6. Iteration

Emma J. Rose
University of Washington Tacoma

Cody Reimer
University of Wisconsin–Stout

Definition and Background

Iteration refers to the stage in the design thinking process of making changes to a system, whether it is a product or service, in order to make improvements. To successfully design a usable system, the design process must include three key principles: an early focus on users and tasks, empirical measurement, and iterative design (Gould & Lewis, 1985). Iteration is a key stage in the design process where new features are added and existing problems are corrected, resulting in a new version. It is a series of incremental improvements in a development lifecycle accomplished through ***ideating***, modeling, ***testing***, revising, and then repeating that process as necessary. As key as iteration is to the design thinking process, in TPC iteration can take many forms, especially within the drafting and writing process. Whether it is making incremental changes to an existing draft or starting from scratch, iteration is important for both writers and designers.

Design is often an interplay between ***ideation*** and iteration (Greever, 2015). As Jennifer Sano-Franchini demonstrated in her entry, ideation is the process of coming up with and elaborating as many new design ideas as quickly as possible, while iteration is the process of getting feedback on designs and making changes in order to improve them. Designers start with many ideas and concepts represented by low-fidelity prototypes such as sketches, which are then winnowed down over time as they are iterated and changed and move into a higher level of fidelity which moves towards the final version of a system. Bill Buxton (2007) referred to this process of moving from many ideas to one as the design funnel.

Iteration is both a process and a product. The process of iteration is embodied in the design mindset that feedback and input will improve a product. Feedback may come from peers during review or stakeholders within an organization, but most often, and ideally, it comes from representative users who are providing input via applied research methods such as ***usability*** testing. In terms of iteration as a product, a new version of a design or changes to design is referred to as "an iteration," which represents a snapshot of a design at a single point in time. Iterations help reduce the redundancy created by different approaches, such as parallel design, wherein different designs are built simultaneously.

DOI: https://doi.org/10.37514/TPC-B.2022.1725.2.06

Design Application

Working iteratively is a key mindset for designers and design thinkers. An important part of design is expecting to get feedback, input, and data that will shape and change a design. ***Prototyping*** and creating early versions of systems helps encourage iteration rather than waiting until a system is complete to share it with others. The speed of iteration is dependent on organizational context, but with the increase in popularity of approaches like agile software development and lean UX, being able to iterate rapidly is an important skill (Gothelf & Seiden, 2013). Proponents of lean UX point to rapid iteration as a means to stay competitive with cost and time (Aarlien & Colomo-Palacios, 2020). In their systematic review of literature on lean UX, David Aarlien and Ricardo Colomo-Palacios (2020) list its key principles: early validation, cross-functional design, solving user problems, measuring performance indicators, and applying tools flexibly (p. 5).

As designers iterate through models, they can evaluate the user experience of each model to aid in subsequent revisions. Low-fidelity models are cheap but abstracted, while higher fidelity comes with higher confidence in the user experience data but also a higher resource cost. According to Jakob Nielsen (2011), the recommended number of iterations for a design is at least two (meaning at least three versions), but the more the better, he suggests, though he also notes the existence of diminishing returns. The limit to the number of iterations is often time, as iterations are cheap compared to alternatives such as parallel design.

The video game industry has benefited from shrinking the time between iterations. As business models shift to longer life cycles for games, releasing new content for existing software generates significant revenue. That content must be tested within the contexts of the existing game, and the content is on a brisk release schedule. Many video game companies iterate with the speed low fidelity affords while gaining the confidence high fidelity permits. This is accomplished by:

- tracking user data and, in some cases, releasing that data back to users for them to pore over so as to better enable what in gaming argot is called "theorycrafting" (Paul, 2011),
- employing proxy servers as "test realms" where they can test code without disrupting service,
- stress-testing changes by controlling player access to those test realms, and
- supporting quantitative analysis with qualitative inquiry.

Riot Games rapidly iterates in their video game *League of Legends*, which uses a player test realm to implement, test, and adjust changes before monthly patches to the live servers. As Cody Reimer (2017) shows, Riot balances quantitative and qualitative feedback to ensure the constant and numerous changes improve user experience. They manage this by releasing typically proprietary player data to players in order to fuel discussion, and then engaging in that discussion on both official and unofficial forums.

Player-developer discussion often focuses on changes being made to the game, such as adjustments for balance or additions of new characters. Lee Sherlock (2016) discusses Riot's technical documentation, specifically patch notes, itemizing changes for each iteration of *League of Legends*. Framing a changelog genre, something often written for experts, for a player audience means communicating both technical coding changes (what was changed and how) as well as design philosophy (why it was changed in this way). Riot's patch notes exemplify one prominent way iteration appears in TPC.

