RICKER LYMAN
ROBOTIC

pocket guide

BIG DATA

A BASIC GAME PLAN FOR THE FIRST 360 DAYS OF ADOPTING BIG DATA TECHNOLOGY

Ricker.Lyman Robotic

pocket guide

BIG DATA

A BASIC GAME PLAN FOR THE FIRST 360 DAYS OF ADOPTING BIG DATA TECHNOLOGY

Ricker Lyman Robotic

Big Data Pocket Guide by Jeffrey Ricker

Printed in the United States of America.

Graphic design by Bambuk Design Studio
http://bambukstudio.com/

May 2018: First edition

https://rickerlyman.com

content

RESEARCH & DEVELOPMENT

FIRST PRODUCTION

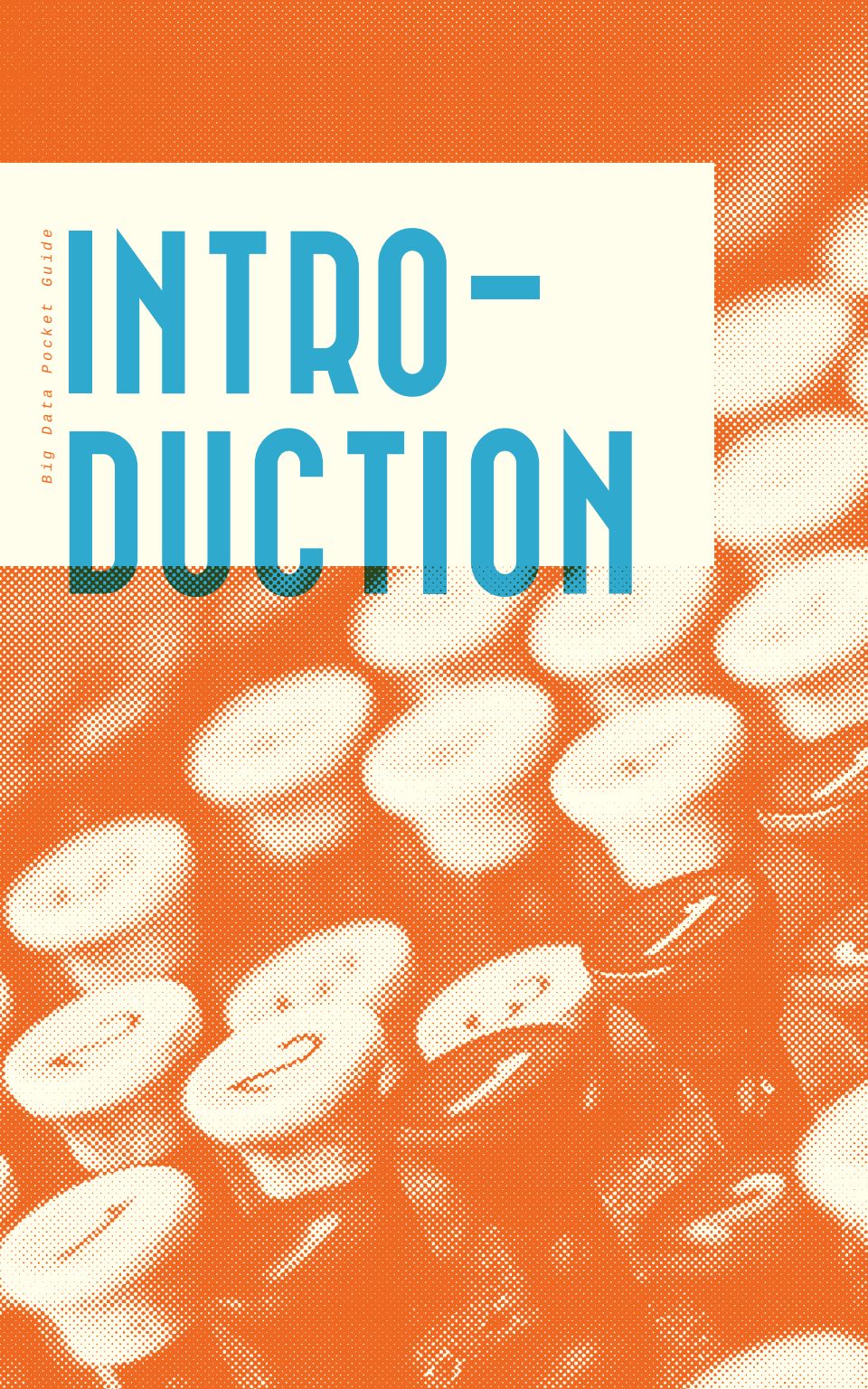
CONCLUSION

“Big data is not about the data”

