it's run like a business rather than a great engineering firm." This new policy was executed in several ways:

- **Radical capital expenditure cuts**, minimizing the investments in improvements of research, development, and production.
- **Constant layoffs and pension cuts**, causing strikes and dropping the morale in both R&D and manufacturing. Bloomberg Businessweek quotes Mark Rabin, experienced engineer who worked in Boeing for 17 years and was a part of 737 MAX flight test group "It was pretty intense low morale because of all the layoffs—constant, grinding layoffs, year after year; so you really watched your step and were careful about what you said."
- **Headquarters relocation.** In 2001 Boeing HQ was moved from Seattle, where it was close to the manufacturing facility, to Chicago. This divide added even more disruption to the communication between engineers and top management.
- **Outsourcing design to external firms.** As airplanes' design is Boeing's core competence, that approach had significant consequences. The outsourcing practice was first used for the Boeing 787 Dreamliner, which was delayed for about 3 years (released in 2011 instead of 2008). Next in the line, the Boeing 737 MAX was forcefully released according to the schedule but apparently with critical quality issues.

In these circumstances, the engineering community had developed a sort of siege mentality. It is well reflected in the words of a union leader on the 40-day strike in 2000: "We weren't fighting against Boeing; we were fighting to save Boeing." In the following years, two-way communication between top management and engineering organization was entirely replaced with constant pressure to reduce costs and deliver faster. As a result, Boeing's values of safety, problem-solving, and engineering innovation were dissolved; a "culture of concealment" took their place.

Congressman Peter DeFazio summarizes in his interview to Bloomberg:

"Boeing was a magnificent company, who made (in my opinion) the best airplanes in the world, until some of these recent cultural changes, which all came from the pressure of Wall Street and bad management."

MCAS, the software that was the root cause of Boeing 737 MAX's two fatal crashes, was also created for a controversial reason. This software emulates the behavior of previous-generation airplanes, so no additional expensive simulator training of a pilot is required (in this way, Boeing tried to match a competitor's offer). But the problem was not in the solution itself. The problem was that the whole development workflow was defective. As it was revealed during the investigations:

- software was not properly designed since it acted as a single point of failure
- side-effects noted during the simulator tests were not properly addressed
- specifics of the solution were not properly communicated to the FAA authorities inspecting the airplane
- all pilots who piloted new airplanes were not properly trained (in fact, they were not aware of MCAS until the first crash)

Rick Ludtke, former Boeing engineer and expert at flight crew alerting systems, concludes: "They could have done better and should have done better, but better wasn't an option."

During the investigation, Boeing made public a large number of internal emails and other text messages. Here are just a few examples:

- "This airplane is designed by clowns who in turn are supervised by monkeys" (about the issues with the plane's flight management computer)
- "I want to stress the importance of holding firm that there will not be any type of simulator training required to transition from NG to MAX" (from Boeing 737 chief technical pilot)
- "We put ourselves in an impossible position by picking the lowest cost supplier and signing up to impossible schedules; I don't know how to fix these things…it's systemic. It's culture. It's the fact we have a senior leadership team that understand very little about the business and yet are driving us to certain objectives" (in reference to the 737 MAX simulator supplier)

This evident crisis had significant implications for Boeing's business. The company's CEO was forced out. The production of 737 MAX was suspended. In 2019, even before the aviation industry collapse caused by travel restrictions of 2020-2021, Boeing reported its first annual loss since 1997.

The new CEO of Boeing, David Calhoun, seems to realize the source of the problems that brought the company to where it is. For example, as making the decision to temporarily stop the production of 737 MAX, the company did not lay off its experienced shop-floor workers. Another example — The New York Times quotes Calhoun as stating: "he would focus on insulating engineers from business pressures." Also, commenting on the new FAA certification to CNBC, Calhoun emphasized: "I have to listen to our employees; I simply have given them the time they need to the natural discourse that occurs in a big complex certification process."



Source: *Watch CNBC's full interview with Boeing CEO David Calhoun* video, CNBC Television Channel, YouTube (2020)

David Calhoun describes his reaction to Boeing employees' messages; a shot from Bloomberg's documentary

Such a clear position of the new Boeing's CEO gives us, the flying public, hope that the company can restore healthy communication between the experts (engineers, pilots, manufacturing professionals) and top management.

Another point that David Calhoun sees as critically important is restoring transparency within and outside the company. Here is what he says in the same CNBC interview:

"Of the lessons I've learned over last year that transparency is where Boeing fell short. And we will not fall short on that subject under my leadership. It will be uncomfortable, but we will be transparent on every subject, whether it is training, whether it is a certification process, everything along the way <...> You will know that I know".

It does not seem that "You will know that I know" was just a nice declaration. In April 2021, Boeing traced back an issue with the electrical power system that could potentially appear in some 737 MAX airplanes released before the grounding of March 2019. The company warned FAA and airlines plus made this information public. About a hundred affected planes (approximately a quarter of the global 737 MAX fleet) were grounded immediately for check-up and fixing.

If this emphasis on transparency is equally reflected in Boeing's internal communication, it can play a significant role in recovering trust within the company. Both transparency and trust are critical for restoring Boeing's engineering culture step by step.

The important note here is that Boeing's case is not the case of Agile Transformation. Boeing's changes were based on a perception that its engineers are a commodity. In contrast, modern transformation (if understood right) is about empowering people and unleashing their capabilities. However, as we already saw earlier in this article, any serious changes tend to create a strong opposition. If leadership does not realize the value of people or does not carefully manage goals, culture, and two-way communication, such opposition can divide the organization into "top management" and "people." If you let this happen, you will likely experience something very similar to the Boeing case.

Key takeaways from Part III

- Intel's case study explains the reasons for organization leaders to act paranoid. We also see that a long-living corporation would likely be forced to go through more than one dramatic pivot in its history. And that it can be hard to notice the business landscape changing, especially while you dominate this landscape.
- As transformation is prepared and rolled out, it usually meets significant resistance. It is important to differentiate the opposition types depending on underlying motivation:
    - desire to keep the status quo at any cost
    - allergy to a specific methodology
    - sincere fear for the organization's wellbeing

  The opponents of the last two types can be potentially converted into true supporters of the change.

- Statistically, at least 70% of large-scale transformation efforts fail. The article highlights several key actions necessary to make the change process viable:
    - Set a compelling goal. The goal must be inspiring for the people, so it cannot be negative or too abstract. The goal should direct changes, so it should be quantifiable and should spot the organization's most severe bottlenecks.
    - Nurture the enthusiasm of people and leaders to keep transformation energized.
    - Stay far enough from books and theories. Although some external wisdom can be essential, it can also act as a poisoned gift. The team might head in the wrong direction, discard promising initiatives, and exhaust people while trying to squeeze such a complex system as every organization is into a predefined template.
    - Save the most valuable assets that define the organization — key people, unique culture, core values, and internal connections.
- Boeing's case study demonstrates how a transformation can lead to literally catastrophic results for the organization, its customers, and the public. Recent investigations of two fatal Boeing 737 MAX crashes showed that the reason behind the crisis was a transformation the company went through for two decades. During this process, Boeing's leadership achieved impressive financial results but destroyed innovative safety-centric culture and the connection with the core engineering force.

Part IV: Adventures of One Animation

Real cases of how organizational models emerge, evolve, applied, and mixed together

As we already saw in earlier parts of this article, nowadays, any mature well-positioned business might feel a major threat of disruption. We can better understand the reaction to such a threat if we look at few cases specifically from the banking industry. Bill Gates is often quoted saying somewhere in 1994-95, "we need banking, not banks." It took some time for this idea to get accepted, but after recovering from the 2008 financial crisis, many banks started to realize the danger coming from FinTech Startups and Big Tech Giants. WSJ cites Jamie Dimon, CEO of JPMorgan Chase, speaking at an investor day in 2014: "When I go to Silicon Valley…they all want to eat our lunch. Every single one of them is going to try."

However, our tale starts quite far from the banks and banking.

