



Human-Centered Agile: Building Next Generation Agile Programs with Human-Centered Design

JOSEPH MONTALBANO, Noblis
BRAD LEHMAN, ICF

Over the past two years the authors have integrated Human-Centered Design (HCD) thinking and methods into several mature Agile programs at a large government agency with striking results, including more robust end-user insight, earlier delivery of this insight, stronger user stories, higher product quality, shorter lead time, and most important of all, better informed Product Owners. This integration took longer and was more difficult than expected not only because HCD practitioners have different mindsets and approaches to obtaining insight into end-user needs than Agilists, but also because Agile teams often struggle to plan HCD work that has to be planned a little ahead of development work. After multiple experiments, we came up with a topology for HCD teams that worked in our scaled programs. We then developed and refined the concept of a Research Roadmap to help teams visualize and plan HCD work ahead of development work without building a queue. This paper outlines our experience and articulates the basics of what we have come to call “Human-Centered Agile.”

1. INTRODUCTION: OBSERVING HUMAN-CENTERED DESIGN AND AGILE IN ACTION

It was after hours when the Agile Coach’s phone rang. It was the client’s senior executive, and he was requesting research about a new buzzword he had heard from a colleague. In decades past the buzzwords discussed on such calls were Joint Application Development (JAD), Rapid Application Development (RAD), the Rational Unified Process (RUP), Spiral, and others. The subject of this request was “Human-Centered Design” or HCD. That caught the Coach a little off guard. Sure, he had heard the term Human-Centered Design before, but he wasn’t sure exactly what it was or what differentiated it from regular Design that wasn’t Human-Centered. Whatever this new three-letter acronym was, he was pretty sure that he didn’t need it. The ask was simple: The Agile Coaches were to observe a planning session (“PI Planning” in SAFe terms [ML]) for another program that was using HCD and having success with it and consider how HCD could benefit their own program.

On the appointed day, the Agile Coaches and Scrum Masters made the short drive to the facility where the planning session was held, making sure to pack their laptops because they expected to be bored. When they arrived at 8 AM, the facility was abuzz with the activity of over a hundred people arriving. There was food (not just donuts, but a spectacular buffet with chef-prepared food). It was obvious that great care had been taken when preparing the event; printed agendas were on all the chairs and there were snacks, mints, and fidget toys on all the tables. A table full of coffee, fruit, snacks, water, and soft drinks was in the back of the room. The Coaches and Scrum Masters secured a table, got some delicious food, plugged their laptops in, and started to catch up on emails. They didn’t expect that they would be interrupted, and certainly not so soon.

An executive stood and spoke to the group about the work they were being tasked with and launched into a detailed description of why it was important and how it tied into the overarching strategy of the agency. The Coaches continued to eat. It wasn’t until the Product Owner (PO) stood up to give an impressive briefing that they dropped their forks. Aligning large programs around a common vision is difficult, and this PO was a rock star at it. Some stakeholders were asking for the inclusion of new features in the roadmap, but the PO demurred, saying she didn’t want to build a queue. It was the smartest and most deliberative Product Roadmap discussion the Coaches had ever heard. It was impressive. After the PO spoke, an HCD team presented the results of the discovery they had done, aligning everyone around user needs and sentiment for the upcoming Sprints. The PO then deftly tied that insight to the features she had prioritized for the upcoming Sprints and began talking about her priorities for the next Sprints. This was even more impressive. These teams were getting a “head start” because of the quality of the user sentiment they received at planning. After the briefings, HCD experts dispersed among the development teams, so that each team could ask questions about the insight they had just heard. HCD experts then helped development teams apply this insight to their stories and

acceptance criteria. It was the best planning the Coaches had ever seen, even better than their own. It sure seemed like this program was on to something big.

Augmenting the planning session with HCD insight was elegant in its simplicity and exceptionally effective. It looked deceptively easy. The Coaches went back to the office and started Googling Human-Centered Design expecting to be proficient practitioners before dinner. They were shocked at the breadth of the subject matter and at the depth of the rabbit hole they found themselves in. Sure, their own teams had been applying Behavior Driven Development (BDD) to improve their stories and acceptance criteria for a while. They were aware of artifacts like personas and journey maps, and on occasion they even used them to document user needs and sentiment. These tools, however, were not an integral part of their discovery process and the PO was still the font of most product and user knowledge for these teams. The Coaches were looking for ways to change that on their programs and emulate the insight and ensuing success with HCD that they had just witnessed.

