



Using Design Methods to Establish Healthy DevOps Practices

ARAS BİLGEN, kloia

Our clients were not able to articulate their needs during their DevOps transitions and kept asking for simple tool replacements, which would not be sufficient solve their problems. To show our clients that they needed to consider their process and culture along with tools, we borrowed methods from design and brought these into DevOps project planning. We were able to do this with a simple training and the right mindset despite the fact that we are not designers.

1. INTRODUCTION

kloia is a small company of seasoned DevOps, cloud, architecture and product development people. We have 20 full-time consultants and around 10 part-time collaborators, located in offices in London and Istanbul. We share the common belief that software practices evolve every day, and that builders of technology should be the first ones to benefit from this progress. We have very strong opinions against “monkey jobs” – tasks that could be eliminated through automation, so that we can focus on more valuable, creative, and enriching aspects of software development.

For us, DevOps practices represent the better way of building software that is fit for the modern era. We believe that any company that builds and uses software should be able achieve more in shorter time with higher quality and better satisfaction with DevOps. We therefore work with companies of any size. Our portfolio includes big corporations like HSBC and Huawei, as well as small startups who are just at the beginning of their journey.

We strongly believe in the value of end-to-end agility in any business, and our collective experience has shown us that this can be achieved when three things are in place, at the same time: Right tools, right processes, and the right culture.

This is very easy to say, but it was very hard to convince our clients to consider. In this paper, we will talk about the challenges we faced with companies who wanted to start with DevOps but were just focused on tools. After a few unsuccessful attempts in trying to convince them about the importance of processes and cultural changes, we decided to borrow methods from an unlikely discipline: Design! We revised our project planning approach based on human-centered design methods and we are happy to share some of our successes and our learnings in this paper.

2. TOOLS, PROCESSES, AND CULTURE

When a company equips their employees with the right tools, empowers and protects them with the right processes, and works actively to foster a culture that nurtures them, they achieve amazing results.

Unfortunately, this is not a common view in the market today. A lot of companies feel that they could just purchase the latest tool and get incredible efficiency gains. This is not true. We have witnessed tremendous spending on the latest and the greatest tools by companies who made these purchases with no input from the actual people who would use these tools, only to see their investment die, cause a vendor lock-in, and pile more onto the company’s technical debt.

A lot of companies think that they could just read a few articles about popular processes and transition into them overnight. This is not possible. We have seen companies who completed their “agile transformation” by slapping project managers with Scrum master titles and enacting theatrical meetings with no value to the team. This expectation is so widespread that there is a whole industry for selling training and certificates for companies who believe that they can purchase processes and implant them instantly. (Fowler) We have seen companies ripped off by so-called Agile coaches who have no qualms about suggesting practices they have read about in other’s reports, with no personal experience on the methods they have recommended; or so-called

experts who blame their clients when their recommendations blow up miserably because “they didn’t do it right”.

3. CHALLENGE: COULD YOU JUST SET UP JENKINS FOR US?

We share these wrong approaches with every prospective client, but we are faced with an issue: They just want a tool to magically fix their issues and not touch their processes, and not talk about their culture.

As consultants we have a choice at that moment. We can either give the client what they want, knowing that it would not solve their problem, but we would still get paid; or we could take a risk of losing the lead and insist on solve the problem differently, once and for all. It is our company policy to do the latter, but this proved to be harder than we thought.

When we describe how a lasting transition needs to be holistic, we usually hear pushback from our potential clients. (Bilgen) Despite the fact that the issues they shared with us during sales cycles could be solved easily with process changes and some cultural interventions, they insist on only new tools. We also hear a set of excuses about processes and culture. Some of the popular ones are:

- We did our digital transformation last year
- We are already agile
- We did the DevOps project last year
- This project is approved as a tool change project
- We don’t have time to change processes

We needed a way to shift the conversation from these excuses and to convince them that they are about to waste their money and time on a project that is bound to fail. We thought that well-researched case studies about why tool-only approaches fail would convince the clients. We did a really thorough review of the DevOps, agile, and management literature on why complex organizational problems could not be solved only by tool changes. We updated our company presentation, created additional follow-up material, and got ready for our meetings.

Unfortunately, it didn’t work. Our potential clients appreciated that we have prepared such an extensive review for them. They were happy to hear how other companies carried out successful DevOps transitions by considering tools, processes and culture together. When we asked them how this applies to their situation, they told us how different they are from the examples we have shown them, and that just a tool replacement is all they need. These are the most common explanations they came up with:

- These are from smaller companies
- These are from bigger companies
- They have a smaller IT department, so they can be nimbler than us
- They have a bigger IT department, so that they can allocate resources for this
- They are not in the same sector as we are
- We are in different cultures (referring to the culture of the country)

Their closing remarks for the meetings was “So, when can we have jenkins?”

