Industry Is Changing, And So Must We

Abstract
This case study is a call to action for research practitioners and academicians to revamp their skills and curriculum respectively. Failure to evolve will likely marginalize the user research discipline in industry.

Author Keywords
Data Science; Large Software Organizations; Usability; Academia.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Ship or sink, not publish or perish, is the mantra for user researchers in large software organizations. Above all, user researchers in industry are accountable for delivering products that meet the needs of customers and delight customers.

User researchers in software organizations are beginning to fail because their impact on products is diminishing. This is because of two trends: (1) software organizations can now execute basic user research, such as usability testing, without the need for user researchers; and (2) software organizations can now
explore user behavior and preferences at scale with high confidence using flighting or A/B testing.

The user research discipline is at an inflection point. This case study presents three recommendations on how user research as a discipline can evolve, with implications for practice and academia. These implications will not only avoid the marginalization of the user research discipline but also evolve the discipline to achieve a new value proposition in the fast-changing world of software development in large organizations.

**Scope**

The audience of this case study is: (1) practitioners in user research; (2) academicians teaching to or doing research with students being groomed for a user research practitioner role.

This case study is not aimed at designers. While designers often get clustered with researchers in both academia and industry, this case study is specifically aimed at practitioner researchers.

This case study is not aimed at software organizations that are not "large". While small-to-medium organizations may also find the content of this case study useful, the recommendations and implications have been purely derived from observing practice in large organizations that are relative mature in software development. Examples of large software organizations include Google and Microsoft. Note that the term "software" is intended to be used generically, as it does not preclude large organizations that also build platforms and hardware.

**A brief history**

This section succinctly captures the long tradition and history of HCI practice (see [1, 4, 11] for details). Notably, the following periods are highlighted that motivate the thesis of this case study.

**User acceptance testing**

HCI practice has its roots well before usability engineering played a formal role in software organizations. For example, in the 1970s, it was common for software engineers to schedule controlled user acceptance tests (UAT) of nearly final system products. Over the years, industry quickly realized that UAT was not effective, as the impact of such testing was limited to largely cosmetic refinements. This realization led to the need for identifying usability issues at earlier points in the software development lifecycle (SDLC).

**Usability engineering**

As industry became more competitive and product cycles became shorter, large software organizations started forming dedicated usability engineering teams. For example, in the 1980-1990s, the profession of usability engineers started to become a formal and recognized career path. Usability engineers came to play major roles in upstream and downstream stages of the SDLC. However, with the booming success of software through the 1990s, there were also countless examples of failed software (e.g. groupware [7]) and the realization that it's not usability alone that matters but also the broader, end-to-end user experience motivated by fundamental human needs and dynamics.
User experience research
Software development pivoted toward scenario-based engineering, of which usability engineering responded by expanding its scope to user experience research. The word “experience” was key, paying a nod to scenarios and experiences as concrete stories of use for end users. For example, in the 2000s, user experience researchers were instrumental in bringing about an organizational change in how software development teams thought about end users not just from a use case perspective (exhaustive state machines for user paths) but from an experiential perspective (key user needs and outcomes). Ethnography became mainstream in practice, as did other social and behavioral sciences that focused on human attitudes, motivations, and habits. It is noteworthy that in 2012, the Usability Professionals Association (UPA) formally changed their name to User Experience Professionals Association (UXPA) [14].

Inflection point
The brief history above sheds light on the different periods of HCI practice. The discipline is, once again, at an inflection point as argued by this case study. The genesis for the inflection point is because of the following two trends.

1. Organizations can now execute basic research
To the credit of HCI practice, the user research discipline is now, for the most part, universally known by large software organizations. Gone are the days where a product or development manager will exclaim “what is usability testing” and instead will demand “I need the usability results for my feature area next week”.

Looking at the user research discipline from the lens of a capability maturity model [3], large software organizations are exhibiting a Level 4 process maturity. Software development teams understand usability testing to the extent that they can self-establish and have self-established process capability in this area.

The need for self-establishing process capability in usability testing is obvious. While user research as a discipline is universally known, large software organizations are primarily engineering organizations, and hence, the ratio of researchers to developers is miniscule [6]. As a result, user researchers are unable to provide coverage to all software development teams.