Evolution of Spotify

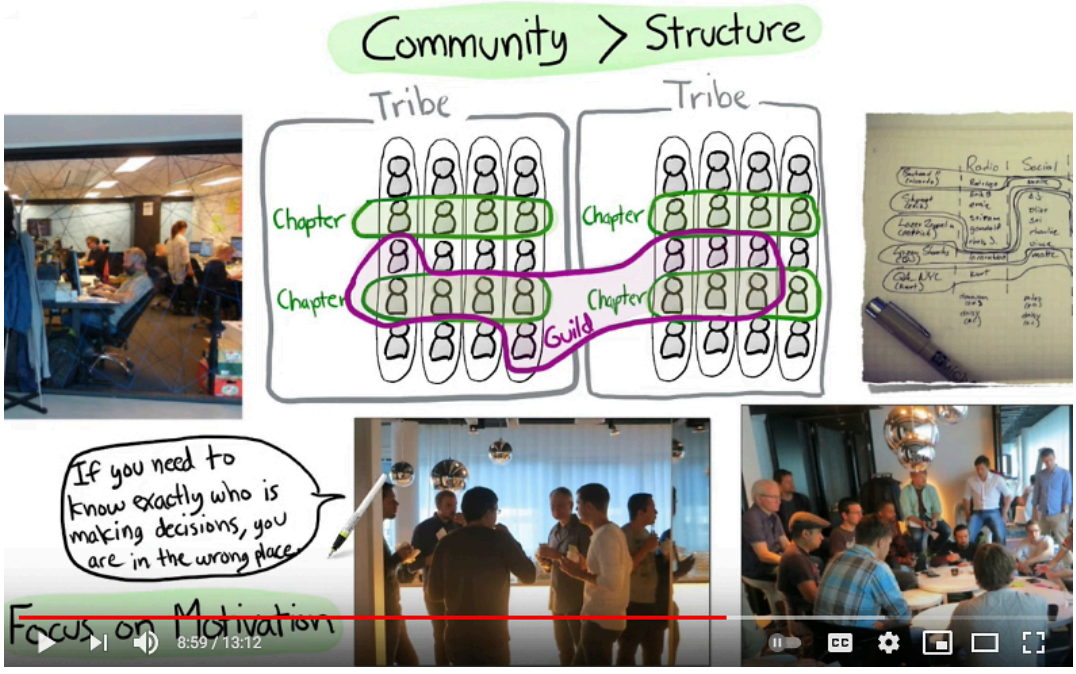
This story began in 2006. Spotify, now the biggest music streaming service, successfully competing with Apple, Amazon and Google, back then was just a little Swedish startup dreaming big. By 2008 the company was able to launch its platform. Every year since, the platform itself, user base, and team size kept growing rapidly.

Spotify has been practicing different Agile techniques on the team level from the very beginning (including Scrum and Kanban). However, as the number of development teams grew at an increasing pace, the company needed to build

an efficient organizational structure that could coordinate hundreds of people, avoiding chaos but preserving agility.

By 2012 Spotify already had about 30+ development teams (up to 8 people each). At that time, a group of external Agile coaches was helping the organization go through the growing pains. Henrik Kniberg, who we already know from Part I and Part II of this article, was among those coaches. At the end of 2012, together with Anders Ivarsson, Henrik wrote a whitepaper "ScalingAgile@Spotify" where they described org structures and techniques Spotify had implemented to that moment.

Later, in 2014, Henrik decided to update this whitepaper — to show how the approaches had evolved and add many exciting details. Interestingly, instead of writing another document, he created (with his own hands) two animated videos — "Spotify Engineering Culture: part 1 and part 2". Videos have their "official" homepages (part 1 and part 2) at atspotify.com. Plus, those who prefer reading rather than watching can find the full transcript of part 1 and part 2 hidden somewhere in Google docs.



Source: *Spotify Engineering Culture - Part 1* video, Henrik Kniberg' Channel, YouTube (2019)

Screenshot of org structure illustration from "Spotify Engineering Culture" video

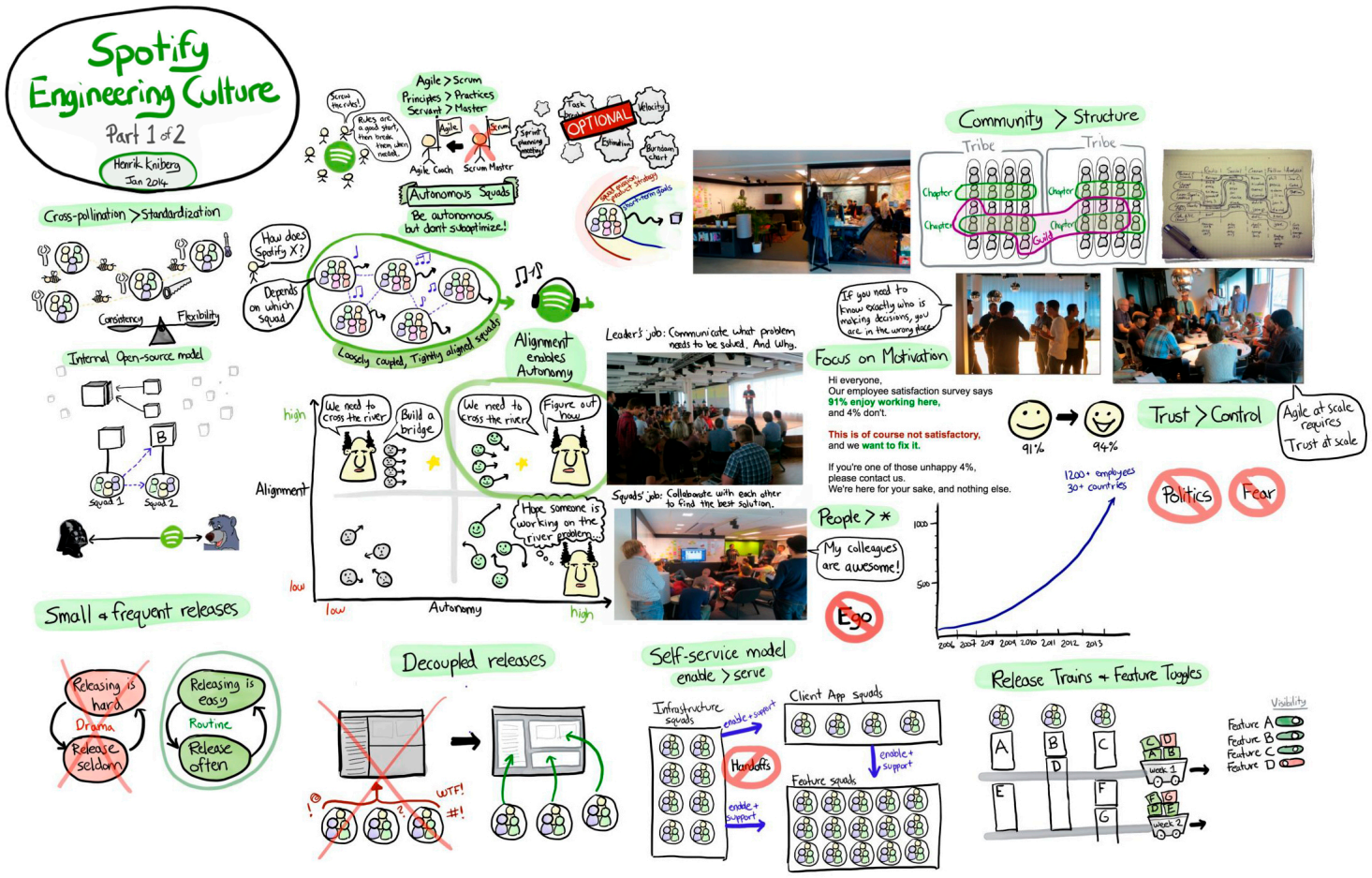
Henrik's videos are short, fun, well explained, and provide more information on the company's structure and process than you can find in an average book on management. Below are some highlights from the "ScalingAgile@Spotify"

whitepaper, “Engineering Culture” videos, and several keynotes of Henrik and other Agile couches that worked at Spotify at the time.

Spotify, as a natural-born Agile company, initially widely used Scrum on the team level. However, as the company grew, some Scrum practices became a limitation, so they were made optional. Teams decide themselves which frameworks to use and in what part (some teams used Scrum, some Kanban, some mixed approaches). “Rules are a good start, but then break them when needed.”

As the company doubled in size every year, it had to create some unconventional structures and processes. To avoid disputes and confusion on “the right implementation” of a specific Agile methodology, Spotify decided to skip most of the existing terminology and create its own set of terms for the roles, processes, and organizational structure elements. The organizational structure elements included:

- **Squad** is a small (up to 8 people), co-located, cross-functional, self-organizing team. There are three kinds of teams — infrastructure, feature, and app squads (more on balance between functional and cross-functional teams see in Part II). Each Squad is an autonomous team — it has end-to-end responsibility for what its builds, from design to deployment. Also, autonomous here means that “Squad decides what to build, how to build it, and how to work together while doing it.” A Squad is synchronized with the company goals through Squad’s mission and alignment framework similar to OKR.
- **Tribe** is a collection of Squads (less than 100 people in total) grouped around certain customer-related or backend/platform functionality (Playback Tribe, for example). Tribe is structured as a “lightweight matrix.” The main difference from the “classic” matrix structure here is that the primary dimension is Squad, which is focused on product delivery.
- **Chapter** is the second dimension in the Tribe’s matrix. Chapter is a virtual functional group (for example, Quality Assistance Chapter or Web Development Chapter). Chapter lead is a line manager of the people from different Squads having the same specialization. Chapter lead is responsible for developing his people, onboarding new hires, setting salaries, etc.
- **Guild** is a companywide informal community of interest, joining specialists of a specific competency and facilitating knowledge sharing and skills improvement (for example, Continuous Delivery Guild).