Gary King,
Harvard
University

Big Data Pocket Guide

INTRO-DUCTION



Despite the press and board room discussion, most companies have not yet adopted big data technology. When considering big data, every business leader has the same basic questions.

basic questions

- *Do I actually have big data?*
- *What strategic advantage will I gain?*
- *Where do I start?*
- *What hardware and software is needed?*
- *What sort of team needs to be assembled?*
- *What are reasonable timelines and objectives?*

Our purpose here is to answer these questions and provide business leaders a basic game plan for adopting big data technology.

WHY BIG DATA

BIG ENOUGH DATA

Big data is any data large enough that it cannot be easily processed or managed on traditional systems. The name “big data” is actually misleading. You do not need petabytes of data to have a big data problem. A better name might be “big enough data.” If your data processing job is taking hours to complete, then you could benefit from big data technology.

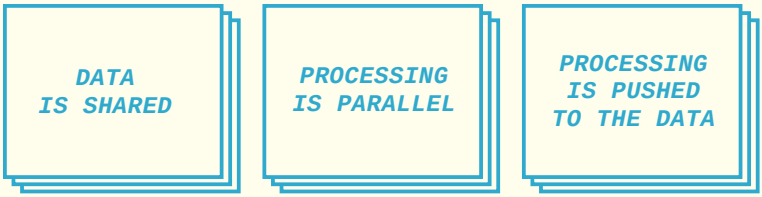
A more accurate name still would be “massively distributed computing.” It’s not very catchy, but that is the essence of big data technology: its power comes from enabling multiple machines to work on the same problem at the same time. For instance, what would take 24 hours to run on one server can be completed in less than an hour on 24 servers.

Perhaps we should just call it *Hadoop*, which has become synonymous with a host of open source projects that encompass most of what we call big data technology.

THREE BASIC CONCEPTS

There are three basic concepts which underlie big data technology:

basic concepts



First, data is shared across servers. No longer is one machine large enough to hold all our data. However, we want all the data to be accessible together. The answer is to make the storage of several machines look like one big hard drive. That is what technologies such as Hadoop Distributed File System (HDFS) and Amazon Simple Storage Service (S3) do.

Second, processing is parallel. To process the data sequentially would take too long. Instead, the data must be processed in parallel. That means, in many cases, fundamentally changing the way we write software. Many technologies have emerged to make this easier, often even trivial.

Third, processing is pushed to the data. Traditional application architectures pushed the data to the processing. No network is big enough or fast enough anymore to move all our data. On the other hand the computation code is much smaller than the data it processes. We must thus bring the code to the data and distribute our processing across multiple servers and have it compute as close to the data as possible. That is what technologies such as Hadoop mapreduce, Spark and Flink do.v

There are dozens of big data technologies and products, and more announced every week, but they each leverage, expand or support these three basic concepts.

THE BUSINESS OBJECTIVE

You are not collecting and processing data for the sake of processing data. Your objective is to use data to do what you do better. In fact, you want to become better at using data to do what you do better. You are hoping to achieve a strategic breakthrough, somehow transform your business so that you use data in ways that are not immediately apparent now. At the very least, you do not want to be left behind by other companies that are transforming themselves into data-driven enterprises. Companies that are now at the forefront of their industries have come to understand that their data is a valuable asset which they can leverage to drastically improve the way they do business.