As the Coaches and Scrum Masters learned more about HCD, they became increasingly convinced of the utility of HCD activities and artifacts. They also became aware they knew less about this somewhat alien way of gathering user insight than they'd assumed. They were clearly descending the Dunning-Kruger curve [KD] and figuring out that HCD isn't just a set of activities and artifacts, but like Agile, it is a mindset.

2. BACKGROUND

Our client, a large government agency with a trillion-dollar budget, realized that although they were delivering plenty of features regularly, their overall outcomes were not quite what they intended. There was still a gap between what users wanted and what they were getting. This gap was large enough that some industrious end-users were looking for alternate solutions to the ones developed by our client. Thankfully, our client had the clarity of vision and self-confidence to recognize and address the problem. Upon seeing the success HCD methods and thinking had brought to other programs, the client, who was used to augmenting their workforce with armies of contractors, decided to incorporate HCD by simply contracting an HCD-focused company to join forces with their existing vendors.

The authors, an Agile Coach and an HCD practitioner, met when they were tapped by their respective companies (technically competitors) to make this integration work for their common client. This paper, based upon our book about our experiences titled *Human-Centered Agile*, forthcoming from Productivity Press, will discuss some of our challenges and lessons learned from this integration. The first problem we encountered was the lack of common understanding of what HCD was and the vocabulary around it. Before we begin, let's define HCD and introduce some terms you might hear when HCD is being discussed.

2.1 What is "Human-Centered Design," anyway?

For the uninitiated, the most disconcerting thing about Human-Centered Design (HCD) is the lexicon. There are hundreds of buzzwords around design, such as User Experience (UX), Lean UX, Customer Experience (CX), Service Design, and Design Thinking, and that's before you get into specific HCD activities or artifacts. It's easy to get overwhelmed and feel like you are being asked to fight your way through a new language with no dictionary.

At its core, HCD is a combination of research, specifically user research, and design that is meant to keep the user in view throughout the entire product delivery lifecycle. Goals are set with the user in mind. Success is defined by seeking and incorporating user feedback along the way so that when we create products, we are doing so in a way that reflects not just their needs and goals, but their attitudes, behaviors, and expectations as well.

This typically starts with discovery, where you learn the basic information about users that help you get started and make early decisions and continues through early conceptual designs all the way through the release and evaluation of your "final" product (as if products were ever final these days), so that your learning informs your strategy in an ongoing way.

When conducted properly, HCD combines the designer's natural curiosity with a team's willingness to think along multiple design paths, accelerates learning, and allows teams to make decisions earlier in the cone of uncertainty than they had previously been able to do (a big deal!). Because the goal is to deliver products and experiences that are highly valued by the user, HCD balances the need to gain user understanding with the risks and stakes of delivery, so that research, design, and implementation are aligned for the greatest risk reduction and cost savings to the team.

One of the evergreen pitfalls of design, and especially Human-Centered Design, is that the word "design" itself is prone to many misunderstandings. People will often think of visual design, or use terms like "look and

feel”, and treat “design” as a coat of paint that follows all your important decisions. This, by the way, is one reason why one hears terms like “UX,” “UI,” “CX,” and “HCD” to try and distinguish from “design.” The problem is that those terms have become buzzwords that have different meanings to different people. Vocabulary around design is frustratingly slippery.

To be clear: Human-Centered Design is the combination of research and design. Without research, there is no “human” in Human-Centered Design. Too often, Human-Centered Designers are parachuted into efforts of various sizes, left to fend for themselves. It becomes their job to persuade their own teams of the value and necessity of thinking about users, not just delivering functionality. We can do better than this, and so can you.

3. OUR STORY: ONBOARDING HCD TO OUR OWN PROGRAMS

Word of the successful and high-quality planning session that the Coaches had witnessed spread quickly. With news that dedicated HCD professionals were going to be joining our programs, expectations were high. Due to the vagaries of government contracts, we didn’t know exactly when our new HCD colleagues were going to arrive. We usually get at least a little notice when vendors are added, so we assumed that we would get at least some chance to plan for their arrival and integration. We were wrong. That turned out to be the genesis of our next problem (a problem closely related to the lack of shared understanding): it was difficult to find a good HCD “entry point” for work that was already planned and scheduled.