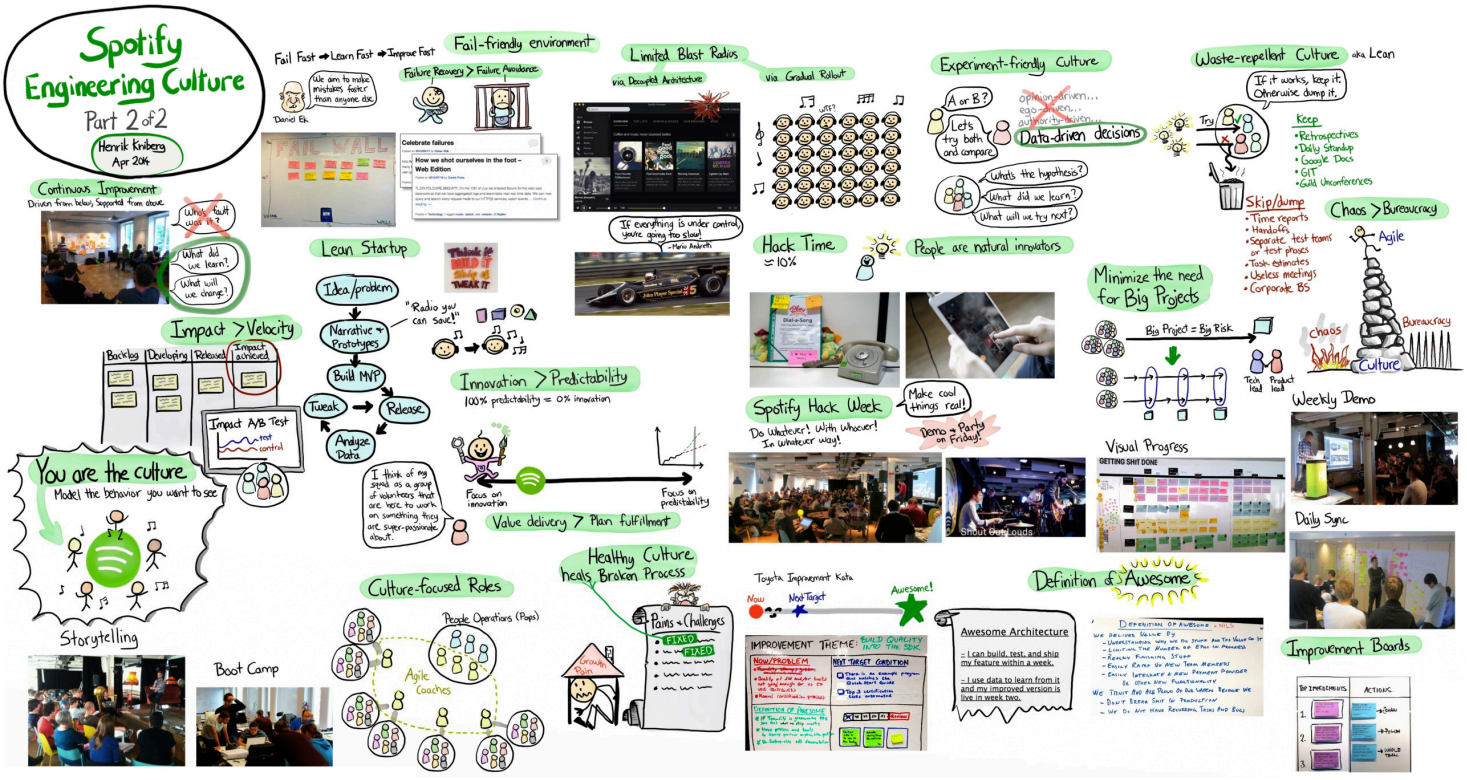
Source: Henrik Kniberg, illustration published at Henrik Kniberg' blog post, *Spotify Engineering Culture (part 1)*, Crisp AB (2014)

The canvas of the "Spotify Engineering Culture" video (part 1)

However, the main message of the videos was about particular mindsets and practices that formed Spotify's internal culture:

- "Alignment enables autonomy"
- "Trust is more important than control"
- "If you always need to know exactly who is making decisions, you are in the wrong place"
- "Fear doesn't just kill trust, it kills innovation"
- "We care more about innovation than predictability"
- "We aim to make mistakes faster than anyone else"
- "Healthy culture heals broken process"

These points might sound like a collection of slogans for those who never worked in a similar environment. Nevertheless, it is safe to say that most of them are critical for the described organizational system to function and evolve.



Source: Henrik Kniberg, illustration published at Henrik Kniberg' blog post, *Spotify Engineering Culture (part 2)*, Crisp AB (2014)

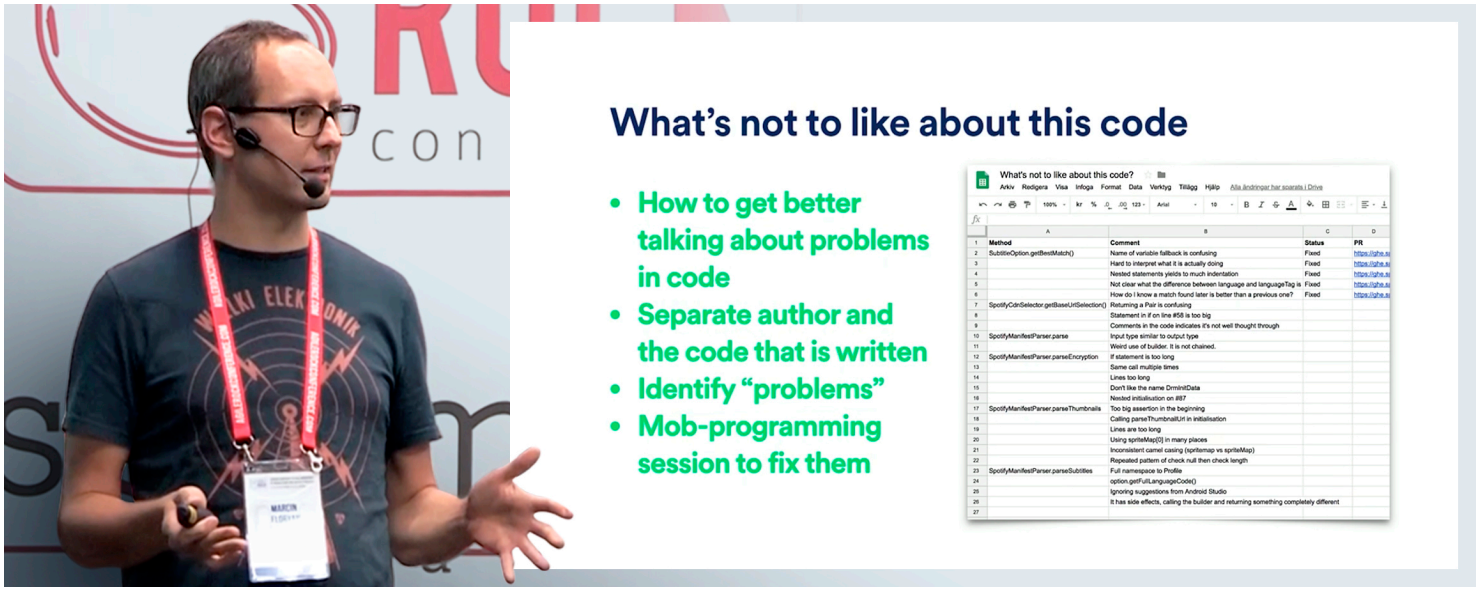
The canvas of the "Spotify Engineering Culture" video (part 2)

Worth noting that all details summarized above are just a snapshot of how Spotify was organized at a specific period of its history — in 2012-2014. Moreover, it is not a 100% accurate snapshot — in Henrik's own words, "this culture description is really a mix of what we are today, and what we are trying to become in the future." But as information spread, the description of Spotify's way of working became extremely popular and eventually formed a legend of the Spotify Model. Many organizations started to use that virtual model as a target for their transformation, and at least some of them succeeded. In "KPMG Global Agile Survey 2019", Spotify Model was graded as the third most popular framework for Agile at Scale.