You want your firm to evolve to a higher level of analytics. There are four levels of business analytics:

01. DESCRIPTIVE	03. PREDICTIVE
02. DIAGNOSTIC	04. PRESCRIPTIVE

One begins with descriptive analytics, which is basic reporting of what has happened, such as sales reports or sentiment surveys. Descriptive analytics provide only hindsight, but they are a necessary building block to moving forward to higher analytics.

Diagnostic analytics is the application of statistical tools to discover trends and correlations. It seeks to answer why events happen, but it is still retrospective.

The next stage in evolution is *predictive analytics*, to seek what *will* happen. A common example is sales forecasting.

The final objective, however, is *prescriptive analytics*, which is the ability to answer how to make something happen. An example of prescriptive analytics is price optimization.

Level	*name*	*question*
1	**DESCRIPTIVE**	**WHAT HAPPENED?**
2	**DIAGNOSTIC**	**WHY DID IT HAPPEN?**
3	**PREDICTIVE**	**WHAT WILL HAPPEN?**
4	**PRESCRIPTIVE**	**HOW DO WE MAKE IT HAPPEN?**

Sounds perfect: use data to know what actions to take in order to achieve our business goals. How do we get started?

FOUR PHASES

By planning or circumstance, most companies follow four phases in adopting big data technology:

01. PROOF OF CONCEPT (POC)
02. RESEARCH AND DEVELOPMENT
03. FIRST PRODUCTION
04. ENTERPRISE INFRASTRUCTURE

Consider it a natural progression. We will briefly describe these four phases here and then explore each of them deeper over the following chapters.

PROOF OF CONCEPT

The first phase is proof of concept. It usually takes 1 to 3 months. The objective is to prove the hype. Can big data technology actually deliver an order of magnitude improvement over traditional technology?

The company picks a single, well-defined business objective and implements a solution using big data technology. The business objective can either be one that is solved now but causing problems or one that is considered too hard or expensive to solve with traditional technology.

We recommend that the company pick an existing process that takes several hours to run. We call it a "big enough" data challenge. We also recommend that companies evaluate different technology alternatives during this phase.

RESEARCH AND DEVELOPMENT

The second phase is research and development. It is usually 3 to 6 months in duration. The objective is to learn how to build a big data solution as a team.

The company takes the proof of concept or a similar business objective, expands it and turns it into a solution that business users can actually use. The results are treated as beta and run in parallel with any system that is to be replaced.

FIRST PRODUCTION

The third phase is first production. It usually takes another 3 to 6 months beyond research and development. The objective is to move the solution from beta to a proper production system that business users can rely on.

In this phase, the information technology department, operations, support and infrastructure all become integrally involved.

ENTERPRISE INFRASTRUCTURE

The fourth phase is enterprise infrastructure. It can take 6 to 12 months to complete. The objective is to create a platform the whole company can use. How to implement this phase is beyond the scope of this paper. However, each of the previous three phase is designed to prepare you for success in this phase.

NOT EXACTLY LINEAR

The four phases are not as clear-cut as they may first seem. They do follow sequentially, but there can be significant overlap.

More than one department may conduct proof of concept projects. In fact, it should be encouraged so long as there is open communication between the teams. Different departments may have significantly different requirements. The sooner these differences are discovered the less impact they will have in cost and risk later on.

Different business solutions can move through the research and development phase and first production in parallel. The company may want more than one production solution in place before it decides to move into the enterprise infrastructure phase.

