

# User-Centering Scaled Agile Product Creation

Florian Grote, Native Instruments GmbH, Berlin

Research paper presented at WorkingProducts Conference

Hamburg, June 23, 2017

The paradigm of the agile organization has come a long way since Toyota pioneered Kanban in the 1970s (Ohno 1988) and teamwork concepts emerged in the 1980s (Wergin 2003). Interestingly, these models of production surfaced in large-scale enterprises, namely car manufacturers, and were focused on hardware production (cars). Yet, when talking about their successors today, we mostly focus on software engineering, and usually begin with the idealized startup way of doing things. This means we start with one team that decides and takes care of everything that is related to getting the product out to its users and then iterating on it with the feedback it gets from the outside. However, the majority of products in the world are not built by startups, and also not by single teams within larger organizations. As a result, Scrum and agile methodologies have been evolved to (re-)adapt to large-scale product development, where multiple teams and stakeholders have to be aligned. This has led to the concept of the Scaled Agile Framework (SAFe, Leffingwell 2016), the Lean Enterprise (Humble et al. 2015), and adapted large-scale Scrum paradigms (e.g. Gloger 2017), to name a few. This paper investigates how such scaled approaches could be setup as user-centered product creation organizations, and which implications for the decision processes around its products can be observed.

I.

One of the hardest things to understand when going user-centered with a team is the difference between "a" user and "the" user. When developing B2C products, developers, designers, and product owners are often users themselves. This is very good and desirable, as it allows them to personally connect to the vision and roadmap outlook. However, for discussions around features or new products, a development team that is very close to the product also has its challenges. Some teams are able to take on an abstract perspective and not let their personal judgement get in the way of team discussions, while others have a hard time restraining their members from offering their personal view whenever features or products are discussed in the team. This makes discussions unnecessarily long and can, in the worst case, lead to features or products that fit the needs of team members, but not of users in the market.

When going user-centered with such teams, there will inevitably be some disappointments in the beginning. People will have to understand that, even though they might be heavy users of the product they are working on, their needs do not represent the needs of the majority of users, or of the users the company wants to address. In an internal study, we conducted user research with both employees and externals, and compared the findings from the two groups. Maybe unsurprisingly, it turned out that the employees were much more technical in their statements, and that their tolerance for a less-polished UI was higher than that of the externals.

Nevertheless, having a high number of users on the development teams is a real asset when going user-centered. It means that the teams already know the fundamental language their users speak, and they can organically relate to problems and needs they hear about in user research.

II.

The ideal situation of user-centered design and development work would be one independent team, free from technical or strategic dependencies, that can focus entirely on creating value for their users. Startups might come close to this situation, as might internal startups or business units within bigger organizations that only need one team to deliver on their release stream. But as soon as the team's work is part of a larger organizational structure, this structure will want to be involved. As soon as the team has dependencies to other teams, whether hierarchical in a strategic relation or technical on a peer basis, these other teams effectively become stakeholders and the team that conducted the user research is not free to independently take and implement decisions based on what it thinks creates the best user/business value. Instead, stakeholders for whom this team is a dependency will naturally want it to deliver the results that are best for them, not for that team's users. This can even be the case in scaled structures, where a team defines strategic goals towards which another team, or set of teams, then is asked to deliver. The delivering team becomes a dependency for the team looking after the strategic goals across a number of teams, and soon enough, there will be conflicts between what the development team and the strategy team believe to best serve the users' needs.

One solution is, of course, to make the coupling between team results as loose as possible, even at the expense of having a strategy that might appear less sound or aggressive, or a user experience that might be less consistent than what could hypothetically be possible by keeping team outputs coupled tightly.

### III.

The collaboration between development, design, and product management, or products, is in a constant mode of renegotiation. Following a hierarchical structure, products typically come up with product concepts based on market openings they see, and then task designers with the definition of the exact feature set required to satisfy the needs of the users who would be willing to buy the product in this defined segment. Designers would then conduct research with these users, report back to product managers, and define requirements and specifications for developers to work from and make the product. In a user-centered agile development environment, teams are organized in a heterarchical structure (McCulloch 1945), exchanging information that leads to decisions which are taken either directly in the product development teams, or in special integration teams that oversee the interoperability of goals for several teams, in cases where more than one team is needed to build the product.