This was very demoralizing because we wanted to address the underlying issues, not just putting a band-aid on the problem. The client had a deeper problem, but they were either not willing to acknowledge it, or they were not aware of it. We needed to find a way to illustrate that the solution to their problem should include the human elements, not only the tools.

For this, we thought about design. Design is all about understanding human needs and coming up with solutions for these needs. Designers have been working with this approach for a long time and the current set of methods they are using are very mature, allowing them to understand users deeply even if the users cannot articulate what they need directly. We have discussed the potential of bringing in some of these methods into our practice and using them for planning DevOps projects.

4. DESIGN PRINCIPLES IN DEVOPS

4.1 Five Design Principles

There are five principles that great designers use to keep the user needs in mind and craft user-centred solutions (Norman, Moggridge):

- **Working Directly with End-Users:** Good designers work directly with the actual end-users of the system they are building. They do not use proxies, they do not talk to managers, or to product owners who claim that they represent the users.
- **Welcoming Ambiguity:** Designers welcome ambiguity throughout the design process, especially at the beginning. By tolerating ambiguity, designers are able to hold multiple possible solutions in mind, and decide as late as possible in the cycle.
- **Giving Form to Ideas:** Good designers do not just talk about ideas, they give form to these ideas through prototypes and models. These activities externalize the ideas and put them out there for critique and improvement. Giving form to an idea detaches it from the person who came up with the idea, and therefore makes it possible to critique the idea without critiquing the person who came up with the idea.
- **Co-Creating in a Safe Space:** Good designers create in safe spaces. Design studios are collaborative settings where people from different seniority levels come together to solve a design problem together. Basic rules govern design studios so that everyone is comfortable in contributing and critiquing each other's work.
- **Experimenting & Revising:** Good design rarely comes in a single shot from heavens. Good user experiences are a result of constant experimentation and revisions. Good designers are very knowledgeable about this, and they are comfortable with iterating on a solution for a long time.

When we look at how DevOps transition projects go in the market, we see the opposite of these principles. Some best practices are considered sacred and spending time to challenge how they may fit into an organization is considered a loss of time. There is an inclination towards action, which is great, but this ambition grows small technical interventions into large projects. All of this is done with little to no concern about the people who will be impacted by these changes. They are rarely consulted, saying that the project is an infrastructure project.

We felt that there is a great room for improvement here. Based on the design principles above, we identified 31 methods that we can use to plan DevOps projects with a more human-centered approach. (kloia) We have leaned these methods out to the point that we could execute 3-4 of them in 2-3 weeks with the entire team. We did not position this approach as an add-on during the sales process. This was the kloia approach, there is no other way to begin a project.

5. HOW DID IT WORK?

We are very happy to see that our approach worked well. We are featuring two cases that demonstrate our approach in detail. Some details in this section are omitted or summarized to protect the business-critical information of each client.

5.1 Softtech/İşbank

Softtech is the technology provider that empowers Is Bank, largest private bank in Turkey, and all of its subsidiaries. As a software company in finance, Softtech seeks to develop solid, banking-grade software solutions, and they are constantly seeking ways to produce better quality software in shorter time. For this goal, they have kicked-off a DevOps initiative, but they have not seen visible improvements.

Softtech was very aware that they are not progressing at their desired velocity and they were humble enough to accept that they would need external help. This attitude was very valuable for us as a start. Softtech started the conversation about their needs around a particular build/release tool. We have acknowledged their need for changes made to the tool, and we advised a wider lens that covers tools, processes, and culture, rather than just looking at a single tool.

To see where the actual problems were, we worked with Softtech for a month. We looked at the company from a holistic perspective – not just from a technical point. To do this, we used four methods:

1. **In-person interviews:** We held 40 hours of interviews with stakeholders from all levels, including junior team members all the way up to directors. We took extra care to make sure that the interviews took place in person. With the exception of a few sessions, all interviews were attended by two kloia employees from different technical backgrounds. This formation made the notes more reliable and made facilitation easier. Analysis of the interviews yielded richer results thanks to the participation of different disciplines, therefore different interpretations. The biggest take-aways from these interviews

were not the technical details of the IT systems. We were able to understand the historic and political aspects of certain technical decisions and how they affected the current flow of work at Softtech.

