

In “Designers – think big!”, Tim Brown challenges us all to reconsider design. In particular, he highlights his learning path, recognizing that he was “being incremental and not having much of an impact.” To overcome this shortcoming, he suggests “...if we take a different view of design, and focus less on the object and more on design thinking as an approach, that we actually might see the result in a bigger impact” (Brown, 2009). Adding just a single word, “thinking”, is an important step in reminding us all how to succeed. Perhaps the problem with using “design” rather than “design thinking” is that we think of design as a single step / single flow, one way process?

In software development, a legacy design/development approach that has time and again been proven to fall short is known as “waterfall” (Royce, 1987). In this model, the creation of a new product moves through a sequential series of steps: analysis, design, construction, testing and support (or similarly named steps). The output of one step flows as input into the next step. There are quality gates, change control mechanisms and typically many detailed and heavy-weight documents. In following the approach, there is little opportunity get input and feedback, significant resistance to change, a disconnect from users, and the expectation that those creating the product will “follow the process.” The result is that it is very risky, often leads to failure and at its core has eliminated “thinking.”

Turning to examples of including thinking, we can look to students from a Theory of Knowledge class as they attempt to create a model of knowledge as described by Overnell-Carter (2014). A model offers little value, until we test it out and find out where and how it is valid. As such, the students test the model and think about how their design is correct, where it is incorrect and how it might be improved:

“We have some refining to do, but we think our first real world test of our model does helps us appreciate the complex relationship between the elements of Theory of Knowledge and avoid the trap of thinking we can really talk about areas of knowledge and ways of knowing in isolation. Our next test will be to observe and map a different kind of class, a history or literature class, for example, to see if knowledge works differently in the humanities and to further test our map” (Overnell-Carter, 2014).

Similarly, we can also use prototypes to interact with our users. And we can do so in an “...open, collaborative and participative approach” (Brown, 2009). The use of prototypes, collaborating with others and seeking feedback early and often can be used in software, accessing water, creating health related products, or even designing video games.

“Another key component to playcentric design is that ideas should be prototyped and playtested early. Immediately after brainstorming ideas, we encourage designers to construct a playable version of their idea. By this we mean a physical prototype of the core game mechanics. A physical prototype can use paper and pen, index cards, or even be acted out.... This way, the game designer receives instant feedback on what players think of the game and can see immediately if they are achieving their player experience

goals.... This might sound like common sense but in the industry today, much of the testing of the core game mechanics comes later in the production cycle, which can result in disappointment” (Fullerton, 2008, p. 11)

In wrapping up, Brown recommends that “...the first step is to start asking the right questions” (Brown, 2009). We need to be creative, thoughtful and iterative in coming up with these questions. And, of course, we need to be thoughtful in how we seek out the answers, analyze the impact of those answers and adapt our approach. In summary, this iterative, feedback seeking, and inspect & adapt approach requires us to think...early and often!

## References

Brown, T. (2018). Designers--think big! Lecture presented at TED Talk in UK, Oxford. Retrieved January 16, 2018, from [https://www.ted.com/talks/tim\\_brown\\_urges\\_designers\\_to\\_think\\_big](https://www.ted.com/talks/tim_brown_urges_designers_to_think_big)

Fullerton, T. (2008). Game design workshop: a playcentric approach to creating innovative games.

Ovenell-Carter, B. (2014, April 3). #tok students map knowledge with tools they created themselves #tokrew11 @sharpcomposer [Web log post]. Retrieved January 16, 2018, from <http://www.ovenell-carter.com/field-school/tok-students-map-knowledge-with-tools-they-created-themselves-tokrew11-sharpcomposer/>

Royce, W. W. (1987, March). Managing the development of large software systems: concepts and techniques. In Proceedings of the 9th international conference on Software Engineering (pp. 328-338). IEEE Computer Society Press.