Almost immediately after their abrupt arrival, a program to which the HCD experts were assigned was having a big planning session. The planning session seemed like the perfect opportunity to introduce them to the nine Agile teams and approximately 100 people on the program. As it turned out, this wasn’t so perfect because the features to be built over the next 5 Sprints had already been selected and refined. Some of the stories to support those features had already been written, and teams were going to write many others during the planning session.

What’s more, while program leadership felt that outcomes had plenty of room for improvement, this was not necessarily the case at the individual team level. Each of the program’s Agile teams had reached its own comfort level at delivering features on a consistent basis and improving them based on user feedback, so to them, adding HCD and all the accompanying questions felt like an interruption or an impediment, or both.

When the HCD folks started asking questions, the questions were sometimes difficult. Teams were used to thinking in terms of delivery—what was the functionality that the client was asking for, and how might the team deliver it efficiently? What were the technical dependencies? Estimation, prioritization, capacity allocation. The classics. The HCD folks started introducing questions more like:

- “How do you know that’s what the end-user wants?”
- “How will you figure out whether you made a good version of the feature or a bad one? What will we be measuring post-release?”
- “What problem is this feature trying to solve? What other ways did you consider solving it?”
- “What are the stakes of release? What would the consequences of a bad release be?”
- “How many of our assumptions can be validated before a team spends time building the solution?”
- “If it goes out as planned, which decisions can we actually iterate on, and which decisions have passed any last responsible moment for further change?”
- “We **will** be learning and iterating, right?”

Sometimes this was a little uncomfortable, but that certainly wasn’t the intent of the questions. A few teams engaged with these questions, which was fantastic. Most teams felt, for lack of a better word, invaded. To the Coach’s horror, some teams felt as if their own (pre-HCD) design efforts were disrespected and given short shrift by the newly added HCD experts. These interactions improved over time and with repetition—the next time an HCD practitioner asked these questions, it was a bit more familiar and less threatening. As time progressed, spirited but productive discussions became common, and friendships were struck (as evidenced by this paper). But there were still difficulties to be worked through.

3.1 The Challenges of Onboarding New Skills and People

We’ve alluded to the first problem that we encountered, which is that teams and leadership didn’t have a shared understanding of what HCD (or UX, UI, CX, or “design”) was supposed to do. Some designers got presented with mostly completed features and asked to “add some HCD to it” as it was in-flight. Some designers were asked to adjust “look and feel” to make the product “intuitive” without any attention to what that might actually mean or how they might accomplish it. Sometimes, the dysfunction took the exact opposite

form—a PO had an idea for a new feature, and the ask would come in the form of a one-sentence description of that feature, with the request to “get mockups by Thursday” so that a team could immediately start work on it.

Designers were left to explain that research is part of the process, which then only led to concerns that any attempt at proper HCD would create runaway timeframes and delays. Having spent most of their time optimizing the delivery of solutions, POs and teams didn’t initially tend to think of HCD research or time validating intended designs as “real” work.” Real work, of course, is what the developers do.

Left to their own devices, HCD practitioners on individual teams tried a variety of methods to integrate their work with their teammates:

- Some prioritized work that was further down the backlog, so they could try to do discovery activities ahead of the feature delivery.
- Some did heuristic reviews and made good-practice improvements to the designs that were about to get shipped out the door, usually in the form of small interaction or labeling improvements.
- Some worked on creating documentation for their teams to use later, such as style guides and pattern libraries.
- Some found opportunities to do collaborative workshops to provide clearer shared problem definition, especially where none existed before, with success metrics, and to engage in collaborative ideation. These were well received and, in retrospect, were critical to the successful integration of HCD.

Overall, the results of these efforts to integrate this work were uneven—some teams reached a next step of design maturity, some teams seemed stuck and unable to integrate their designer. There were a wide range of activities taking place and artifacts being generated, such that teams often felt like HCD meant completely different things. POs that spoke to one another would find that one designer was spending their time polishing interaction design for near-release features, while another might be trying to conduct user interviews for a feature planned for construction further in the future. Individually, none of these approaches were incorrect, but they didn’t add up to a consistently improved user experience.

We learned that having HCD expertise at the team level was good, but this team topology didn’t provide the lift we were looking for from this integration. It was a good way to ease new team members into the work in a way that would get them acquainted with the details gradually, but we found that this topology didn’t allow HCD practitioners to do anything more than limited ad hoc activities with very short durations. We wanted more.

