Managing Enterprise Data Strategies Exercise Guide

Jesse Anderson

Contents

1 Use Case Evaluation

1.1 Objective

This 60-minute lab helps you evaluate your use case and team with Big Data technologies. We will:

- Evaluate your use case
- Evaluate the team and company's core competencies
- Decide on technologies to focus your team on

1.2 Use Case

We need to decide if your use case requires Big Data or small data technologies:

- Is it Big Data?
- Will it become Big Data?
- Is a small data technology a better fit?
- Do all nails look like a job for a hammer named Hadoop?

1.3 Core Competencies

Now that you have decided that your use case must be solved with Big Data technology, which one(s) is the right one(s) and why? This portion requires an honest and realistic look at the team's make up and abilities. To determine if your team can execute a Big Data project, ask yourself:

- Does my team have a background in distributed systems?
- Does my team have a background in multi-threading?
- What programming language(s) does my team use?
- What small data technologies is my team already familiar with?
- Some teams have an unrealistic view of their abilities. Does your team suffer from this?
- Do you and the team realize the increase in complexity associated with Big Data?

1.4 Technologies

After your honest and realistic look at the team, you need to decide on technologies. Your technology choices may receive push back from the team. Make this a starting point for discussion with the team.

- What are the technologies that will make the project successful?
- What are the technologies that my team can be successful with?
- Does your team or company suffer from NIH (not invented here)? The team may want to roll their own or create their own distributed system. (Usually not a good idea.)
- Does the team have the training necessary to be successful?

1.5 Recipe For Success

What is your team's probability of success?

These are the questions I ask myself while assessing a team's probability of success. This isn't a true/false dichotomy. It is a true probability.

To estimate the probability, you must have honestly and objectively answered the preceding questions. If you answered:

- For the majority of questions that your team is capable, you should have a high probability of success. The team may only need some help or training in advanced areas or topics.
- That your team is capable for about half the questions, you should have a medium probability of success. The team needs training and/or mentorship to be successful.
- For the majority of questions that your team is **not** capable, you may have a low probability of success. You should really think hard about undertaking the project. This may be a time to get external help. Your team will need training and mentorship. Without doing that, you'll be setting the team up for failure, which isn't fair to you or the team.

The majority of teams that are starting out on their Big Data journey fall into the medium- to low-probabilities. Identify these risks early so that they don't explode later on in the development cycle.

2 Project Evaluation

2.1 Objective

This 60-minute lab helps you evaluate your Big Data project. We will:

- Talk about prototypes
- Consider Cloud versus on-premises clusters
- Evaluate the project or proof of concept
- Iterate on the phase of crawl, walk, run

2.2 Recap

To recap from the previous exercise, you've already:

- Decided your use case needs Big Data
- Taken an honest look at your team and requested any help they might need.
- You've decided on which technologies to use

2.3 Prototyping

We need to decide on a prototype and start coding it.

- Which team members are responsible for the coding?
- Is that team member capable of doing their part in the project?
- What is the expected time frame for finishing the project?
- Does the company management often confuse a prototype with a finished product?
- Is the prototype small enough to be accomplished in a shorter time frame?
- At the same time, is the prototype difficult enough to show business value and the team's abilities?

2.4 Cloud or On-Premises

Using the Cloud is a common practice for creating Big Data POCs (proof of concept). Purchasing large amounts of hardware to facilitate clusters is often a waste of money. Unless you already have computers or clusters for the POC, I highly suggest using a Cloud provider.

Individual developers should use virtual machines for writing the code and doing initial testing. Unit tests should give a preliminary pass or fail of the build.

- Which Cloud provider should you use?
- What are the costs to run the cluster for budgeting purposes?
- If you have to use an on-premises cluster, does the hardware have the specifications to run Hadoop and the other services that are necessary?
- Do the developers have computers sufficiently powerful to run the required VM and software?
- Does the QA team have a cluster to test on?
- Have you made it as easy as possible or self-serve as possible to spin up a cluster?
- Have you automated the deployments of code so that testing isn't as manual?

2.5 Evaluating the Project or POC

It is incredibly important, and often not done, to evaluate the project. In Agile methodology, this would be similar to a retrospective, except that it would be for that part of the project or entire POC. This is an important time when the team can learn from previous issues.

- Did the POC solve the business need?
- Did the POC take the amount of time you thought it would?
- How high of quality is the code that makes up the POC?
- Would you actually put this level of code quality in production?
- Was the POC so simple that gave the CxOs or upper management a false sense of timelines and complexity?

2.6 Iterating on the Next Phase

Projects that have an all-or-nothing approach often fail due to timelines and complexity. It is important to break up the overall design or system into phases. I call this crawl, walk, run. I encourage you to break up your development phases to gradually create the system rather than all at once.

- How can you break up your system into crawl, walk, run?
- Have the rest of the management been informed which parts or features of the system will be in which phase?
- Is the next phase in the project vastly more complex or time consuming than the last phase?
- Do the phases have a logical progression?
- Is one feature dependent on a feature down the line?

2.7 Recipe For Success

Big Data systems are incredibly complex. Part of your job will be to educate your coworkers of this fact. Failing to do so will make everyone compare your progress to the easier projects like a mobile or web project.

Despite your best efforts and planning, Big Data solutions are still hard. When you start falling behind in the project or start hitting a roadblock, I highly suggest getting help quickly. I've helped companies who had been stuck for several months. Had they reached out for help sooner, they could have saved months of time and money. This isn't admitting failure or an issue with the team; Big Data is really that hard.

For more information about these services, go to http://www.smokinghand.com.