But let's get back to Spotify's story. The overwhelming popularity of the concept was entirely unexpected for those who worked at Spotify. In mid-2015, Henrik described that clearly in his post "No, I didn't invent the Spotify model." Moreover, other coaches and team members tried to explain to the public that the company still has many challenges, that there is no "Agile Nirvana" at Spotify, and that Spotify Model is not a model per se, so organizations should not thoughtlessly copy it. The latest is definitely a good recommendation, although it equally applies to any other model, framework, or methodology. As a result, most of the comments you find on the Spotify Model today would warn you to be cautious with it (see a recent post on accenture.com, for example).

Meanwhile, Spotify itself is naturally immune to any imperfections of the model carrying its name. As mentioned above, this model was never considered 100% real or final to any degree. As the company grew, it was regularly facing new challenges and finding new ways to solve them. For example, Kevin Goldsmith, at the time Spotify's VP of Engineering, during his talk at the 2015 InfoQ conference, described new organizational elements that Spotify added within a year or two. Those included POCLAC (includes Product Owner, Chapter Lead, Agile Coach) and Trio (joins leaders of engineering, design, and product for a particular Tribe). Other interesting details on the evolution of Spotify's Engineering Culture we can learn from the talk that Marcin Floryan, Tribe Lead of Playback Tribe, gave at the 2018 Agile Rock conference. Marcin described several new practices his team uses, including Tribe Standup, quarterly Big Room Planning, Taps with Squads (meetings of Tribe leaders and a Squad), and several others.

In this way, Spotify successfully evolves and expands, attracting new customers and employees, opening new regions, and winning over Big Tech competitors.



Source: *Marcin Floryan | From Principles to Practices | Agile Rock Conference* video, Scrum Ukraine Channel, YouTube (2018)

Marcin Floryan explains how his team is managing to solve challenges related to psychological safety

Revolution at ING

One of the early and well-known cases of Agile Transformation in the banking industry is the case of ING Bank. The bank's change process is described in many publications and research papers, including the case study by William Kerr and his

colleagues from Harvard Business School and the research article by Martin Calnan and Alon Rozen from Ecole des Ponts Business School.

The ING Group is a multinational bank headquartered in Amsterdam, eleventh in the list of largest European banks. Throughout all its history, ING tried to stay on the edge of the industry's innovation. However, while recovering from the financial crisis of 2008, the bank started to recognize that it is not moving at the right speed. Nick Jue, at the time CEO of ING Netherlands (ING NL), uses a colorful metaphor in his interview published on bcg.com:

"This picture of an elephant being chased by greyhounds is exactly how people look at banks. People view banks as big, inflexible animals, and they are chased every day by "greyhounds" such as fintechs and other new companies. What I try to do is train our organization, the elephant, to be as fast and as flexible as a greyhound. I want to remain an elephant, because I want to keep the power of the elephant. But I also want to be fast and flexible."



Source: *Nick Jue on Transforming ING Netherlands and Introducing an Agile Way of Working* video, Boston Consulting Group Channel, YouTube (2020); illustration for the article by Martin Calnan and Alon Rozen *ING's Agile Transformation—Teaching an Elephant to Race*, Journal of Creating Value, SAGE (2019)

Nick Jue explains the underlying reasons for the ING Bank's transformation

Nevertheless, as it often happens nowadays, the change process initially started bottom-up — from the bank's IT department. In 2010, when Ron van Kemenade became CIO of ING NL, he discovered a wide selection of challenges caused by legacy structures, systems, perceptions, and processes. One of those challenges was ING's mobile app, ranked the lowest in category: "it had one star, and the only reason it had one star is the fact you cannot rate an app with zero stars."

Ron remembers that his first natural reaction was to initiate an emergency project to develop a brand-new mobile app. However, quite soon, he had two engineering managers involved in the project coming to his office to explain that a decent mobile app will never be created with the current way of doing things. Fortunately, they had a vision of a better way — Agile software development. By the end of this conversation, Ron was totally convinced: "You've asked for one team for mobile app — I will actually give you two; you've asked for two months to prove — I will give you three. And I will put you in the middle of the building, and I will protect you every way possible."

New mobile app development was the ultimate success: three months to a working demo; nine to go live; two years to level functionally get a higher rating than the competitors. These and following wins were widely celebrated, and Agile practice had gradually expanded to the rest of the IT organization (there were 175 Agile teams by mid-2013). But that was just the beginning.



Source: *The Agile Transformation at ING* video, IT Revolution Channel, YouTube (2016)

Ron van Kemenade tells the story of changes in the IT organization of ING

As the IT organization of ING NL was going through its transformation, engineers started to communicate with business trying to correlate what they do with customer needs. On the business side, the perception of IT was gradually improving too. However, most of the time, the cooperation between business and IT was limited to assigning Product Owners to Agile teams.

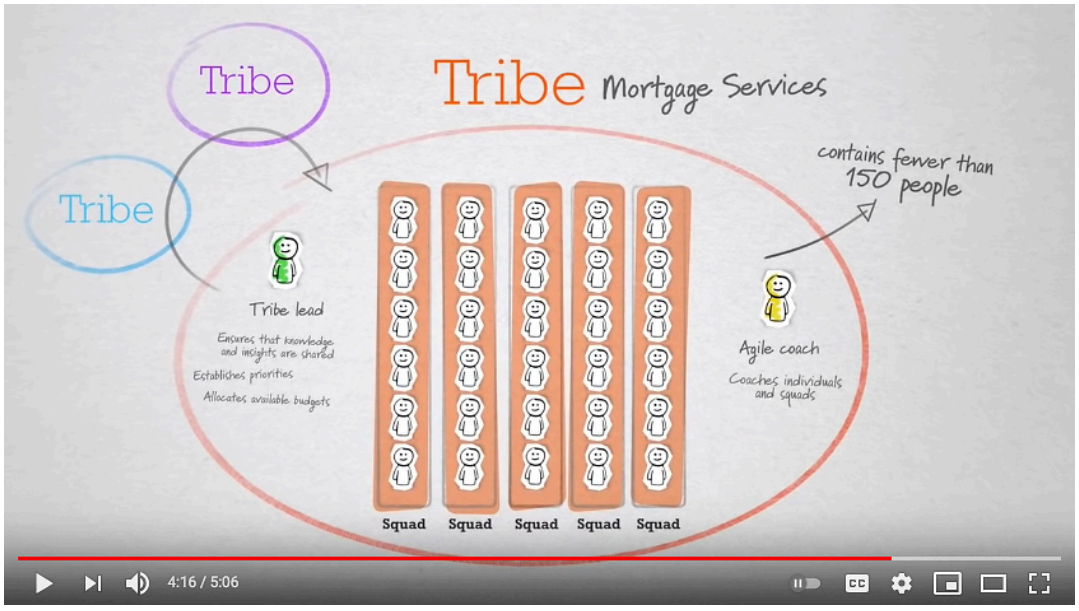
In parallel, the bank's leaders were getting increasingly concerned with the "we are not moving fast enough" situation. The major shift happened in 2014. William Kerr's case study quotes Marije Lely, a COO for business lending,

explaining: “It all started with ING NL’s management team, who visited a number of companies, such as Spotify, Netflix, Google, and Zappos. Our business is financial services, but we specifically looked at non-financial companies to be truly inspired, sort of ‘un-banking the bank.’”

By the end of 2014, ING leaders decided to instantly rebuild a big part of the Dutch organization basing on the “Spotify Model.” Notably, the bank used the assistance of Boston Consulting Group (BCG) while preparing the transformation, and later on, BCG included the “Spotify Model” into its guidance for Agile transformation.

Bart Schlatmann, COO of ING NL at the time, in his interview to McKinsey Quarterly, describes how leadership was preparing this radical transformation: “We spent two months and five board off-sites developing the target organization with its new ‘nervous system.’ In parallel, we set up five or six pilot squads and used the lessons to adapt the setup, working environment, and overall design”.

While designing the new organizational structure and working on other preparations, leaders also tried to communicate changes clearly. Vincent van den Boogert, ING NL Chief Customer Officer at the time, described those efforts in his interview for William Kerr’s case study: “It was first about the Why we are doing this, and What the direction was that we want to take, the dot on the horizon — and many colleagues understood that. That was more or less the good and inspiring news.” Interestingly, at the initial stage of transformation, ING had borrowed some of the educational materials from Spotify (for example, “aligned autonomy” we saw in part 1 of Henrik’s video). But eventually, the bank received its own creative animated video.