STRATEGIC CONSIDERATIONS

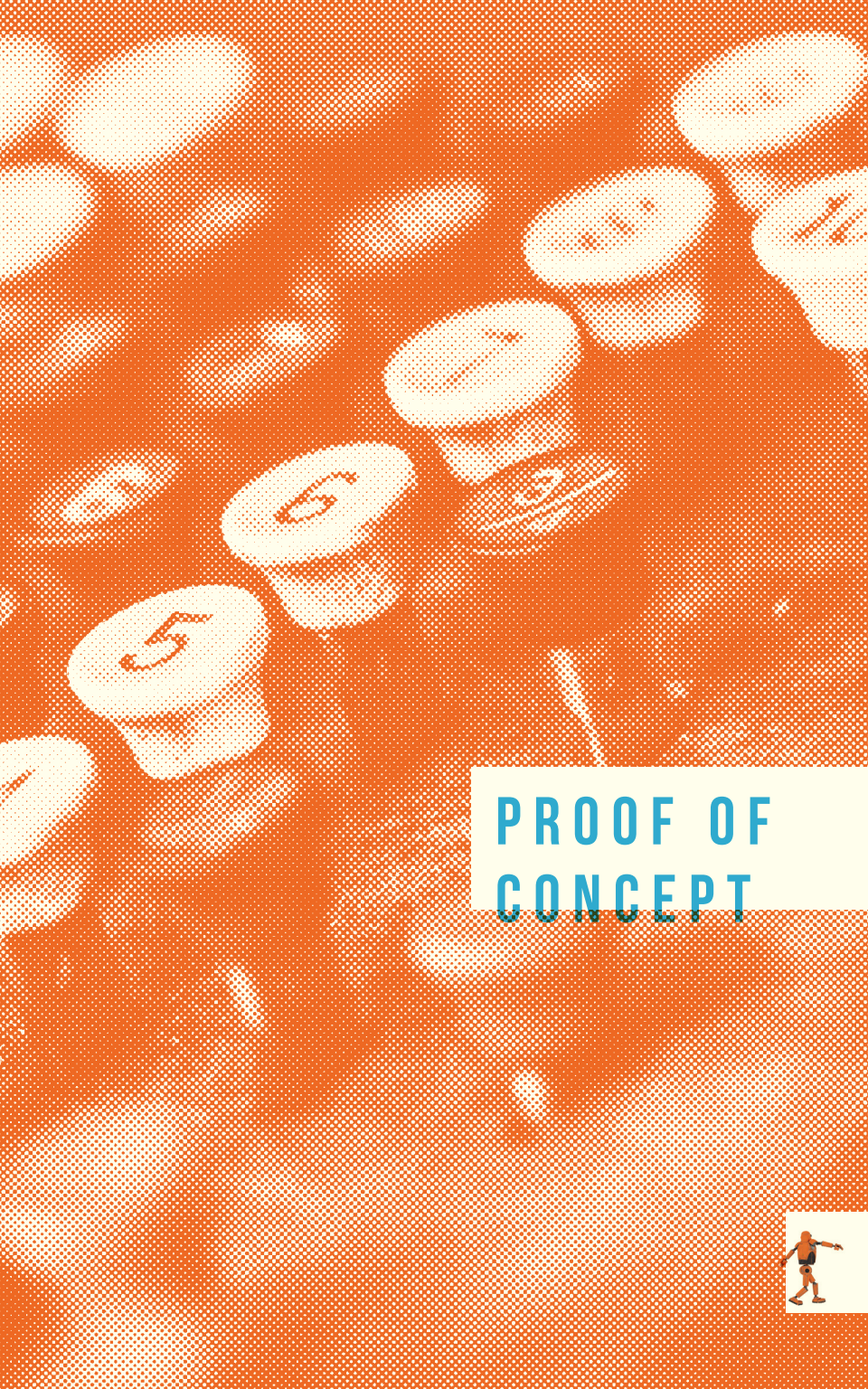
In each phase, there are a number of strategic decisions to consider. The primary ones are:

strategic decisions

- *Software*
- *Hardware and networking*
- *Team composition*
- *Training*
- *Business processes*
- *Road blocks*

In the following chapters, we will describe each of the four phases in detail and discuss how to address these strategic considerations.