Thus, it is now becoming commonplace for software development teams to budget money specifically for usability testing. For example, it is not atypical for a software development team to execute usability testing on their own – primarily outsourcing the work to third party vendors – using their own budget. This process allows software development teams to “own” basic usability testing. A by-product of this behavior has also resulted in a recent boom for third party vendors who can execute usability testing quickly on a contractual basis. A great example of such a vendor is UserTesting.com [13].

2. Organizations can now explore user behavior at scale
The advent of the Internet has transformed software from a physical good to an online service. An overwhelming majority of customers do not stand in line to physically get their hands on the next version of the Windows operating system. Instead, software is now distributed online as a service, more commonly known as Software-As-A-Service (SaaS).
SaaS has enabled software organizations to explore and understand user behavior and preferences at scale using A/B testing [12]. While comparing A/B testing to usability testing is a matter of apples versus oranges, it is still noteworthy to mention that from a software development team’s point of view, A/B testing in a lot of ways does in fact replace usability testing. This is because A/B testing directly links software permutations to business decisions (e.g. click conversion rate), operates at scale, and can be run in a fraction of the time it takes to run a usability test.

Intersection of the two trends
The fact that organizations can now execute basic research and explore user behavior at scale implies the need for user researchers to re-examine their value proposition. This case study is not arguing for or against software development teams owning and executing usability testing: in fact, there are potential downsides of doing this. Similarly, A/B testing is not the panacea for understanding user behavior, as argued by Grudin [8].

The argument of this case study is simply the need for user researchers to evolve beyond traditional usability testing as a key value proposition of their work and discipline. There are signs of such evolution occurring, but it is not yet the norm.

Evolution
This section provides three examples of HCI practice by researchers in industry that have been the cornerstone of impactful work over the past couple of decades. These practices are legitimate, i.e. they provide immense disciplinary value to the products and organization. However, that value is diminishing and will continue to diminish over time. Hence, each example of today’s practice is associated with an evolutionary recommendation of how the practice can morph in order to maintain and increase the basic value proposition.

1. Act as user experience gatekeepers
During the historical periods of usability engineering and user experience research, practitioner researchers were more often than not gatekeepers of the user experience. In other words, researchers had the accountability of a formal sign-off on whether or not a product was good enough to ship to customers.

Today, because of the two trends, each and every one touching the product is a user experience gatekeeper. This includes developers, marketers, designers, and customer service & support professionals (after the product has shipped). Practitioner researchers seeking to act as sole user experience gatekeepers will only risk their own marginalization in the organization. In the new age of A/B testing and fierce competitive pressure in terms of quick software releases, large software organizations do not need an excuse to stop a product from shipping, rather a reason always exists to ship a product. A/B testing has created the test bed for live user testing on production code, and in fact, end users now expect to be key stakeholders themselves in early releases such as beta.

Evolutionary recommendation
Instead of gatekeepers, practitioner researchers need to adapt their methods to meet the high velocity of software development through A/B testing. Researchers should identify opportunities worthy of A/B testing,
expand their knowledge set to weigh in on what makes a good or bad A/B test, and focus on the tradeoffs between immediate business metrics and user experience in the long run. Essentially, practitioner researchers should be process enablers rather than ship blockers.

Additionally, practitioner researchers should expand on end users as key stakeholders. The technology (e.g. devices, software) that large software organizations build today is ever so critical to the everyday lives of end users. To ensure technology meets every day needs, organizations must involve the users more in how the technology is built. Activities like crowdsourcing feedback from end users is becoming commonplace to affect how technology is built, though such a pursuit has its own tradeoffs which is beyond the scope of this case study.

2. Validate usability of design
The history and making of modern day practitioner researchers is rooted in usability engineering. As Grudin [8] argues articulately, it can now take minutes to validate the usability of a particular design through A/B testing versus weeks through a traditional usability study.

In many cases, validating the usability of a design may make sense in a lab setting, despite the availability of A/B testing. For example, gaining insight into human behavior (the "why" behind user behavior) through a usability study is valuable and cannot be deeply understood through A/B testing alone. At the same time, in many cases, a usability study can be a redundant, if not a resource intensive, activity that provides little or no return on investment. For example, trying to understand which design performs better is much better suited for A/B testing than a usability study.