Source: *Agile way of working at ING Belgium* video, ING Belgie Channel, YouTube (2017)

Screenshot of org structure illustration from “Agile way of working…” video by ING

Worth noting that since the business of ING is quite different from that of Spotify, many details of model implementation varied significantly. For example, in the bank's case, many Squads would need to have team members with a wide variety of specializations. If we look at the Squad from the "Agile way of working" video, it has specialists in marketing, product management, formula management, data analyst, user experience, and IT (engineers). Since the Squad's size is supposedly 9-10 people or less, there is room for 3-5 engineers only. It is hard to say if the bank managed to decompose all its offering into truly independent products, small enough for such teams to handle (this contradiction we already discussed in Part II).

But the most remarkable aspect of the 2014-2015 ING NL change process was the "shock therapy" method used to push it through. The target was set for the organization of 3,500 people (marketing, product and channel management, IT) to be fully operational in the new format by June 2015. Vincent van den Boogert: "The leadership layer directly below the ING NL management team was 'fired' and had to apply again. <...> It's tough, but we didn't hire back some people who were seen as heroes at the company. That was the first shock; people realized that this was serious. The first hired tribe leads selected, in turn, the chapter leads, who repeated the process to select the customer journey experts." Similarly, the rest of the transforming part of the organization had to re-apply for their jobs within about two months.

Thanks to the significant effort of change leaders, HR, and the employees themselves, the plan became a reality. On June 15, 2015, all newly selected employees gathered in the biggest Amsterdam stadium and jumped up at the same moment for the photo that would signify the inflection point in Bank's history. The next day, everybody started their new jobs in the completely restructured Agile-style office (nice interior design, glass walls, open spaces, whiteboards, and colorful couches).

However, there were significant casualties too. The renewed organization had lost a quarter of its workforce. Marije Lely recalled, "There were some people who were well-known for their expertise that were not selected because they didn't fit with this new way of working. <...> We had to say goodbye to 25% of our colleagues, so that was a radical and heavy change." Vincent van den Boogert shared similar thoughts, "I don't know if it is a regret, but we may have over-weighted the focus on culture in the hiring. It was really painful to see accomplished colleagues leave." The focus on culture Van den Boogert mentioned refers to the Orange Code — a manifesto describing the ING's core values and behaviors (officially introduced at the beginning of 2015). "Orange Code Assessment" was a part of the re-hiring process.

Later on, in 2016, the transformation was expanded to Sales and Services — 7,000 more people. However, for this part of the ING NL organization, another model (based on elements of Zappos organizational structure) was used. This is a good illustration of the pragmatic approach to the frameworks — ING neither tried

to directly copy-paste a model nor to create something from scratch for the sake of novelty. Nick Jue summarized this approach in the interview he gave to TechQuartier, listing the elements bank has taken from different sources:

- From Spotify — concepts of business and IT together in one team, end-to-end team's responsibility (from idea to customer) plus structural elements like Tribes and Squads
- From Netflix — quarterly business reviews (QBRs)
- From Google — hiring methods (which Bart Schlatmann described as "peer-to-peer hiring approach")

Nick Jue concludes, "We took all the best from companies out there, and we do the Agile in ING way, not as a copy of a certain company."

Moreover, the ING's own model kept evolving as the bank was getting more experience and change process spread from HQ (Netherlands) to other regional offices. For example, from late 2017, ING NL and ING Belgium worked on the "Cross-border Agile" initiative to unify banking platforms for both countries. While both regional offices were executing virtually the same model, the cross-regional Tribes experienced significant difficulties with product delivery and Squads' alignment. We learn these details from the article and keynote of Cesario Ramo, who was invited to help one of the Tribe leaders. Cesario is an experienced consultant and change agent specializing in LeSS implementations. Therefore, his approach was to improve ING's Agile model by embedding the elements of the LeSS framework.

How successful was the transformation for ING overall? First of all, it was definitely not fatal, despite the 25% loss of experienced people at the "shock therapy" stage. Moreover, the bank's change leaders are very positive about the outputs, which included shortened time-to-market for the products and increased employees' satisfaction. At the same time, the business metrics' dynamics we can see at TheBanks.eu analysis does not reflect any radical differences appearing from 2013. ING stays competitive and keeps growing at a stable pace, similar to other major banks in the region. Although not stunning, those results can, in fact, be considered promising, as changes might pay off more noticeably in the long run.

Transformation of Sberbank

Sberbank (now Sber), Russia's largest bank by virtually all measures (for example, at some point, it had more than 300,000 employees in total), for many years was perceived as a giant post-Soviet monster. Bank had started its gradual change process in 2007 after Herman Gref became its CEO. Over the years, Gref's team worked on converting unfriendly rigid institution into the modern customer-centric bank. Despite all the criticism Sberbank still receives, it is fair to say that organization was practically reinvented within 5-7 years.

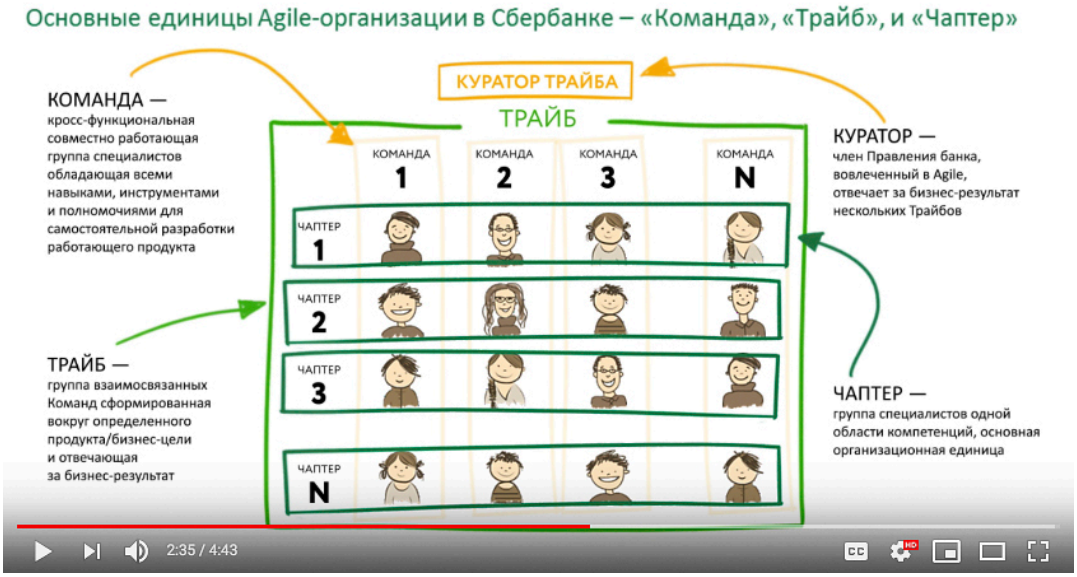
The modernization process was supported by massive investments in IT development and IT infrastructure. In 2011 bank created a specialized software development company — SberTech. One of the main assignments of the new SW vendor was building a unified platform for the bank's operations. This project was completed by the end of 2015, and at that time, SberTech already employed 6,500 engineers and managers.

Meanwhile, beginning from 2013-2014, Herman Gref started to share a new concern with his colleagues and the public. He argued that the new real competitors of the banks are not other banks and not even fin-tech startups. Instead, real competitors are new technological companies (Google, Amazon, Alibaba, Facebook, Yandex) that will inevitably expand their powers to the banking services. In January 2016, during an emotional speech at a panel discussion of Gaidar Economic Forum, Gref outlined the situation his bank facing:

"We were very proud of what we have accomplished within recent years... We invested a tremendous amount of money... Sberbank now is the largest IT company in the country... We just received some prize for the biggest project of centralized IT infrastructure... But once the project was finished, we realized that we are absolutely not competitive. That was a terrible shock for me personally... We developed an excellent system for a bank, but those we compete with are not the banks <...> This year, we will make 41,000 changes to our platform... Amazon makes 10,000 changes to its platform in a day. Our time-to-market is months; their time-to-market is hours... So we decided to make a breakthrough <...> We already have Agile in our IT, but this will not help if we do not have Agile in the whole organization... We have to rebuild all our processes."

Here we need to take a step back to understand why the CEO of Russia's biggest state-owned bank considered western Big Tech Companies as actual competition and had chosen Agile as a strategy for the new phase of the bank's transformation. In the interview published in HBR-Russia shortly after the speech quoted above, Gref mentions that the bank's top management travels every year to Stanford for special one-week corporate training. Within several years, as a part of these and other trips, Sberbank's change team visited (or listened to speakers from) many leading tech companies and transforming organizations, including Apple, Google, Netflix, Uber, HP, Cisco, Alcoa, JPMorgan Chase, Citigroup, and ING Bank. Also worth mentioning that Sber used McKinsey's assistance to design and execute the change process. And, at some point, McKinsey added the elements of the Spotify Model into its guidance for Agile transformation.