3.2 Poor Initial Team Formation Created Silos of HCD

At the beginning of the integration HCD practitioners were assigned to Agile teams where they were with developers but separated from one another. Each designer faced their own specific hill to climb, in terms of gaining buy-in from POs, gaining access to users, or otherwise trying to incorporate basic HCD methodologies into their work. This didn’t work so well.

HCD practitioners teamed up with the Agile Coaches and started lobbying leadership directly for their own shared services team that could provide mutual support across programs and support an integrated UX vision. Fortunately, there was enough leadership support that we were able to form a shared services HCD team with an engaged PO and a talented Scrum Master. Along with the new shared services HCD team, they were able to organize and prioritize support work for HCD practitioners and to establish communications and outreach so that all the Agile teams could share learning and get the most out of HCD methods in their local context. This understanding still came primarily in the form of conversations but having an official team with a client stakeholder that could bridge the gap between consultants and client stakeholders was a big win.

Previously we only had some HCD practitioners at the individual team level, but now we had more. HCD Practitioners now had an internal advocate and clearer communication channels (Figure 1). This shared services HCD team could now do:

- Strategic planning
- Outreach and education to stakeholders and teams
- Collective recruitment
- Swarming where necessary
- Coaching

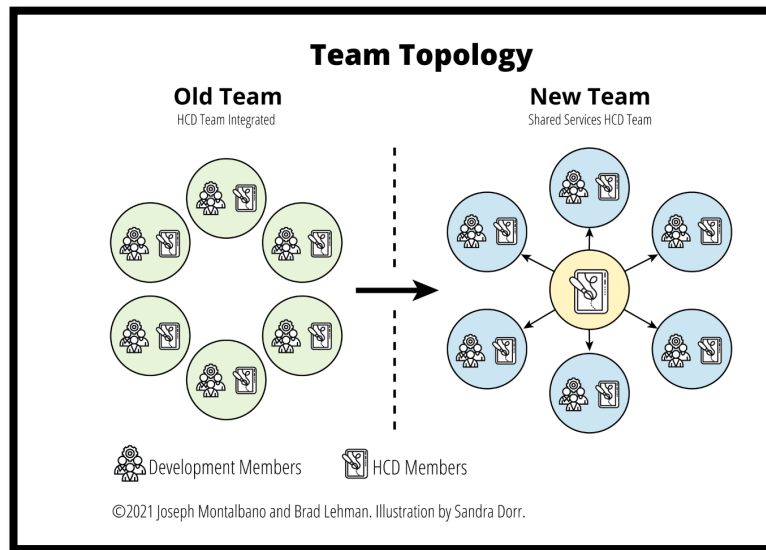


Figure 1. The initial team topology with HCD practitioners on each team (left) and the topology we ended up with, the shared services HCD team as well as HCD practitioners at many individual teams (right).

These activities provided great value to the program and began the hockey-stick improvement that we saw with the addition of effective HCD methods. Let's look more closely at some of what worked here.

Strategic HCD Planning: Weekly Meetings. A small planning team comprised of the HCD PO, Scrum Master, and most of the shared services HCD team met weekly. This helped the HCD PO (who was fairly new to the idea of HCD) have deeper conversations and get support from experienced practitioners in how to discuss HCD with other stakeholders and how to think about integrating the work better and more consistently. Most of the experiments seen below that became practice were a result of this level of PO strategic engagement.

Outreach was a particularly rich area of experimentation, and it could only have been accomplished through shared services. In addition to informal communications from the HCD PO to other POs and stakeholders, the team set about creating a few different approaches to outreach to the various teams, and to getting better buy-in and shared expectations, so that the incorporation of HCD could be successful.

Outreach: Workshops. With leadership support (i.e., participants were conscripted or, to use government vernacular, they were "voluntold"), we engaged teams in two-day workshops *that included users as participants* designed to create user-focused problem definitions, success metrics, and solution ideation. These workshops were particularly successful in resetting release goals and focusing on understanding what would create value for the user. While there was some initial resistance to committing teams to workshops for two whole days, the impact was so great for the first teams that this became a standard practice across programs to jump-start a deeper HCD engagement and create better overall product outcomes.

Outreach: Open Studio/HCD 101. The HCD PO and practitioners created an "Open Studio" format (similar to what Agilists might know as a "Lean Coffee" format) in which any stakeholder or team member from any team could come with questions for HCD practitioners in a less formal and more conversational format than other planning sessions. Pastries were provided to encourage people to stop by (this really worked, by the way). In addition, stakeholders and team members could sign up for an "HCD 101" hour-long orientation that was similar to a lunch-and-learn, but was created by the HCD team specifically so that the language and descriptions of the approach to HCD would be consistent, and could become familiar. At a minimum, this gave us a common dictionary.

