DOI:10.1145/3126156

Gregorio Convertino and Nancy Frishberg

Viewpoint
Why Agile Teams Fail Without UX Research

Failures to involve end users or to collect comprehensive data representing user needs are described and solutions to avoid such failures are proposed.

LESSONS LEARNED BY two user researchers in the software industry point to recurrent failures to incorporate user experience (UX) research or design research. This leads agile teams to miss the mark with their products because they neglect or mischaracterize the target users' needs and environment. While the reported examples focus on software, the lessons apply equally well to the development of services or tangible products.

Why It Matters to the ACM Community

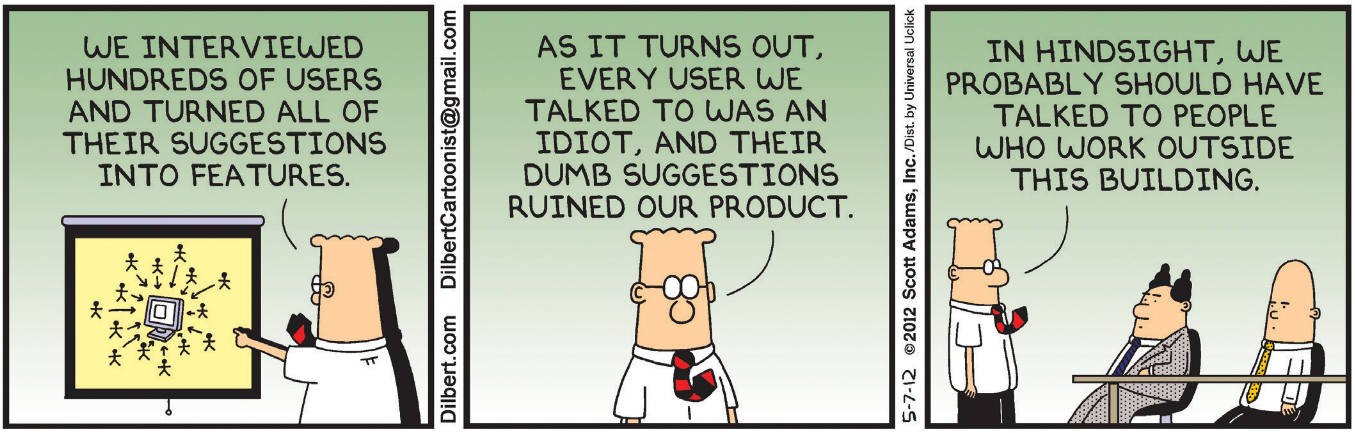
Over the past 15 years, agile and lean product development practices have increasingly become the norm in the IT industry.[3] At the same time, two synergistic trends have also emerged.

- End users' demand for good user experience has increased significantly, with wide adoption of mobile devices. Any new application needs to do something useful or fun, plus it needs to do it well and fast enough. In 2013, technology analysts found that only 16% of people tried a new mobile app more than twice, suggesting that users have low tolerance for poor user experience (UX) (where UX is the totality of user's interactions supported by the app to accomplish a goal).[9]

Even when customers are involved, sometimes the teams may still fail to involve the actual end users.

- With growing emphasis on good UX design, UX professionals, both designers and researchers, are gradually being incorporated as required roles in software development, alongside product managers and software developers. A 2014 Forrester survey of 112 companies found that organizations in which there was systematic investment in UX design process and user research self-evaluated as having greater impact than those with more limited scope of investment.

These trends describe a new context that often finds agile teams unprepared for two main reasons. First, while the agile process formally values the principle of collaboration with customers to define the product vision, we and our colleagues in industry too often observe this princi-


WE INTERVIEWED HUNDREDS OF USERS AND TURNED ALL OF THEIR SUGGESTIONS INTO FEATURES.
AS IT TURNS OUT, EVERY USER WE TALKED TO WAS AN IDIOT, AND THEIR DUMB SUGGESTIONS RUINED OUR PRODUCT.
IN HINDSIGHT, WE PROBABLY SHOULD HAVE TALKED TO PEOPLE WHO WORK OUTSIDE THIS BUILDING.
Dilbert.com DilbertCartoonist@gmail.com
5-7-12 ©2012 Scott Adams, Inc./Dist. by Universal Uclick

DILBERT © 2012 SCOTT ADAMS. USED BY PERMISSION OF ANDREWS MCMEEL SYNDICATION. ALL RIGHTS RESERVED.

ple not being put into practice: teams do not validate requirements systematically in the settings of use. Second, even when customers are involved, sometimes the teams may still fail to involve actual end users. As Rosenberg puts it, when user requirements are not validated but are still called "user stories," it creates "the illusion of user requirements" that fools the team and the executives, who are then mystified when the product fails in the marketplace.[10]

In this Viewpoint, we illustrate five classic examples of failures to involve actual end users or to gather sufficiently comprehensive data to represent their needs. Then we propose how these failures can be avoided.