The resulting big shift had officially started in September of 2016 with the video "Sbergile: Change story" published on the bank's website. The video opens with Herman Gref saying, "Agile is the most radical transformation in our history." And, without much surprise, we see a new variant of Agile organizational structure animation in the middle of the video.



Source: *Sbergile: Change story* video, SberTV Channel, YouTube (2016)

Screenshot of org structure illustration from "Sbergile: Change story" video

Notably, Sber combined the Agile techniques used in Spotify and ING with other Agile methodologies and named the resulting framework "Sbergile." In this way, the bank can keep improving its own Agile model without running into limitations of common Agile at Scale frameworks (see the list in Part I). Another interesting fact is that at the begging of 2017, Sberbank had hired Bart Schlatmann from ING as Chief Transformation Officer (although, Schlatmann stayed in Sber for about one year only).

The transition was massive but did not happen at once. In 2018, at a conference marking two years of transformation, Yulia Chupina, Vice President responsible for Team and Culture Development, described the bank's journey to that moment. As we see from Yulia's slides, the Agile part of the organization grew from 580 people (6 Tribes, 86 Teams) at the first stage (started September 2016) to more than 11 thousand people (21 Tribe, ~1400 Teams) at the third stage (started March 2018).



Source: *Sbergile*, Dmitry Bakhtin's Channel, YouTube (2018)

Yulia Chupina describes the stages of Agile transformation in Sberbank

Similar to ING, Sberbank moved the restructured part of the organization to the new Agile-style office (nice interior design, glass walls, open spaces, whiteboards, and colorful couches). However, another more important aspect of the change process was notably different: Sber has likely managed to keep the majority of its people during the transition. Natalia Danilenko, Head of HR in SberTech, emphasized during her talk at another conference that none of the mid-managers quit, although the role and authority were dramatically changed for most of them.

But what was the big goal of Sberbank's transformation? This becomes clear if we look at Sberbank's "Strategy 2020" presented in December 2017 at Investor Day in London. At the begging of his talk to investors, Herman Gref revealed the key formula for the Sber's future: "The banks are crying because the Silicon Valley eats their breakfast every day... We want to enjoy the breakfast of those who are eating other banks' breakfast." To implement this formula bank decided to build an extensive integrated ecosystem. This ecosystem would provide customers with not only a full range of financial products but also with a large variety of other products and services — from health, education, and entertainment to transportation, legal services, and industrial equipment. As a base for the new ecosystem, Sber planned to use its IT platform and a huge collection of customers' data (in 2017 bank already had 85 million active customers). "I do not believe in banks," Gref concluded.



Source: *Sberbank Investor Day (eng)* video, SberTV Channel, YouTube (2017)

Herman Gref explains why Sberbank has to build its own ecosystem so it can compete with Big Tech Companies

How successful was the transformation for Sberbank overall? Although the management team's plans did not look entirely realistic in 2017, it seems that the bank has rather succeeded. At the end of September 2020, the Agile transformation was "officially completed," reaching the mark of 30,000 people in 3,000 product teams. A few days earlier, Sberbank (renamed to Sber) launched its ecosystem at a spectacular online event. The new ecosystem includes Sber's own gadgets carrying its own personal assistant, own movie streaming service, own app store, own educational platform, own cloud computing platform, partners' services from taxi to food delivery, and much more. One month later, at the next Investor Day online conference, Herman Gref and his management team presented their new plans — "Strategy 2023".

Notably, Sberbank's Agile model keeps actively evolving despite the declared "completion" of the transformation. For example, in December 2020 bank organized a professional conference — Sbergile Talks. At her keynote, Xenia Yashina, Head of Agile transformation office, described the stages of the Sbergile evolution. According to Ksenia, Sbergile today merges a wide variety of Agile frameworks and practices expanded with Sber's own organizational innovations: Spotify Model, ING Model, Scrum, Kanban, and lately LeSS (to deal with large products).

As was already mentioned, Sberbank today is still widely criticized in Russia for monopolistic practices (although the same we would hear about Big Tech Companies too), internal bureaucracy, and problems with the customer experience. Nevertheless, the ability of a giant state-owned institution to transform itself and reach ambitious goals is truly impressive.

Forming your change

The amazing adventures of Henrik's animation show us the story of an advanced organizational model that was created incidentally — from a brief description of how things work (or should work) in a specific company. Several reasons might have made the Spotify Model so popular:

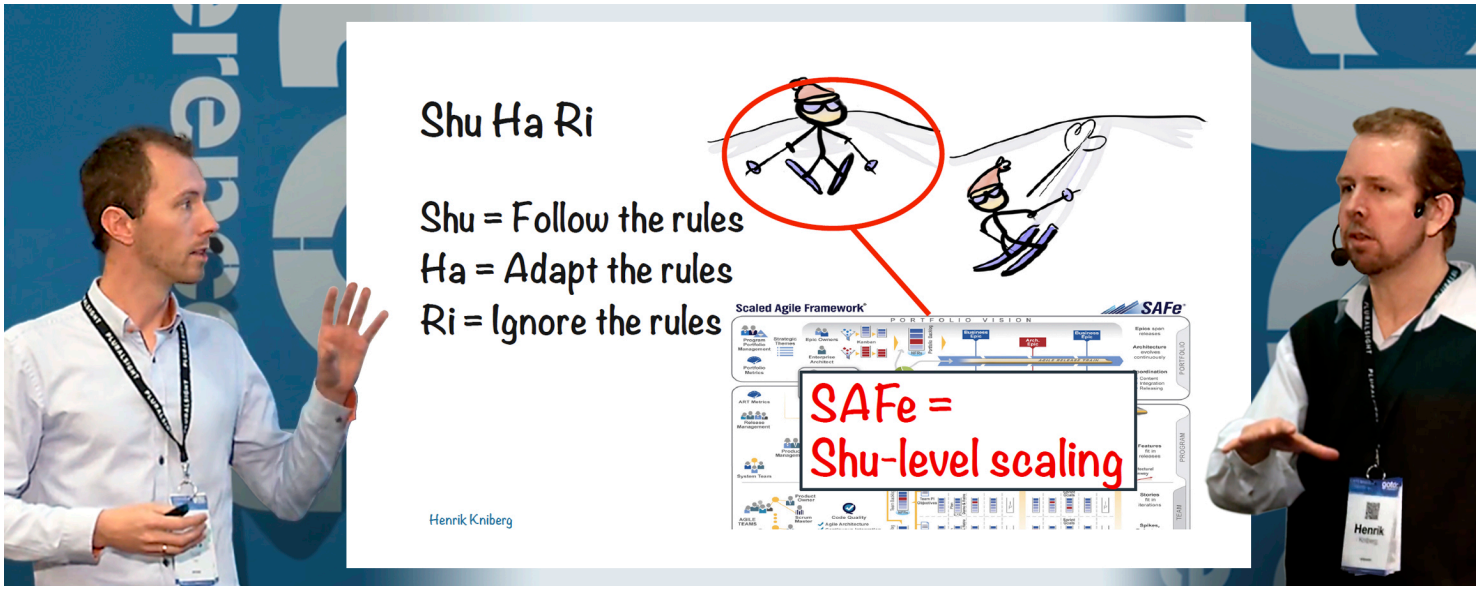
- The model was described in a brief, coherent and straightforward way.
- The model gave a clear idea of the target organization structure. Other scaling frameworks are usually more focused on processes and define only specific structural elements necessary for a framework to function.
- There was no institution behind the model mentoring how exactly things should be, so every organization was free to customize the model in a way it seemed more appropriate.

But what about other common Agile at Scale frameworks listed in Part I? Can we use them in a similar manner?

How to tailor a model

As was argued earlier in this article, any methodology can become dangerous if applied thoughtlessly or dogmatically. However, if we treat an original framework as a starting point, select valuable elements only, and add new ones when necessary — we can utilize the best of what a framework can provide.