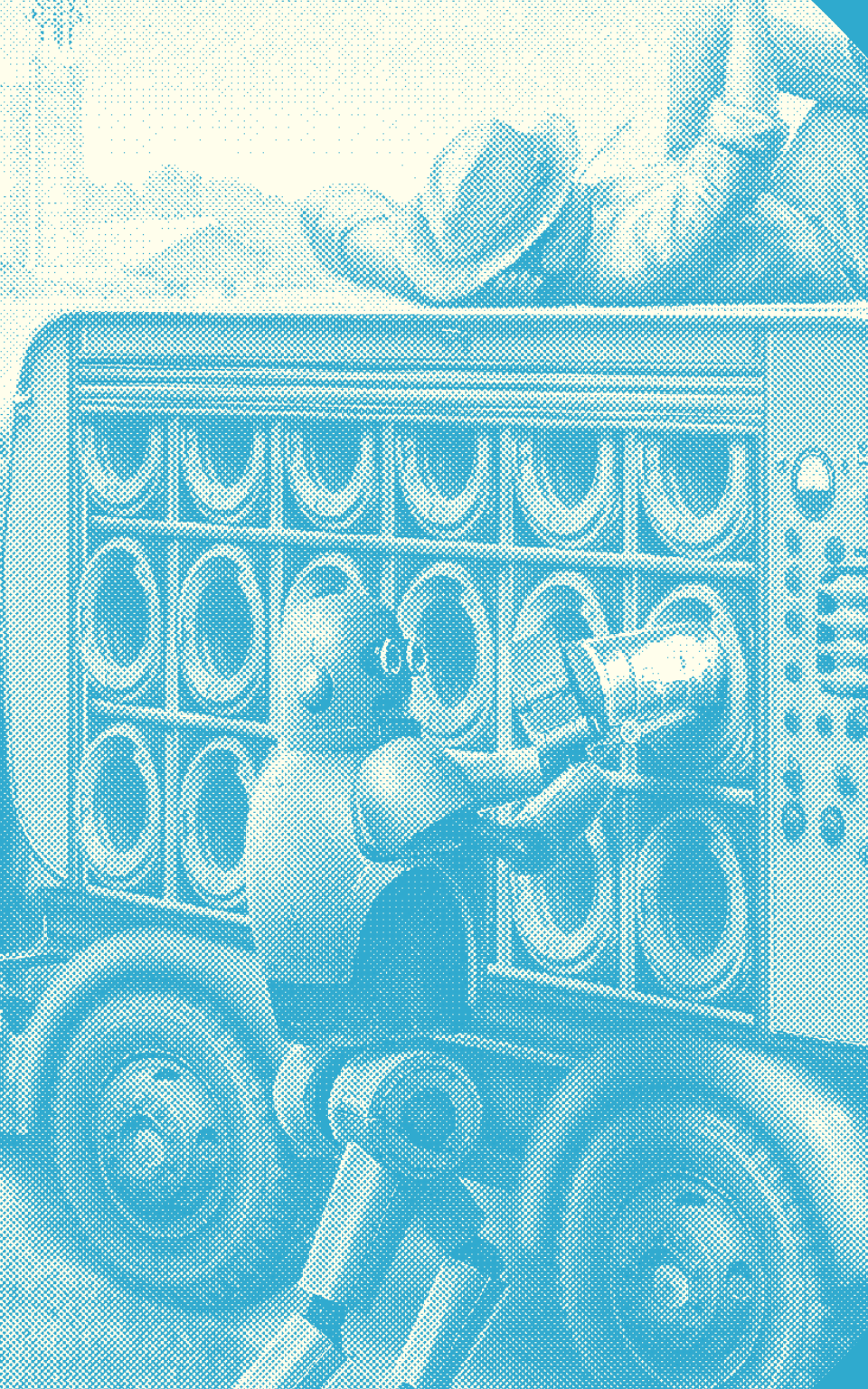
The foremost strategic consideration is that big data technology is constantly moving. It is a natural tendency to make a decision, draw a line in the sand and say, “There, that’s done. This is the way it will be.” Your big data decisions will need to continuously evolve. Build a team and infrastructure that will move with technology, not meet it as it exists today. Your company needs to build wagons, not forts.

