What is the real difference between Data Science and Software Engineering Teams?

Josh Poduska May 16, 2019

Although there are lots of similarities across Software Development and Data Science , they also have three main differences: processes, tooling and behavior. Find out.

In my previous article, I talked about model governance and holistic model management. I received great response, along with some questions about the differences between Data Science and Software Development workflow. As a response, this piece highlights the key differences in processes, tools and behavior between Data Science and software engineering teams, as well as best practices we've learned from years of serving successful model driven enterprises.

Why Understanding the Key Differences Between Data Science and Software Development Matters

As Data Science becomes a critical value driver for organizations of all sizes, business leaders who depend on both Data Science and Software Development teams need to know how the two differ and how they should work together. Although there are lots of similarities across Software Development and Data Science , they also have three main differences: processes, tooling and behavior. In practice, IT teams are typically responsible for enabling Data Science teams with infrastructure and tools. Because Data Science looks similar to Software Development (they both involve writing code, right?), many IT leaders with the best intentions approach this problem with misguided assumptions, and ultimately undermine the Data Science teams they are trying to support.

Data Science != Software Engineering

I. Process

Software engineering has well established methodologies for tracking progress such as agile points and burndown charts. Thus, managers can predict and control the process by using clearly defined metrics. Data Science is different as research is more exploratory in nature. Data Science projects have goals such as building a model that predicts something, but like a research process, the desired end state isn't known up front. This means Data Science projects do not progress linearly through a lifecycle. There isn't an agreed upon lifecycle definition for Data Science work and each organization uses its own. It would be hard for a research lab to predict the timing of a breakthrough drug discovery. In the same way, the inherent uncertainty of research makes it hard to track progress and predict the completion of Data Science projects.

The second unique aspect of Data Science work process is the concept of hit rate, which is the percentage of models actually being deployed and used by the business. Models created by Data Scientists are similar to leads in a sales funnel in the sense that only a portion of them will materialize. A team with 100 percent reliability is probably being too conservative and not taking on enough audacious projects. Alternatively, an unreliable team will rarely have meaningful impact from their projects. Even when a model didn't get used by the business, it doesn't mean it's a waste of work or the model is bad. Like a good research team, Data Science teams learn from their mistakes and document insights in searchable knowledge management systems. This is very different from Software Development where the intention is to put all the development to use in specific projects.

The third key difference in the model development process is the level of integration with other parts of the organization. Engineering is usually able to operate somewhat independently from other parts of the

business. Engineering's priorities are certainly aligned with other departments, but they generally don't need to interact with marketing, finance or HR on a daily basis. In fact, the entire discipline of product management exists to help facilitate these conversations and translate needs and requirements. In contrast, a Data Science team is most effective when it works closely with the business units who will use their models or analyses. Thus, Data Science team needs to organize themselves effectively to enable seamless, frequent cross-organization communication to iterate on model effectiveness. For example, to help business stakeholders collaborate on in-flight Data Science projects, it's critical that Data Scientists have easy ways of sharing results with business users.

II. Tools and Infrastructure

There is a tremendous amount of innovation in the Data Science open source ecosystem, including vibrant communities around R and Python, commercial packages like H2O and SAS, and rapidly advancing deep learning tools like TensorFlow that leverage powerful GPUs. Data Scientists should be able to easily test new packages and techniques, without IT bottlenecks or risking destabilizing the systems that their colleagues rely on. They need easy access to different languages so they can choose the right tool for the job. And they shouldn't have to use different environments or silos when they switch languages. Although it is preferable to allow greater tool flexibility at the experimentation stage, once the project goes into deployment stage, higher technical validation bars and joint efforts with IT become key to success.

On the infrastructure front, Data Scientists should be able to access large machines, specialized hardware for running experiments or doing exploratory analysis. They need to be able to easily use burst/elastic compute on demand, with minimal DevOps help. The infrastructure demands of Data Science teams are also very different from those of

engineering teams. For a data scientist, memory and CPU can be a bottleneck on their progress because much of their work involves computationally intensive experiments. For example, it can take 30 minutes to write code for an experiment that would take 8 hours to run on a laptop. Furthermore, compute capacity needs aren't constant over the course of a Data Science project, with burst compute consumption being the norm rather than the exception. Many Data Science techniques utilize large machines by parallelizing work across cores or loading more data into the memory.

III. Behavior

With software, there is a notion of a correct answer and prescribed functionality, which means it's possible to write tests that verify the intended behavior. This doesn't hold for Data Science work, because there is no "right" answer, only better or worse ones. Oftentimes, we'll hear Data Scientists discuss how they are responsible for building a model as a product or making a slew of models that build on each other that impact business strategy. Unlike statistical models which assume that the distribution of data will remain the same, the distribution of data in machine learning are probabilistic, not deterministic. As a result, they drift and need constant feedback from end users. Data Science managers often act as a bridge to the business lines and are focused on the quality and pace of the output. Evaluating the model and detecting distribution drift enables people to identify when to retrain the model. Rather than writing unit tests like software engineers, Data Scientists inspect outputs, then obtain feedback from business stakeholders to gauge the performance of their models. Effective models need to be constantly retrained to stay relevant as opposed to a "set it and forget it" workflow.

Final Thoughts

In general, there are several good practices for Data Scientists to learn

from Software Development , but there are also some key differences to keep top of mind. The rigor and discipline that modern Software Development has created is great and should be emulated where appropriate, but we must also realize that what Data Scientists build is fundamentally different from software engineers. Software Development and Data Science processes often intersect as software captures much of the data used by Data Scientists as well as serving as the “delivery vehicle” for many models. So the two disciplines, while distinct, should work alongside each other to ultimately drive business value. Understanding the fundamental nature of Data Science work can set a solid foundation for companies to build value-added Data Science teams with the support of senior leadership and IT team.