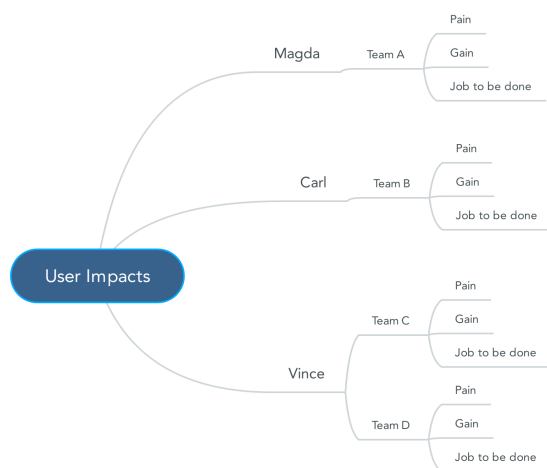
The traditional advice for building products that involve multiple teams in an agile environment is to decouple them as much as possible (Leffingwell 2016). Dependencies between teams are seen as counterproductive, as they limit the independence of individual teams in the pursuit of delivering the highest user value. While dependencies that limit team productivity are certainly to be avoided as much as possible, sometimes products can require multiple teams to work in very similar areas, where features depend on each other. Instead of opting to not build such products, such interdependencies can be dealt with in a cross-team setup.

In Scrum, the typical suggestion would be to hold a Scrum of Scrums (SoS) with all involved teams. This is often disliked by teams, who see it as just another process-driven meeting in their calendar. In my experience, teams come together almost automatically when they have a common need, or problem to solve. This happens for technical issues, and usually lasts until the dependency is resolved, if that can ever be achieved. Otherwise, it just continues whenever one of the affected sides sees a need for collaboration.

In addition, when serving specific user needs requires work from multiple teams, it makes sense to synchronize the respective product owners, potentially even to the point of them forming a product owner team for the purpose of serving this specific need. This can be a team around a product, if the release train is shared among the teams and their product owners, or it can be a team around a portfolio of products, or even just around a feature or set of features if the product is bigger than the release train of the affected teams. A structure of

programs that bring together product owners and other relevant stakeholders, such as marketing and support managers, can help to synchronize on releasing the highest user value across coherent portfolios of products. Nevertheless, there will also usually be ad-hoc teams that form around specific needs that require collaboration, such as ensuring compatibility with a new major version of the operating systems that are supported by the products. Following the American sociologist and network theorist Harrison C. White (1992 / 2008), such context-based collaboration between teams could be described as instances of loose coupling.

This highlights the need to differentiate the user needs towards which multi-team setups deliver. Any product that requires the work of multiple teams will likely deliver value towards a set of user needs, not just one single overarching need. It is of course still very helpful to have one main goal that unites the efforts of all teams involved in the work, but on the level of individual teams, it is better to have individual goals driven by separate user needs. The hard work of product management is to first define the overall, uniting goal, then identify separate needs that are facets of the overall goal, define team goals out of these facets, and lastly ensure that teams working towards these facets still align so that the overall goal can be reached.



*User impact map representing sub-goals for individual teams.*

To have a release stream that is not following the priorities of its own current end users might seem like a strange idea. Yet, for a variety of reasons, this is what we encounter in numerous streams every day, and not just in consumer products. One of the most common trade-offs a products team has to make is between development opportunities and user value. Development opportunities can include everything from quick wins for new features that

might not be the highest on the list of user priorities, to consciously increasing technical debt in order to realize user value in time.

#### IV.

The overall goal for a coherent portfolio of products can be defined as part of a narrative for the whole brand. Such a narrative spells out—in broad terms—how the brand might develop over the next five years, from a user's view. As a brand narrative, it is used to provide context for the work of the teams and align their efforts. Even though it represents a collection of overarching goals for the brand, it is itself only an explication of a part of the even wider narrative that covers the company's ecosystem of products.