Establishing Collective Recruitment. Teams were able to overcome one of the constant HCD struggles of siloed engagement by pooling their participant recruiting resources. This allowed each team to have access to a wider group of users (and, in some cases, their *first* group of users), while also creating templates and practices for reaching out to participants in a consistent manner and tracking participation to avoid participation fatigue, which was already becoming evident.

Team Support: Coaching and Swarming Opportunities. We often speak of HCD as one skill, but in fact it is a blend of many, incorporating both design practices and research practices. One of the subtler but very valuable outcomes of having a shared services team was that embedded team members had a place to go for

coaching and reflection in areas that were not natural strengths for individual designers, and to discuss both the approach to work on their team and their actual deliverables. We could observe considerable HCD skill growth in a way that was not possible with embedded and siloed designers alone. Additionally, because HCD work can happen in slightly uneven waves, the ability to swarm from a shared services team to areas of immediate need was a clear and immediate benefit.

3.3 Teams Struggled to Plan HCD Work

This disfunction only emerged after the formation of the shared services HCD team and several highly visible successes. We found that the teams were suddenly struggling to plan their work. It was a strange thing for a program that, minus the HCD folks, had been together for a long time and was mature in their practices. It took us a few Sprints to figure it out, but the issue was that the Agile teams weren't used to planning for work that wasn't consumed in the current Sprint. Some thought it was "wrong and anti-Agile." They were used to picking up features, writing stories, and delivering them in one Sprint.

Identifying and prioritizing the HCD work to be delivered in future Sprints had added complications, with teams effectively competing for practitioners and needing to align HCD work ahead of their implementation work in such a way that findings could still be meaningfully incorporated into delivery.

3.3.1 The Research Roadmap

To create a good understanding of how much HCD work was being requested, what type of work it was, and what teams the work was attached to, we came up with the concept of the Research Runway (Figure 2). This helped teams plan HCD work based on type of activity, just ahead of the next stage of work (sometimes design, sometimes implementation) so that it was both ready and fresh when the Agile teams needed it. We think of this as a visual representation of the HCD backlog, and it very useful in helping guide priority discussions. By seeing all the work that was intended, it became easier for POs to decide which research was higher priority when choices had to be made.

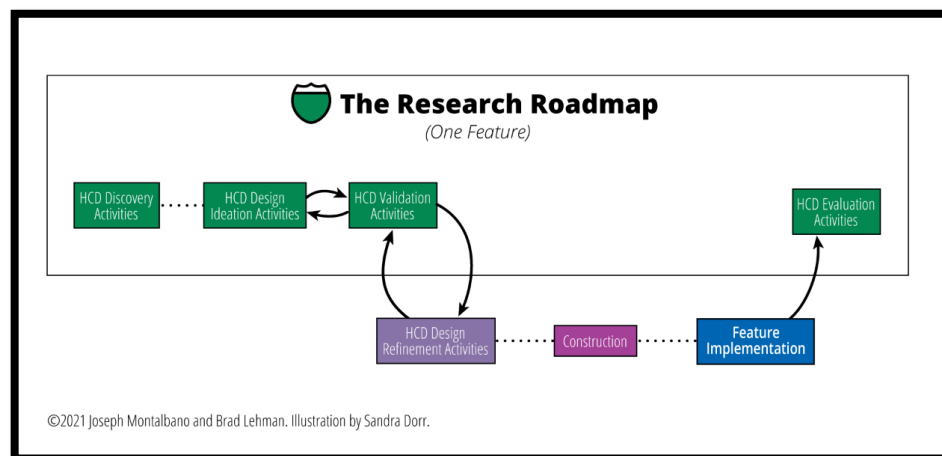


Figure 2. Activities that make up the Research Roadmap for a single feature.

Stories were created for research tasks and for collaborative ideation, such as the user workshops described above. Each of those tasks was captured onto the Research Roadmap, so that a holistic picture of HCD activity could be seen across the program. This turned into an essential tool for helping us plan HCD work at scale.

In many ways the Research Roadmap was the most straightforward improvement we made. It didn't require much experimentation to get this to the point where POs and teams could see the big research picture and could plan their research better, even if it wouldn't be consumed in the current Sprint. It also allowed each team to clearly see what other teams were doing, which doubled as an outreach benefit. ("Wait, these other teams are talking to users both in the beginning and for validation? Should we be hearing from our users more often too?")