Five Cases of Neglect or Mischaracterizations of the User

We identified five classic cases of failures to involve actual end users.

The Wild West case. The first and most obvious case occurs when the team does not do regular testing with the users along the development process. Thus the team fails to evaluate how well the software built fits target users, their tasks, and their environments. A real-life example of this failure is the development and deployment of Healthcare.org, where the team, admittedly, did not fully test the online health insurance marketplace until two weeks before it opened to the public on October 1, 2013. Then the site ran into major failures.[8]

Chooser ≠ target user. The second case is neither new nor unique to agile. The term "customer" conflates the chooser with the user. Let's unpack these words:

- A customer is often an organization (the target buyer of enterprise software, that is, product chooser) as represented by the purchasing officer, an executive or committee that makes a buying decision.
- A customer is the target user only for consumer-facing products. For enterprise software, target users may be far from the process of choosing a product, and have no input about products the organization selects.

Agile terminology adds to the confusion: product teams write *user stories* from the perspective of the person who uses the software, not the one who chooses it. Then a *customer demo* (or stakeholder review) at the end of an iteration confirms that each *user story* is satisfied. Here is when the terms customer and user are conflated. For enterprise software and large systems, practice teaches us that often the "end-of-iteration customer" is someone representing the product chooser rather than the end user.

Agile teams without user research are prone to building the wrong product.

So the end-of-iteration demo cannot be the sole form of feedback to predict user adoption and satisfaction. In addition, the software development team should also leverage user research to answer questions such as:

- What are the classes of users (personas)?
- Have we validated that the intended users have the needs specified in the user stories?
- What are the current user practices before the introduction of the product and the impact afterward?
- How would we extend the tool to support new personas or future use cases?

Internal proxies ≠ target user. The third case is about bias. Some teams work with their in-house professional services or sales support staff (that is, experts thought to represent large groups of customers) as proxies for end users. While we appreciate the expertise and knowledge these resources bring, we are wary of two common types of misrepresentation in these situations.

First, internal proxies are unrepresentative as end users because they have multiple unfair advantages: they know the software inside out, including the work-arounds; have access to internal tools unavailable to external customers; and do not need to use the product within the target users' time constraints or digital environment.

Second, the evidence internal proxies bring to the team is also biased. Professional sales and support staff are more likely to channel the needs of the largest or most strategic existing customers in the marketplace. They are more likely to focus on pain points of existing customers and less on what works well. Also, they may ignore new requirements that are not yet addressed by the current tool or market.

Therefore internal staff cannot be the sole representative of "users"—as shown in the "Dilbert" comic strip at the beginning of this column. User research welcomes their comments about competitive analysis, current insights about information architecture or other issues, which complement customer support data, UX research, and other sources of user feedback.

Executives liking sales demos ≠ target users adopting product. Enterprise software companies, during their annual customer conferences, use a sales demo to portray features and functions intended to excite the audience of buyers, investors, and the market analysts about the company strategy. However, positive responses to the sales demos should not be taken as equivalent to assertions about a product's user requirements. Instead, these requirements need confirmation via a careful validation cycle. Let sales demos open a door toward users with the help of choosers and influencers.

Similarly, Customer Advisory Boards (which draw from customers who have large installations, or who represent a specific or important segment of the market) stand in for all customers and offer additional opportunities to showcase future features or strategy. However, a basic law for success in the software industry is "Build Once, Sell Many."[7] This principle creates an inherent tension between satisfying current customers and attracting new ones. Therefore, a software company needs to constantly rethink their tiered offerings to include new market segments or customer classes as these emerge, and avoid one-off development efforts.

Confusing business leaders with users or the sales demo with the product prototype leads companies to build products based on what sales and product managers believe is awesome (for example, see Loranger[6]). Instead, we advocate validating the designs with actual end users during the product development.

Big data (What? When?) < The full picture (... How? Why?). Collecting and analyzing big data about digital product use is popular among product managers and even software developers, who can now learn what features get traction with users. We support the use of big data techniques *as part of* user research and user-centered design, but not as a substitute for qualitative user research. Let's review two familiar ways to use big data on usage: user data analytics and A/B testing.

User data analytics can quickly answer questions about current usage: quantity and most frequent patterns, such as How many? How often? When? Where? Once a product team has worked out most of the design (interaction patterns, page layouts, and more), A/B testing compares design alternatives, such as "which image on a page produces more click-throughs"? In vivo experiments with sufficient traffic can generate large amount of useful data. Thus, A/B testing is very helpful for small incremental adjustments.