In the brand narrative, input from the teams is integrated with strategic goals for the entire company and the brand, user research on the brand level, and findings from quantitative market research. From a birds-eye-view, this is where it all comes together and has to work as a consistent story that keeps users loyal and engaged while also attracting new customers. To that end, detailed strategic design research is undertaken that works with current and prospective users to define what might come in the form of products and features years down the road.

For development teams, the brand narrative is supposed to serve as a source of information and context into which they are asked to embed the solutions they come up with to serve the needs of their users. This task would be incredibly hard if the users a team is asked to serve did not align with the brand narrative and the overall company strategy. For this reason, the definition of which users to serve is a high-priority task for product management.

#### V.

In a situation where multiple teams work on the same product and towards the same strategic goals, managing user feedback across the organization becomes an important task. It should be communicated clearly who will be addressed by the items which teams currently work on and will tackle in the foreseeable future, in order to align everybody and secure the spaces for teams to maneuver freely. In the end, it is about having an overarching frame that brings together the individual teams and the company's strategy, and that is dynamic enough so that both can inform each other and have an influence of the future development of this frame. The format of the narrative has worked well for this task.

At the same time, teams need the ability to choose sub-frames that fit into the overarching frame, and they need to rely on having the freedom to find out for themselves how to best

create user value in their sub-frame. Product management can serve as the nexus between those frames, with its main job being the communication of findings in both frames and the prioritization of increments for existing frames or defining the outline of entirely new frames. Keeping the discussions that are inevitably a part of this work focused on the user is no easy task. Personas can help, by providing an abstract, simplified view on the variety of users' needs that are typically served by a product (Goodman 2012). It has to be clear that personas represent users, but they are not "the" user, either. While team members might be on the one end of the spectrum, having a very, sometimes too detailed perspective on user needs, personas are typically at the other end, where differences that matter to product development are sometimes abstracted away in an effort to make discussions across very different products groups easier. Once personas are established and understood by everyone in product management, they can become a good basis for prioritization and discussions around MVPs. However, it is crucial to limit the selection of personas to one primary and one or at maximum two secondary personas per product. Otherwise, the multitude of personas becomes unmanageable when discussions are needed across many different products in the portfolio.

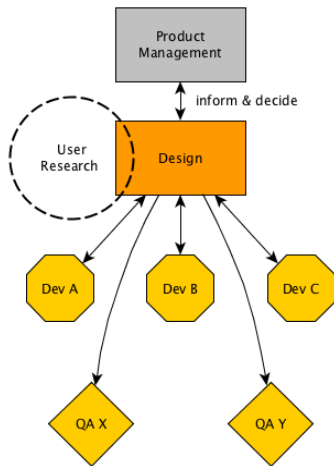
## VI.

In many ways, organizations can be seen as structures that create decisions. Enough decisions bundled together can take on the form of products. In any case, though, products are not possible without the more or less complex decision processes taking place within organizations. As sociological systems theory tells us (Luhmann 1995 / 2000, Baecker 2003), organizations are the only type of system that can structurally couple to other social systems, thus making it an essential part of society. This basic ability is what enables organizations to engage with suppliers, investors, marketing agencies, regulatory bodies, and yes, their users. All of these connections to the outside have to be dealt with internally, if the organization wants to use them to make informed decisions that lead to successful products. Karl E. Weick is calling this internal representation an "enactment" of what the organization observes on the outside (Weick 1979). This means that the organization has to find ways to efficiently deal with the results of these connections to the various activities that happen on the outside. At the same time, only members of the organization can take the decisions that later can lead to a product. Yes, design or feature questions can be posed to users, but this decision power was then lent to them by members of the organization, and in the end these members will be accountable for the success of the product. When things go wrong, all decisions will ultimately be attributed to members of the organization, and it will not be possible or recommended to hold the users accountable. With regard to products, the decisions taken