One good example of this approach is the implementation of the extra-complex SAFe framework at LEGO Group. Lego also recruited Henrik Kniberg as a change agent and Agile coach. At GOTO 2015 conference, Henrik, together with Lars Roost, at that time Program Manager at LEGO, gave a talk with the dramatic title "Is SAFe Evil?" Trying to provide a fair answer to that bold question, they explained: "Tools are neither good nor evil. It all depends on what you do with them… We look at SAFe as a toolbox. There are a bunch of tools in it. You can use tools to hit yourself in the foot or for valuable stuff." Next, Henrik and Lars described the specific components of SAFe that the change team implemented. Plus, they explained the trial and error process the team went through, customizing those components and inventing new ones. For example, a big part of the story was about the evolution of the Big Room Planning Event (in SAFe it is called "PI Planning"). "The question is it SAFe or not? Are we following the rules or not?" Henrik asked himself, closing that part of the talk, "The answer is — who cares."

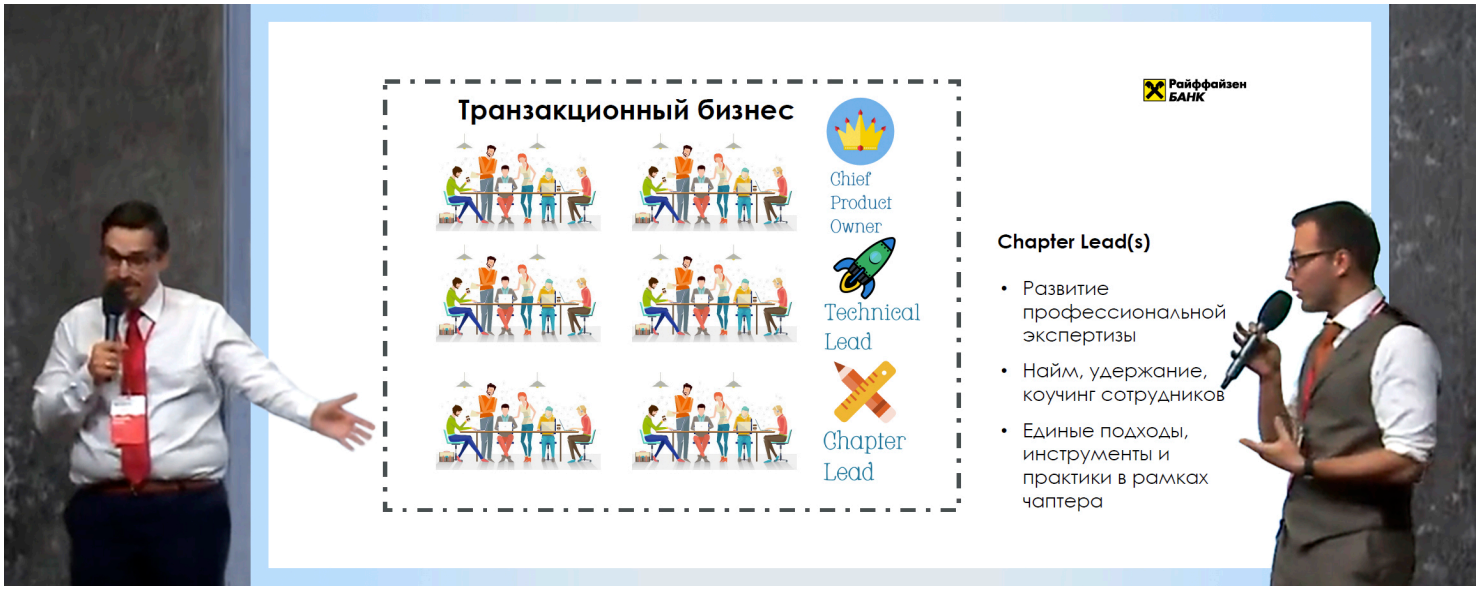


Source: *Is SAFe Evil? • Lars Roost & Henrik Kniberg • GOTO 2015* video, GOTO Conferences Channel, YouTube (2015)

Henrik Kniberg and Lars Roost (on the left) explain when SAFe can be helpful: Henrik uses the ShuHaRi concept and the example of the snowplow technique in skiing

Another example of how the frameworks are treated pragmatically we find at one more bank. This time we would look at the change process in Raiffeisen Bank, one of the biggest banks in Austria with a number of subsidiaries in Central and Eastern Europe, including Raiffeisen Bank Russia. From the article in HBR Russia, we learn that similar to the other cases we saw, the bank's Agile transformation had gradually started in 2012-2015 from the IT part of the organization. At that time, change leaders experimented with various Agile techniques and different ways to apply them. Although not all of those attempts brought considerable success, the bank was able to develop a critical skill — to learn from the failures and keep only the solutions that worked well.

In 2016, the top management of Raiffeisen Russia got involved in the Agile transformation, including Sergei Monin, Chairman of the Management board. As a result, the change process had successfully expanded to the business part of the organization. At the end of 2017, giving a talk at a conference, Sergey Shcherbinin and Nikolay Knysh, at that time the key change agents at Raiffeisen, described the structure of the product-oriented part of the organization (IT+Business) as a customized combination of Spotify and ING models.



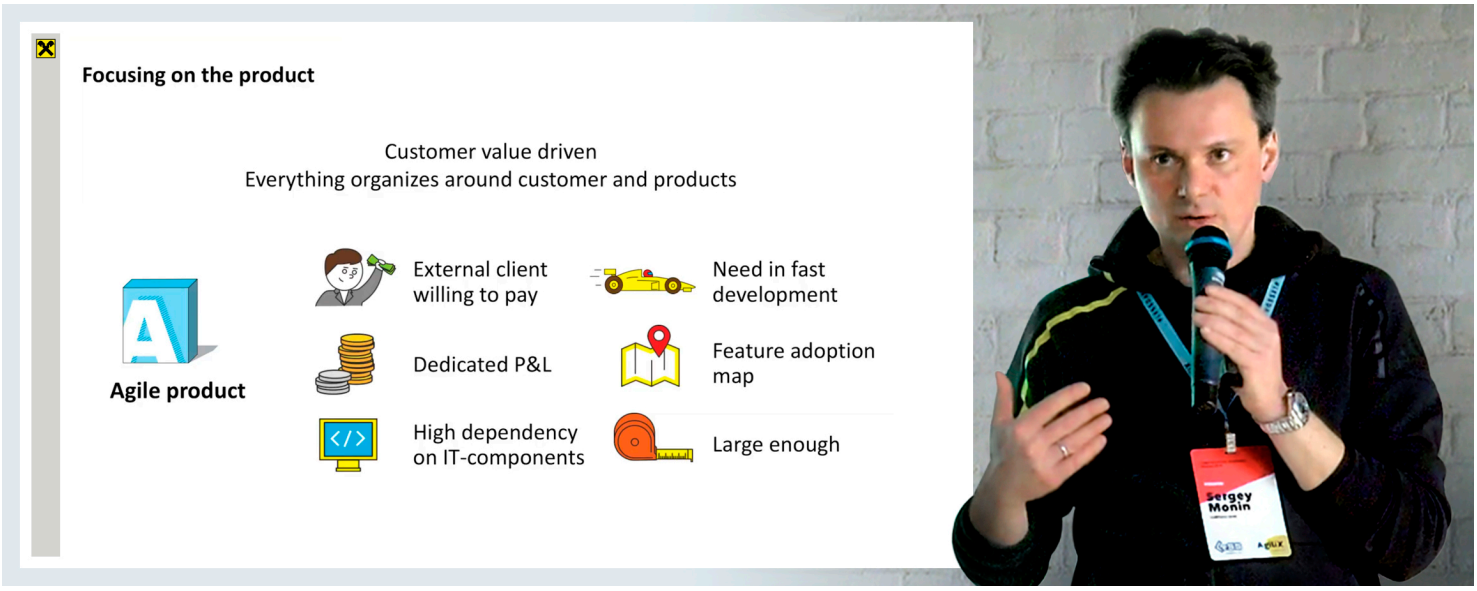
Source: *Sergey Shcherbinin, Nikolay Knysh. Vanilla Scrum vs Big Enterprise* video, ScrumTrek Channel, YouTube (2017)

Sergey Shcherbinin and Nikolay Knysh (on the left) describe Raiffeisen's experimental organizational structure (partially based on Spotify Model) using the example of one of several divisions

Remarkably, one of the central questions Raiffeisen's leadership tried to answer during the transformation was "What is our product?" As we saw earlier in Part II, the limited ability of a small team to handle large products is one of the main drawbacks of the autonomous product team concept. Simply put, you might not be able to make all your products small enough, so each of them would fit the single team capabilities. And that was precisely the challenge bank faced.

In late 2019, Sergei Monin with Hannes Eickmann, Agile Transformation Lead, during their keynote for a conference, explained that preparing for the big shift in 2016-2017, the bank's leadership considered several scaling frameworks. For example, they rejected SAFe because "SAFe is a very rigid framework, lots roles and rules, no empowerment of the teams." Note that similar criticism of SAFe we often hear from many change professionals, including Marty Cagan, who we met in Part I.