Pedagogical Integration

When teaching design, building in opportunities for iteration is key. As is the case with many creative processes, iterative design varies in execution. Different disciplines approach it differently: Modeling (sketches, paper prototypes, wireframes, etc.), testing (usability in its myriad forms), and revising are adapted to the contexts and needs of the specific discipline, practice, or strategy employing it.

For students new to the concept of iteration, instructors may compare iteration to the writing process and how each draft and revision is an iteration with opportunities for feedback. Students might benefit from thinking of sketches like reading notes, wireframes like rough drafts, and the finished design like a polished deliverable. Another way to introduce students to iteration is to have them explore platforms such as Wikipedia, Google Docs, and GitHub, or iterative project management models like Agile and Scrum. Ask students to examine or trace how specific iterations in these models evolve over time and how feedback is key to shaping each iteration.

In order for students to experience iteration as product and process, it should be built into the structure of the course. Make each iteration a discrete stage and build assignments around each deliverable. Take, for example, a project where teams of students are designing a website for a community partner. Build in three iterations: early sketches, prototype, and a functional site. At each iteration, students should engage in rounds of critique and, if possible, usability testing, to further explore how the design is refined over time and how each level of fidelity is an opportunity to explore and improve design elements based on feedback and data. Using community- or client-based projects is a valuable way to teach iteration since each design iteration can be shared with stakeholders for review and critique. Early client concerns can be addressed in early, low-fidelity iterations before too much time has been spent. Further, the choice of fidelity can highlight the rhetorical nature of iterative design (Rose & Tenenberg, 2017) as different stakeholders might not understand or appreciate lower fidelity "unfinished" work.

Critique and revision are key components of iteration. Students should have the opportunity to engage in daily or weekly design critiques or reviews. One helpful mindset to teach to students is to embrace feedback with patience and view critiques as opportunities to improve their design. It can also be helpful to

generate a set of questions that is used consistently in regular reviews, such as: "What have we done so far?" "What works in the current design?" "What still needs to be done and how might we do that?" Keeping a constant approach to reviewing design can reduce redundancy and create common expectations that can aid in the iteration process.

References and Recommended Readings

Aarlien, D., & Colomo-Palacios, R. (2020, July). Lean UX: A systematic literature review. In *International Conference on Computational Science and Its Applications* (pp. 500-510). Springer, Cham.

Buxton, B. (2007). *Sketching user experiences: Getting the design right and the right design*. Morgan Kaufmann.

Da Silva, T. S., Martin, A., Maurer, F., & Silveira, M. (2011, August). User-centered design and agile methods: A systematic review. In *2011 AGILE Conference* (pp. 77-86). IEEE.

Gothelf, J., & Seiden, J. (2013). *Lean UX: Applying Lean principles to improve user experience*. O'Reilly Media.

Gould, J. D., & Lewis, C. (1985, March). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Greever, T. (2015). *Articulating design decisions: Communicate with stakeholders, keep your sanity, and deliver the best user experience*. O'Reilly Media.

Karat, J., & Karat, C. M. (2003). The evolution of user-centered focus in the human-computer interaction field. *IBM Systems Journal*, *42*(4), 532-541.

Lepore, T. (2010, May). Sketches and wireframes and prototypes! Oh my! Creating your own magical wizard experience. *UXmatters*. https://www.uxmatters.com/mt/archives/2010/05/sketches-and-wireframes-and-prototypes-oh-my-creating-your-own-magical-wizard-experience.php

Nielsen, J. (2011, January). *Parallel & iterative design + competitive testing = high usability*. Nielsen Norman Group. https://www.nngroup.com/articles/parallel-and-iterative-design/

Paul, C. A. (2011). Optimizing play: How theorycrafting changes gameplay & design. *Game Studies*, *11*(2). http://gamestudies.org/1102/articles/paul

Reimer, C. (2017). Dialogic, data-driven design: UX and League of Legends. In L. Potts & M. Salvo (Eds.), *Rhetoric and experience architecture* (pp. 241-257). Parlor Press.

Rose, E., & Tenenberg, J. (2017). Making practice-level struggles visible: Researching UX practice to inform pedagogy. *Communication Design Quarterly*, *5*(1), 89-97.

Sherlock, L. (2016). Patching as design rhetoric: Tracing the framing and delivery of iterative content documentation in online games. In J. deWinter & R. M. Moeller (Eds.), *Computer games and technical communication* (pp. 157-170). Routledge.

Spinuzzi, C. (2005). The methodology of participatory design. *Technical Communication*, *52*(2), 163-174.

Sullivan, P. (1989). Beyond a narrow conception of usability testing. *IEEE Transactions on Professional Communication*, *32*(4), 254-264.

Taylor, T. L. (2006). Beyond management: Considering participatory design and governance in player culture. *First Monday*. http://firstmonday.org/ojs/index.php/fm/article/view/1611/1526