2. **Hands-on code reviews:** To understand the complexity we are working with, kloia architects and leads have done technical deep-dives with Softtech teams. We examined the overall project architecture, DevOps pipelines, dev and test environments, test automation capabilities, and release processes during these deep-dives. We held an additional session to hear about security concerns.
3. **Value stream mapping workshops:** We looked at the flow of activities through a value stream mapping workshop. The goal of the workshop was to illustrate the value stream that begins with the formation of an idea, all the way down to when it breaks in production. This workshop was so detailed that it took us 3 days to complete. It produced amazing insights thanks to the heartfelt involvement of the entire team we worked with at Softtech. The biggest value of the workshop was that it was bottom-up. We heard many of the stories we heard during interview come up again in the VSM workshop from different parties. These comebacks affirmed our initial findings. Moreover, they were moments of sharing among the team in a safe space.
4. **Diary studies:** The value stream mapping workshop revealed potential pockets of inefficient use of time. We carried out a two-week diary study with developers to understand the root cause of these inefficiencies. At the end of each day, each developer wrote down notes about how their day went, what kind of work they got done, and what impediments they had. We periodically checked in with each developer to reflect on their notes together. This study revealed additional insights about why the particular tool itself was not the problem.

In just a month, we have examined all departments related to SDLC and offered a multitude of solutions on multiple time horizons. Some of these solutions were prototyped with proof of concepts. Our final presentation was received very well by the upper management due to its holistic nature. One general manager commented that each topic in our presentation had been raised to him in the past by independent parties, but this was the first time he saw such a candid summary and related steps to be taken.

Softtech was able to internalize our recommendations and focus their DevOps efforts differently based on our input. At the end of the project, Softtech realized the hidden potential of their work culture. They took steps to make new technological investments in measurement and test automation to increase software quality and make collaboration with business units easier. The issues about the build/release tool were recognized, but not prioritized. From our perspective, this was a big indicator that our approach was successful.

5.2 Huawei Turkey R & D Center

Huawei is a global provider of information and communication technology (ICT) infrastructure and smart devices. Innovation and research are the key concerns of the company, and Huawei has been investing over 10% of its annual revenue in R & D. Huawei Turkey R & D Center (HTRDC) started its operations in 2010 and currently it is the second largest software R & D center out of China. HTRDC delivers projects to more than 40 customers in 30+ countries and more than 450 engineers work behind many Huawei technologies.

Huawei has a fierce business environment. Huawei is committed to meeting customer needs and is known for meeting the impossible deadlines of their clients, therefore, time-to-market rules the daily operations. To achieve this speed, priorities and teams in Huawei are very fluid. This extreme agility makes evolving or developing new processes a challenge. Moreover, Huawei units in different countries have a strong tie to the headquarters in Shenzhen, which can make new process approvals take a long time. On top this, Huawei is an international ICT company, which implies that they need to meet stringent regulatory standards in multiple jurisdictions.

Within this landscape, different software teams in Huawei Turkey R & D Center have experimented with different DevOps practices and tools, but achieved limited success. The Quality and Operations team recognized this inefficiency. They asked for kloia's guidance to make these efforts succeed permanently and become a part of the daily operations at Huawei Turkey R & D Center.

To understand what has been done in the past and what can be done to institute lasting DevOps practices, we used four methods:

1. **In-person interviews:** We held 20 hours of interviews to understand the history of the DevOps methods tried at Huawei. Almost half of these interviews were group interviews, due to the difficulty of booking team members for interviews. This difficulty was an evidence about the tight schedules and heavy workload at Huawei. We made sure to bring this up during the interviews as a question, and it

acted as a starting point for discussing what stops them from completing more work in less time using DevOps practices.

2. **Value stream mapping workshops:** We ran an extensive value stream mapping workshop with the teams at Huawei. The goal of the workshop was to illustrate the entire value stream for the teams at Huawei. After illustrating the entire flow, the teams divided into pairs and issued recommendations for each step. We calculated the efficiency of the teams at Huawei and analyzed the finalized flow to point at possible wastes.
3. **Challenge mapping workshops:** We had a long list of problems and possible solutions from our interviews and the value stream mapping workshop. However, we needed additional context around these recommendations for two reasons. First, our interviews were aiming for depth of understanding, not breadth. Therefore, the recommendations we hear during the interviews ran the risk of being only locally applicable. Second, the value stream mapping workshop was conducted with multiple teams. There were instances where recommendations from different teams conflicted with each other, because each team assumed their views in how the problem should be fixed.

For these two reasons, we wanted to bring everyone in to see how teams articulate their problems and map their possible solutions against them in a challenge mapping workshop. Challenge mapping is a creative problem-solving technique that alternates between brainstorming and critical thinking. (Basadur).

The teams at Huawei were able to generate around 50 solutions during the workshop. Some of these solutions were trivial improvements that the teams could immediately take back to their work. On the other hand, some of them required buy-in from multiple teams and strong championship from the executive team. Despite the low feasibility of these complex solutions, we kept them in the final report to give a context about what can be done further. Because the executive team would not have the time to review 50 recommendations at once, the teams classified their solutions into four thematic buckets, which were then presented to the executive teams with examples at the end of the project.