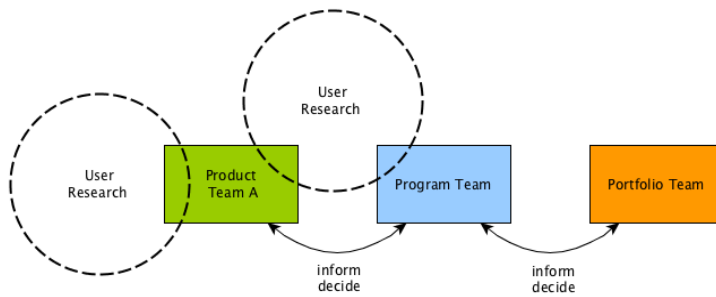
to produce them reflect the routine structure of the organization (as opposed to the hierarchical structure one might find in an organization chart). These products can be described as an objectified expression made by this structure, a statement the organization makes to the outside world, in the hope of being economically successful.

When moving from a waterfall organization to becoming user-centered in all teams, contexts of decisions begin to change. Whereas before, decisions were typically driven top-down in a hierarchical way, and taken at least one level above (in hierarchical terms) of the persons or teams actually implementing them, this now changes to decisions in the context of perceived user need, taken directly inside the team that also has to implement the outcome of the decision. Again, this is describing an ideal case. In reality, for products that have multiple teams working on them, decisions will need to be coordinated and dependencies therefore effectively become contexts for decisions.

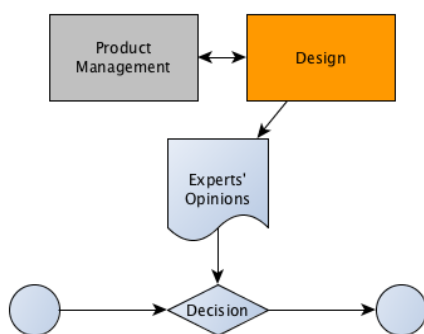
In the end, though, decisions in a user-centered organization are supposed to be attributable to specific user needs that have been observed in research. This gives the process of conducting user research and then interpreting the observations significant importance in product creation. After all, the prevailing interpretation of what the users actually need will guide all relevant product-related decisions in the organization. That is why, when going user-centered, organizations run the risk that the old guard of experts tries to position itself as the main scholars of the user and the interpreters of their statements. Likewise, the selection of interviewees can be challenged from all sides, questioning whether the "right" users were interviewed. These obstacles highlight the importance to clearly define which user(s) teams are targeting, and working with a supporting research team to set up the interviews and observation sessions can help to establish trust in both the process and the selection of research participants among the stakeholders.



*Design-driven decision process, where product management and developers are separated. Developers are not involved in user research.*

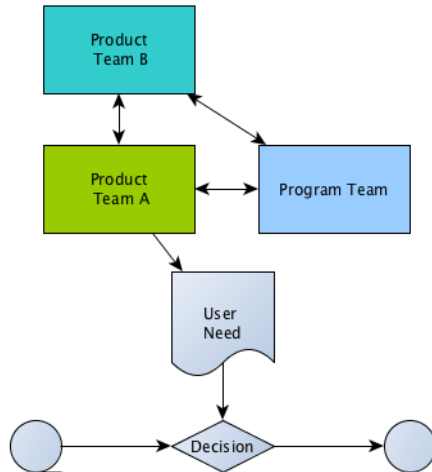


*Scaled approach to user-centered product creation. User research happens both in the product teams (developers and designers) and with both program (product management) and product teams involved. Portfolio acts as high-level product management institution, only taking decisions on a high abstraction level.*



*Traditional decision process with experts' opinions as context, informed by design and product management teams.*





*User-centered decision process with user need as sole context, itself however being informed by communication inside the product team and the scaled framework, i.e. other product teams with dependencies and a program team bringing together requirements from the brand teams.*

## VII.

The observation that products are an expression of the organizational structure that creates them still holds true for user-centered approaches. Whenever a full product vertical is not delivered by just one single team, the structure will imprint itself into the outcome of the development efforts. This manifests itself in the following, not exhaustive list of events:

- Teams negotiate who serves which user(s)
- Teams discuss the overall allocation of users to be served
- Product managers work hard to define frames in which teams can focus on users independently
- Technical discussions take place across teams that take user orientation into account
- Product managers communicate findings on user needs and solutions that address them in strategy discussions
- Product managers and teams discuss the integration of findings on user needs into an overarching narrative

With all this taken into account, it becomes clear that in order to build a user-centered team structure, managing the internal understanding of who the users are is just as important as conducting the user research in the first place. It is product managers who need to do the hard work here, as they are in the unique position to convey the findings from overarching frames and sub-frames of user orientation from strategy to development teams and vice versa. To be able to do this, product managers have to moderate the discussions on all levels, and keep up to date with research methodology as well. With these new responsibilities, the

work as a product manager gets more challenging because of the increased communication requirements, but at the same time also much more exciting and rewarding by changing the focus from inward-facing market analyses on abstract data to connecting with real users and finding out how to best serve them by creating value in their lives.

## References & Further Reading

- Baecker, D. (2003). *Organisation und Management: Aufsätze*. Frankfurt a.M.: Suhrkamp.
- Baecker, D. (2017, March 19). Agilität, Hierarchie und Management: Eine Verallgemeinerung. Retrieved June 16, 2017, from <https://catjects.wordpress.com/2017/03/19/agilitat-hierarchie-und-management-eine-verallgemeinerung/>
- Baecker, D. (n.d.). Komplexitätsforschung II: Design Research [Billet]. Retrieved June 16, 2017, from <https://kure.hypotheses.org/261>
- Brown, T. (2009). *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. New York: Harper Collins.
- Gaver, B., Dunne, T., & Pacenti, E. (1999). Design: Cultural Probes. *Interactions*, 6 (1), 21–29. <https://doi.org/10.1145/291224.291235>
- Gaver, W. W., Boucher, A., Pennington, S., & Walker, B. (2004). Cultural Probes and the Value of Uncertainty. *Interactions*, 11(5), 53–56. <https://doi.org/10.1145/1015530.1015555>
- Gloger, B. (2017). *Scrum Think big: Scrum für wirklich große Projekte, viele Teams und viele Kulturen*. München: Carl Hanser Verlag GmbH & Co. KG.
- Goodman, E., Kuniavsky, M., & Moed, A. (2012). *Observing the user experience: a practitioner's guide to user research*. Waltham, MA: Elsevier.
- Humble, J., Molesky, J., & O'Reilly, B. (2015). *Lean Enterprise: How High Performance Organizations Innovate at Scale* (1st ed.). Sebastopol, CA: O'Reilly and Associates.
- Leffingwell, D. (2016). SAFe 4.0 Introduction. Overview of the Scaled Agile Framework for Lean Software and Systems Engineering. Scaled Agile, Inc. Retrieved from <http://www.scaledagileframework.com/introduction-to-safe/>
- Luhmann, N. (1995). *Social Systems*. Stanford: Stanford University Press.
- Luhmann, N. (2000). *Organisation und Entscheidung*. Opladen: Westdeutscher Verlag.
- McCulloch, W. (2016). A Hierarchy of Values Determined by the Topology of Nervous Nets. In *Embodiments of Mind* (3rd ed., pp. 39–45). Cambridge, MA: MIT Press.
- Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. Portland: CRC Press.
- Weick, K. E. (1979). *The Social Psychology of Organizing*. Reading, MA: Random House.
- Wergin, N.-E. (2003). Teamwork in the Automobile Industry – An Anglo-German Comparison. *European Political Economy Review*, Vol. 1(No. 2 (Autumn 2003)), 152–190.
- White, H. C. (1992). *Identity and Control: A Structural Theory of Social Action*. Princeton: Princeton University Press.

White, H. C. (2002). *Markets From Networks: Socioeconomic Models of Production*.  
Princeton: Princeton University Press.

White, H. C. (2008). *Identity and Control: How Social Formations Emerge* (2nd ed.).  
Princeton: Princeton University Press.