

Learning Rapid Implementation of AI and Data

by Ikhlaq | posted in: artificial intelligence, Berkeley method of entrepreneurship, Berkeley-X Viewpoint, Blockchain, data, entrepreneurship, Sutardja Center News |

[Ikhlaq](#)



What's Golf Got to Do With It?

Our ability to deliver on the promises of AI, data, and other digital transformations is key to our national agenda and global security. Just imagine what would happen if another, less friendly nation educates more people to readily develop these technologies. It's pretty much the same as losing the industrial revolution and becoming irrelevant, both from an economic and national security point of view.

So how do we currently teach keen people to climb the steep learning curve needed to develop these technologies? Today, we basically teach the theoretical foundations for each subject and then hope for the best.

If we taught people to play golf the same way that we teach people to master complex topics at universities, then it would be a miracle if we actually developed good golf players. In our hypothetical golf curriculum, we could expect that in year one, students would only be allowed to place the ball on the ground, and they would practice this for about one year. What about hitting the ball? No, that will happen much later. In the second year, we would focus completely on the backswing. How about hitting the ball in year three? No, the third year will be focused on understanding the materials inside the ball, and maybe calculate the optimal number of dimples for best flight. Finally, after another year of golf theory, everyone graduates. Everyone is supposedly an expert at Golf. The only problem is that no one in the class has ever played it or even hit the ball. But of course, that can be learned on the job!

In Our Model, You Start By Hitting the Ball First

This is what we are trying to fix with AI, ML, and blockchain in our Rapid AI/Data-X courses. A student who takes a different approach to learning these subjects may learn many theories without understanding how to actually implement an AI or data science solution.

So here is an important idea for experiential learning; Would it not be better to start by letting students just hit the ball, and then find out what they need to improve? Some coaching along the way would also have a very positive effect.

This is the method we decided to try for AI, ML, and other data implementations. We would teach students how to use the fundamental tools at a simple level, and then see what they can do and what they need to learn to do next. And while coaching is part of the process, a significant part of learning is self-driven with the goal of making a challenging project work by

the end of a short period of time.

We tried this rapid type of implementation in our Data-X class at Berkeley. And even to our surprise, we found that our students actually learned to create amazing projects in just a few weeks, not 2 years! We saw projects that identified knee problems from an MRI/CAT scan, that predicted energy use from past data, and that can identify fake news. With almost 150 students in the class, we are seeing over 25 projects in a semester.

Now don't get me wrong, this model is not at all against theory. It is absolutely essential to focus on the theoretical frameworks at every world-class university. However, in this case, we are working to fill an implementation gap. Theory is still absolutely necessary, but it has to be the right theory, i.e. relevant and connected to the implementation and tools. In fact, the course we developed includes a different balance of a) relevant theory, b) understanding of tools, and c) an open-ended project that serves as a defining challenge.

Our learning lesson: It seems to work!

Let me share an example of a comment I saw from a student who took the class:

“I think this class is so awesome because it teaches the tools and concepts that are most commonly used in workplace teams that are involved with data science and applied machine learning. The vast majority of teams that I've applied to within the past year use the tools taught in this class. When I arrived at my data science internship this summer, I already knew how to use most of my team's stack.”

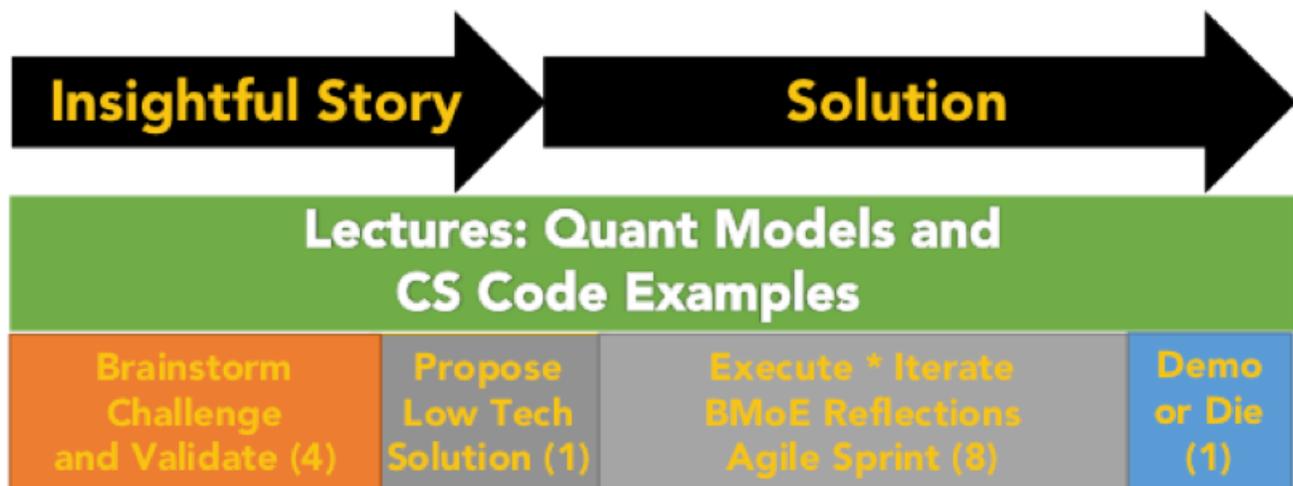
I cannot say that this is particularly good or bad as a first result, but it frankly seems promising to me.

How We Put It All Together:

For our rapid, implementation-oriented method to work, we needed to bring innovation behaviors into the approach. In the first 4-5 weeks of the class, we actually do not ask students to write code for their projects. Instead, we have them develop what we would consider an “insightful story” or “narrative” that describes what they intend to build.

In real life, every successful project, innovation, and/or venture starts with a story narrative. Typically, a story is used to test the concepts as well as develop interest from other stakeholders of a project. By week five, students convert their story into a “low tech demo” which captures many aspects of the project’s implementation, but without code. It is a super-light prototype which can be easily modified until it’s “right” in multiple ways.

This approach is common to design as well as entrepreneurship, and it is also good practice for students to learn this innovation behavior.



After the low-tech demo, students get the green light to go into an agile development cycle and prepare for a demonstration in eight more weeks.

Similar to the MIT Media Lab, the slogan here is not “Publish or Perish,” instead, it is “Demo or Die.”

There is a solid stream of math theory lectures, notebooks with code samples, and explanations of open source computer science tools along the way. These components (project, tools, and theory) when integrated in the right balance result in a very rapid experiential learning curve.

Key concepts: Story first, low tech demo, agile sprint, coaching for innovation behavior, and the three integrated elements of theory, project, and tools.

There is much more nuance than we have been experimenting with. I can say from experience, there are many ways for engineering projects to go wrong and we are correcting for many of these issues in different ways. I'm sharing our experience with this approach because even at this early stage, it does seem to be promising.