3.4 Not Just Changing a Process, Changing a Mindset

At the top of this paper, we said that “HCD isn’t just a set of activities and artifacts, but like Agile, it is a mindset.” Throughout this experience, we’ve come to believe that while it’s important to have the right processes and team topology, we also need continual reinforcement of basic core HCD values, lest they be forgotten in the heat of delivery (as Peter Drucker reminds us, “culture eats strategy for breakfast”). We found clear patterns of factors that led individual teams to better succeed with incorporating HCD with Agile. The most successful teams demonstrated what we have come to call “Human-Centered Agile,” the simultaneous application of Agile and HCD thinking and practices to deliver customer value quickly.

Key success factors for Human-Centered Agile can be summed up as follows:

- **Have an Engaged PO.** This is listed first because its importance cannot be overstated. POs, both for the shared services HCD team and the Agile teams need to care about user-based outcomes of their delivery, not just outputs.
- **Secure Access to Users.** This is a mental shift for Agile teams and POs as HCD requires access to end-users: Humans. As many as possible. No humans = no Human-Centered Design.
- **Get Permission to Fail Fast.** In our experience, this is often something leaders say but don’t mean. For success, teams really do need to experiment. Not all experiments work. That’s OK. There must be cultural shared expectations and psychological safety about the value of failed experiments.
- **Use Clear Metrics Tied to Design Success.** Metrics should be defined during problem definition, so outcomes are clearer.
- **Allot Time for Research.** Time for discovery and validation must be planned before feature delivery so that teams have time to incorporate findings.

3.5 Results

Our experiments that yielded the largest benefits started with outreach and training. We found that aligning everyone to common expectations *and language* around HCD was critical for success. Our experience is that this is best planned and delivered well in advance if possible. Like Agile training, we found it best for everyone to hear the same message at the same time from the same person.

The second success factor that we identified was team launch. In our case we were at least able to use our sub-optimal team launch to find the configuration that worked best for us: a shared services HCD team in addition to some HCD resources at the team level.

Our third and final success factor was the use of the Research Roadmap to help teams visualize and plan HCD work and to get ideas from the work of other teams. Finally, we also believe that, as a mindset change, continual time and communication efforts need to be devoted to reinforcing specific core values.

The results are clear: Human-Centered Agile is better than Agile or Human-Centered Design alone.

4. WHAT WE LEARNED

It’s worth noting that the integration of HCD methods and thinking into mature programs was more difficult and time consuming than we thought it would be in the beginning. It did, however, generate significant benefits including increased product quality, higher end-user satisfaction, better insight delivered earlier, faster team decisions, and much better-informed Product Owners. In retrospect, it may sound like we had a well-coordinated response to this problem, but we didn’t. In true Agile fashion we found success in a variety of experiments, adapting our process as we went based upon what worked.

Given a chance to do this again, we would much prefer to have started with a team topology that made a little more sense. Our final outcome, going from embedded and distributed HCD practitioners to embedded practitioners with shared service support was a much better approach. We would have paired that topology with a stronger outreach campaign that preceded the arrival of new team members by having POs go through an orientation to HCD (such as the HCD 101, which we eventually developed). The hope is that this would have helped create a shared vocabulary and that the POs would be primed to think about how HCD could be involved. We’d also like to simply give POs a chance to ask the “won’t this slow me down?” types of questions, without leaving individual practitioners to persuade teams about the value of HCD without additional support.

5. ACKNOWLEDGEMENTS

The list of people who share credit for this work is long, and starts with our federal client, who we cannot name, for having the wisdom and foresight to ask for these competencies to be blended in the first place, and for being patient as we worked the bugs out. We would like to thank all the staff at Noblis and ICF for having the commitment and focus to try blending these ways of working, and the courage to speak up when they didn't work. Finally, we would like to thank Michael Keeling, our volunteer guide from Agile Alliance who worked with us through comments, revisions, deadlines, and multiple refactors.

REFERENCES

[ML] Montalbano J., Lehman, B. "Using Human-Centered Design with SAFe" retrieved June 21, 2021 from website:

<https://www.scaledagileframework.com/using-human-centered-design-with-safe/>

[KD] Kruger, J., Dunning, D. 1999. "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments." Journal of Personality and Social Psychology, Volume 77, Pages 1121—1134.