4. **Proof of concept development:** The teams at Huawei came up many good ideas during our sessions, and some of them were only technical in nature. To be able to see if these solutions have merit, we have built multiple proof of concept systems with them. This approach is very similar to prototyping in design. We have timeboxed our involvement in many cases, so that we could finish the PoC, get feedback from the potential future owners of the PoC, and then iterate. We have applied this approach extensively to build a pipeline that meets the needs of the Huawei teams.

At the end of two weeks, we came up with a list of quick wins that address issues in many aspects, from culture to deployment pipelines, from team structures to firewall configurations. After this initial discovery and planning phase, we spent about two months with the teams to make tailored DevOps practices a part of their daily flow. We have also raised the knowledge and awareness level of all interested parties through training and informal meetups.

Huawei teams were able to identify the DevOps processes that work in their environment and build a pipeline that met their needs. The practices were also codified in a checklist and made part of the recurring quality audits. Huawei continues to invest in continuous improvement of their software teams. We are glad to hear reports from our acquaintances that the changes we made in 2018 are still in effect today.

6. WHAT WE LEARNED

Based on the outcomes in our clients and their feedback, we are glad to see that our approach that incorporated design methods to understand our clients worked well. We are happy to see that we found a way to have a broader conversation with our clients, hear their needs as fellow human beings, and foster a more meaningful connection during our work. We feel that we were able to suggest better, more tailored solutions to our clients based on this connection, whether it is in the technical or in the cultural domain.

We have learned three important lessons throughout this journey:

6.1 Mediocre consultants are the biggest enemy of good consultants

We feel that our approach that incorporates design methods made us better consultants. But we did not come up with this idea solely by ourselves – mediocre consultants forced us to consider something radically new.

The reason why we had difficulty with our clients at the beginning was because they had contact with dishonest consultants who claim to solve all of their problems overnight with a tool replacement, or with a

magic certification course. We are glad that we now have a way to cope with this issue, but we would like to flag this sales-oriented, unethical behavior in our industry. There is a good discourse about the “Agile Industrial Complex” and we encourage further discussions and awareness in this aspect. (Fowler)

6.2 You don’t have to be a designer to apply design methods

When we decided to use design methods for our projects, we felt a brisk wave of internal resistance. My peers at kloia felt that this approach could work, but they had one big doubt in their minds: “We are not designers, how are we going to apply these methods?”

Their doubt was partially founded. No one in the company had social studies or design degrees or worked with human-centered design practices before. But no one is born with these skills; even the most successful designers and researchers learn these approaches at school, with professional training programs, or through their practice at work.

We did the same. I have put together an overview document that covers research fundamentals and simple description for each method we will use. We then paired up to run these sessions together. After each session, we held a small retrospective to review where they struggled with and I offered coaching tips for the upcoming sessions. It took us 2-4 months to get our digital transition team up to speed with the methods. Their design research skills are not on par with designers with years of training. But that is OK, because we are learning with each new client and our knowledge is sufficient to address the needs of our clients in the DevOps space.

6.3 Being an expert in a technical domain should not be an excuse for omitting the humans in the equation

The DevOps community is very good at coming up with solutions. They know how to identify bottlenecks, how to measure things, how to rearchitect systems, and how to achieve significant improvements. They are very good in coming up with solutions.

But every now and then, a project shows up where the bottleneck in the system is not technical – it’s human in nature. This is where the experts in the technical domain struggle. Tool changes or process interventions can relieve the problem temporarily, but one needs to look past the technical realm into the human nature to truly understand the problem and offer solutions.

We found out that this is possible by getting inspiration from a human-centered field: Design. But we do not think that design is the only discipline that offers this broad insight. If you are striving to create better solutions that address technical and human aspects equally well, we invite you to take a step back and embrace the human nature, curiosity, and emotions just as strongly as the technical limitations of your client.

7. ACKNOWLEDGEMENTS

We consider ourselves lucky to have a chance to work with the amazing teams at Softtech, Huawei and many companies we have not had a chance to name here. I would like to thank my shepherd Kevin Normand for his patience and timely guidance in giving this experience report its final shape.

REFERENCES

- Basadur, Min. "Simplexity Explained - Discover the Processes and Tools for Innovation." Basadur. <http://www.basadur.com/simplexity-explained/>.
- Bilgen, Aras. "How to Prepare Successfully for the Transition to DevOps." Kloia Blog. <https://www.kloia.com/blog/what-makes-devops-transitions-successful>.
- Fowler, Martin. "The State of Agile Software in 2018." Martinowler.com. <https://martinowler.com/articles/agile-aus-2018.html>.
- kloia. "Digital Transition by kloia." kloia. <https://digitaltransition.kloia.com/>.
- Moggridge, Bill. *Designing Interactions*. Cambridge, MA: MIT Press, 2007.
- Norman, Donald A. *The Design of Everyday Things*. Cambridge, MA: MIT Press, 2013.