Finally, with some involvement of Cesario Ramo (see ING case above), leaders had settled on the LeSS framework. The core practices of LeSS allow several feature teams to work virtually independently on the same product. Having this solution, the bank had structured its product portfolio so that each product would have an external client, separate P&L, and the Product Owner with enough authority. Still, by the end of 2019 bank had only 7 product teams using the LeSS practices. Each LeSS product team was a reasonable size — 2-3 feature teams. And all other products were handled by 16 "regular" small autonomous product teams (more on various types of teams see in Part I).



Source: *"LeSS for Senior Leadership" the case of Raiffeisen bank* video, LeSS Day Europe Channel, YouTube (2019)

Sergei Monin explains how Raiffeisen bank has defined its products

To summarize frameworks' customization cases, let's look at the recommendations from KPMG Global Agile Survey 2019. As was already noted in Part I, despite the limited regional coverage and the lack of the latest data, this research provides an informative overview of the Agile at Scale frameworks and helpful insights about their implementation in reality. Here is what the KPMG team writes on the "Select a framework and make it your own" topic:

"When evaluating which framework to use, it is crucial to ensure it fits your organization. Before making your selection, it is important to carefully evaluate the framework for its synergies with your organization's ambition, culture and maturity. Do you require hands-on support and guidance? SAFe would be handy in these cases as it is one of the most elaborate frameworks you can use. Conversely, if your organization is more attuned to learning by doing, the Spotify model would typically be a good way to model. It is really about what you need as not all frameworks cover all parts by itself. Ultimately, the focus should be on which framework suits your organization's ambition, culture and maturity rather than which framework is 'the best'".

Where to start

Suppose we are convinced that Agile transformation is crucial for our organization, and we want to move on before it is too late. Where do we start?

The first what we would notice is a massive selection of consultants specializing in Agile techniques. Although initially distracting, this can be a positive factor if we manage to select true specialists with relevant hands-on experience.

Another positive factor is that organizational transformation existed long before Agile, so it is an established discipline. Since all generations of change processes have shared challenges, learning how to solve them can be quite helpful. The basics here are the works of John Kotter, from his classic "Leading Change" and "The Heart of Change" to the later updates to his methodology in "Accelerate." However, as with any solid theory, there is a risk of taking Kotter's recommendations too seriously, getting stuck in preparations, and losing momentum.

To get a better sense of what are the Agile organizations and how can your organization be converted into one of them, consider one more video and the books that were already referenced earlier in this article:

- In his keynote "Alignment at Scale" for the Agile Africa 2016 conference Henrik Kniberg summarized the main ingredients of Agile culture and techniques for an enterprise.
- In "Brave New Work," Aaron Dignan dedicated about one-third of the book to the specifics of taking an organization through the modern transformation.
- "No Rules Rules" by Reed Hastings and Erin Meyer is written entirely as a step-by-step action plan of converting your organization into a new Netflix.
- In "Working Backwards," Colin Bryar and Bill Carr tried to summarize Amazon principles and practices in the way those can be applied in other organizations.
- In "Empowered", Marty Cagan and Chris Jones described in detail the org structure of a real company utilizing the autonomous product team concept (see more in Part II) and added two chapters about the transformation process.

Key takeaways from Part IV

- The article tells the story of the so-called Spotify Model that emerged as the unforeseen consequence of one white paper and two animated videos, capturing the organizational structure and processes of Spotify in 2012-2014. The model features:
    - unconventional matrix structure based on concepts of Squad, Tribe, Chapter, and Guild
    - signature engineering culture built on specific mindsets and principles, including: "trust is more important than control," "we aim to make mistakes faster than anyone else," "healthy culture heals broken process"
    - freedom of interpretation, since virtually no one believes this model was ever complete or final

- We looked at several cases from the banking industry that were initially based on the Spotify Model in combination with other frameworks and techniques. Banking served as is a good example since it was heavily affected by the threat of disruption. In each bank, Agile transformation had initially started in IT. Next, it scaled to the business part of the organization with the support of top management. Still, there were significant differences in change process strategies:
    - ING Bank's case is one of the most famous in the industry. Bank used Spotify Model as a base structure, adding many custom elements. The change process in the bank's HQ was forced in radical "shock therapy" manner — 25% of the people involved were lost at the launch stage. Nevertheless, ING maintained its competitive position and continued growing, transforming other parts of the organization and adding other Agile methods to its techniques portfolio.
    - Sberbank had started its big shift with the ambitious idea to convert the bank into an IT company with its own platform and ecosystem. The goal was to compete with new tech companies (including Big Tech) on their grounds. Bank used the Spotify Model and ING Model to form a custom framework (Sbergile) and merged in other frameworks and techniques later. The product-related part of the organization was transformed in stages and with no "human resource" losses. Bank keeps moving towards its goal — both platform and ecosystem were successfully launched and now expanding.
    - Raiffeisen Bank demonstrated a pragmatic approach to the utilization of transformational frameworks and techniques. The bank went through series of "trial and error" while implementing Agile on a team level in IT. When the change process spread to the whole organization, a structure similar to Spotify and ING models was used, but with significant adjustments. Later, facing the usual "our products are too big to be handled by one team" dilemma, Raiffeisen added LeSS techniques to support products it could not grind small enough.
- As an additional case study (outside of the banking industry), we looked at SAFe implementation at Lego. Although SAFe is viewed as an overcomplicated framework killing the agility and empowerment of the product teams, change leaders managed to bypass dogmas and treat the methodology creatively, thus reaching positive results.

Closing thoughts

To conclude this article, let's go through a few core ideas which were already mentioned earlier:

- It looks like the only way to succeed with organizational transformation is to build your own unique model. A rational way to do this is by learning, customizing, and experimenting with the available tools — frameworks and techniques which were partially borrowed from the big disrupters. Another valuable source of information is the practical experience of the organizations that went through phases of similar transformations (the best info is that you gather informally, not relying on attractive success stories).
- As you prepare and roll out the change process, make sure you have a good plan on retaining and developing the most valuable assets your organization has — culture, core values, key people, and the sum of people's competencies.
- When you realize that your organization is a complex adaptive living system, you would also see that the change process must be constant. Although there should be milestones with clear and (preferably) quantifiable results, do not set a "project finish date" if you want your organization to thrive.
- Converting your organization into Lean and Agile organization should be more feasible if you proceed in a Lean and Agile manner: whenever it is possible — select a reasonably small part of the organization (find or build a team of early adopters); apply transformation in small iterative steps; establish the immediate feedback loop; measure the results; correct the course at every step; make a radical pivot when necessary; keep the momentum and expand the change process involving other parts of the organization.
- If your change process has started from bottom to top, involving your top management can be a key factor for further success. At any level, the role of leaders is to inspire people, facilitate and nurture the change process and, primarily, deal with impediments. In that sense, top managers can remove the roadblocks that are out of reach otherwise. Note that once highly positioned leaders start to participate, they would likely make substantial changes in the current plans and approaches. That is the price you pay to go from small wins to scaled transformation.
- The two central processes that the change leaders must heavily invest in are communication and education. A change process can succeed only if a continuously growing number of people share new ideas and values. Initially, leaders need to "sell" the reasoning behind the transformation and, above all, the transformation's goal and plans.

Further on, they need to build an environment and processes allowing teams that already work in a new way to share their achievements, successes, and failures with other teams and the rest of the organization. Later, the same communication channels should become a natural part of an Agile organization, as they are necessary for the alignment and motivation of people. Education is, in a way, a part of the communication process too. The organization learns most effectively when teams share the best practices they have discovered. Another important role of education is to align people on the specifics of Agile processes and structures. For example, a centralized high-quality training program would equip all teams with a common vocabulary and a unified understanding of new practices.

- Some leaders might not fully realize that, but one of the essential features of an Agile organization is transparency. The underlying logic is simple. When converting your organization to an Agile way of working, you try to achieve better utilization of the talents and abilities of your people. That is why you go through all the hassle of establishing autonomous product teams and eliminating middle management. From that point, you empower your people on the ground to use their best judgment and make all important decisions within the team's responsibility area. However, to make the right decisions, people must have all the necessary supporting information available. For example, simplifying a bit, every contributing individual in a product team needs to know the P&L of the product she or he is working on and the role that product's P&L has in the company's P&L. You cannot empower people without providing them with essential strategic and tactical contexts.

Thank you for reaching the end of this article! I hope you could find something useful for your particular situation. I will be happy to hear your comments and ideas. Please contact me at my LinkedIn page: linkedin.com/in/peter-pirogov.

Good luck with your transformation journey!