Every software company is in the business of finding and keeping new customers. Suppose the logs show the subscribers of an online dating application are not renewing. Should the company rejoice or despair? If people are getting good matches, and thus are satisfied, non-renewal implies success. If they are hopelessly disappointed by not getting dates, non-renewal implies failure. Big data won't tell you which, but observing and listening to even a handful of non-renewing individuals will.

In brief, quantitative data is useful but has two limitations: First, it will not tell the team why the current features are or are not used.[5] Different classes of users can have different reasons. Second, it will not identify what additional or alternative features appeal to a new class of users unfamiliar with the product. To answer these questions the team needs to rely on qualitative research with existing and proposed classes of users.

Market Research ≠ User Research

Finally, we point to the growing and worrisome tendency in industry to mix up user research with market research.

Market research groups make great partners for user research. While user research and market research have a few techniques in common (for example, surveys and focus groups), the goals and variables they focus on are different.

▸ Market research seeks to understand attitudes toward products, categories, or brands, and tries to predict the *likelihood of purchase*, engagement, or subscription.

▸ User research aims at improving the user experience by understanding the relation between actual usage behaviors and the properties of the design. To this end, it measures the behavior and attitudes of users thereby learning whether the product (or service) is usable, useful and delightful, *including after decision to purchase*.

We urge organizations to act strategically and connect market research, user research, and customer success functions. This requires aligning goals and sharing data among Marketing, Sales, Customer Success, and the UX Team (typically in Product or R&D).[1,4]

The Way Forward: Educate Managers and Agile Development Teams

We have shown five different ways that agile teams without user research are prone to building the wrong product. To avoid such failures, we invite software managers and product teams to assess and fill the current gap in a team's competencies. The closing table gives short-term and longer-term action items to address the gaps.

Actions to address gaps in UX competencies.

Short term

1. **Analyze the current skills of the team and flag the gap.** A functional product team needs several key skill sets or UX competencies: UX research, UX design, UI software development and prototyping.[11] These might be filled by training the current team members or hiring UX professionals full-time or part-time.
2. **Support product managers (or product owners) with investment in UX.** Too often, product managers find their role is a sort of "kitchen sink" for any task that is not software development. We encourage product managers to find additional resources in the UX competencies, to benefit both product and their workload.

Longer term

3. **Integrate UX competencies**
   a. Teams need UX research competencies as well as UX design skills (interaction, visual). Other related skill sets include content development and documentation; accessibility; globalization and localization.
4. **Collect and prioritize findings from user research**
   a. Seek user feedback early and often.
   b. Create channels to learn from end users and appropriate surrogates.
   c. Prioritize UX issues during backlog grooming; remove friction and measure delight.
   d. Build new features only after steps 4.a.–c. are done for each key version of the product.

References

1. Buley, L. The modern UX organization. *Forrester Report.* (2016); https://vimeo.com/121037431
2. Grudin J. *From Tool to Partner: The Evolution of Human-Computer Interaction.* Morgan & Claypool, 2017.
3. HP report. *Agile Is the New Normal: Adopting Agile Project Management.* 4AA5-7619ENW, May 2015.
4. Kell, E. Interview by Steve Portigal. *Portigal* blog. Podcast and transcript. (Mar. 1, 2016); http://www.portigal.com/podcast/10-elizabeth-kell-of-comcast/
5. Klein, L. *UX for Lean Startups: Faster, Smarter User Experience Research and Design.* O'Reilly, 2013.
6. Loranger, H. UX Without User Research Is Not UX. (Aug. 10, 2014) *Nielsen Norman Group* blog. http://www.nngroup.com/articles/ux-without-user-research/
7. Mironov, R. *Four Laws Of Software Economics.* Part 2: Law of Build Once, Sell Many. (Sept. 14, 2015); http://www.mironov.com/4law2/
8. Pear, R. Contractors Describe Limited Testing of Insurance Web Site. *New York Times* (Oct. 24, 2013); http://nyti.ms/292NryG
9. Perez, S. Users have low tolerance for buggy apps. *Techcrunch.* (Mar 12, 2013);[http://tcrn.ch/Y80ctA
10. Rosenberg, D. Introducing the business of UX. *Interactions.* Forums. XXI.1 Jan.–Feb. 2014.
11. Spool, J.M. Assessing your team's UX skills. *UIE.* (Dec. 10, 2007); https://www.uie.com/articles/assessing_ux_teams/

Gregorio Convertino (gconvertino@informatica.com) is a UX manager and principal user researcher at Informatica LLC.

Nancy Frishberg (nancyf@acm.org) is a UX researcher and strategist, in private practice, and a 25+-year member of the local SIGCHI Chapter BayCHI.org.