The underlying point here is that researchers naturally have an incumbent practice of validating usability of designs through lab testing. Today, practitioner researchers need to be deliberate about when and when not to run a usability study in light of other sources of experimentation and big data.

EVOLUTIONARY RECOMMENDATION
Instead of just validating the usability of designs, practitioner researchers need to concentrate on human insights behind the blanket of quantitative data. Quantitative data can help to unveil areas of exploration where researchers can go deeper and explore the underlying human phenomena behind the numbers. Furthermore, researchers can start to test hypotheses around human behavior at scale. Such interactions allow large software organizations to identify technology opportunities, while maintaining usability evaluation as an additional goal.

3. Execute a usability study
Executing a usability study was the cash cow of practitioner researchers. The empirical evidence around a specific UI or feature is what defined the value of a user researcher.

Today, software engineers can fund their own research studies and execute "common" research on their own (e.g. remote, unmoderated user testing). Thus, practitioner researchers need to re-consider their own role in light of this transition. By simply stating that only researchers can execute a "proper" or "rigorous" study or experiment will once again only contribute to
the discipline’s marginalization, as execution of a study is a skill that can be acquired and practiced by anyone in short order, if not outsourced.

The prior point emphasized the need to focus on the qualitative, human insights behind quantitative data. This point emphasizes that practitioner researchers can add more value to the organization by taking a longitudinal view on the knowledge produced by running studies over time.

EVOLUTIONARY RECOMMENDATION
Instead of just executing individual, often disparate studies, practitioner researchers need to take a page out of academia and become synthesis experts of human behavior. Academicians, using empirical evidence, ultimately strive to create models, theories, and frameworks based on a large body of empirical evidence, some of which they have undertaken but a lot of it that has been undertaken by other academicians.

In the same spirit, practitioner researchers should contribute to the organizational knowledge of human insights over time. While individual studies provide immediate, tactical benefits, the unique value that researchers can provide is to consolidate the body of human knowledge intelligently and constantly update this body over time beyond one product. The ability to create and expand on models, theories, and frameworks is a unique skillset that user researchers possess over other disciplinary roles in large software organizations.

Implications
The industry is changing, and so must we. The following themes provide two implications as action science.

From an academic perspective, course curriculum and training has to evolve to reflect the skills required in today’s software industry. For example, academic institutions should explore the offering of “data science” as a mandatory HCI course that focuses on the science of A/B testing, the intersection of A/B testing and behavioral sciences, and most importantly, the interpretation of the data as a result of A/B testing. Curriculum and training should also explore the emphasis of business communication and its impact on business decisions, not just from a user experience perspective but also from a financial perspective.

From a practice perspective, the previous section touches on how researchers need to evolve in order to succeed as practitioners in large software organizations. Practitioner researchers need to brush up their skills, both technical and non-technical. Technically, researchers must explore how their expertise in human behavior and insights can add a multiplier effect to the two trends. Non-technically, researchers must expand their scope around soft skills and business impact in order to persuade software engineers to make more user-centered decisions based on the science of human behavior and insights.

Conclusion and future work
This case study is a call to action for academia and practice. While the findings presented here are anecdotal in nature, they offer potential for real impact on HCI practice.
The three recommendations and two implications speak to the urgency for HCI to focus on how practitioner researchers will fare with fast-changing software development practices of large organizations. It is the authors’ hope that by seeding such a conversation through this case study, HCI academicians and practitioners can advance the state of practice. To that end, a plausible next step is for HCI practitioners to refine and add to the implications presented in this case study, and for HCI academicians to engage more deeply with practitioners for improving course curricula and training. Indeed, the time is right to rekindle prior conversations (e.g. [2, 5, 10]) around what today’s industry needs from HCI practitioners and what academia is delivering.

Acknowledgements
The content of this case study is solely the opinion of the authors, and not the authors’ organization. The joint talk with Jonathan Grudin [9] motivated the context for this case study. We would like to thank the Operating Systems Group, in particular the User Research team and its leadership team. In addition to the reviewers, we would like to thank Jonathan Grudin, Eli Goldberg, and Paula Bach for helping to refine the case study.

References