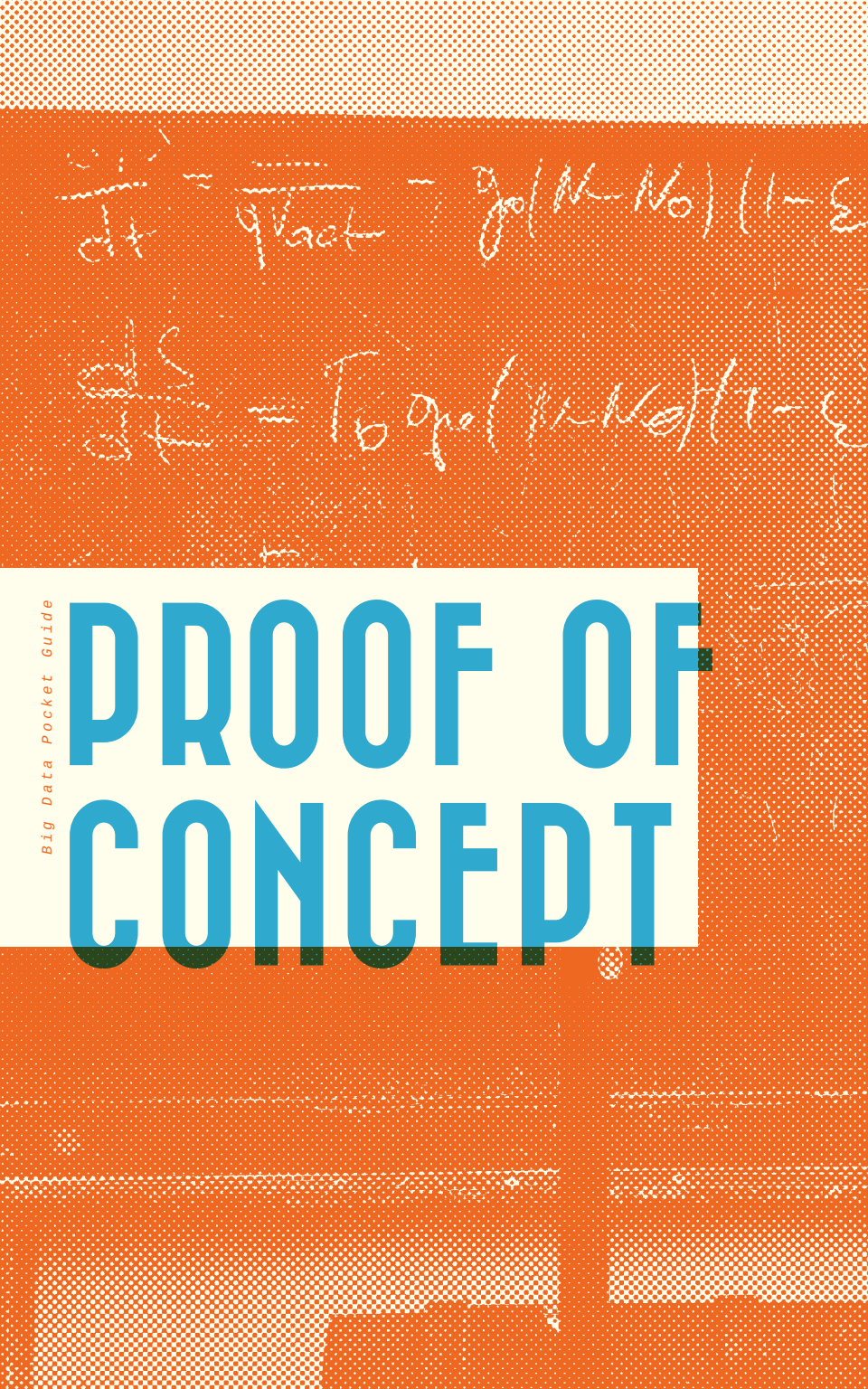


PROOF OF CONCEPT

“The world is one big data problem”

Andrew McAfee




Big Data Pocket Guide
PROOF OF CONCEPT

OVERVIEW

OBJECTIVES

The first phase in big data adoption is proof of concept. A better name might be *proof of value*, since you are proving the value of the technology to your company rather than the technology itself, but the term proof of concept is more readily understood.

Your primary objective of this phase is to prove the technology to yourself. Some other secondary objectives are:

- *Become familiar with the various technologies such as HDFS, map-reduce, Hive, Spark, big tables, data flows, etc.*
- *Become familiar with a big data cluster and the various services*
- *Select a software or consulting vendor to work with in future phases*

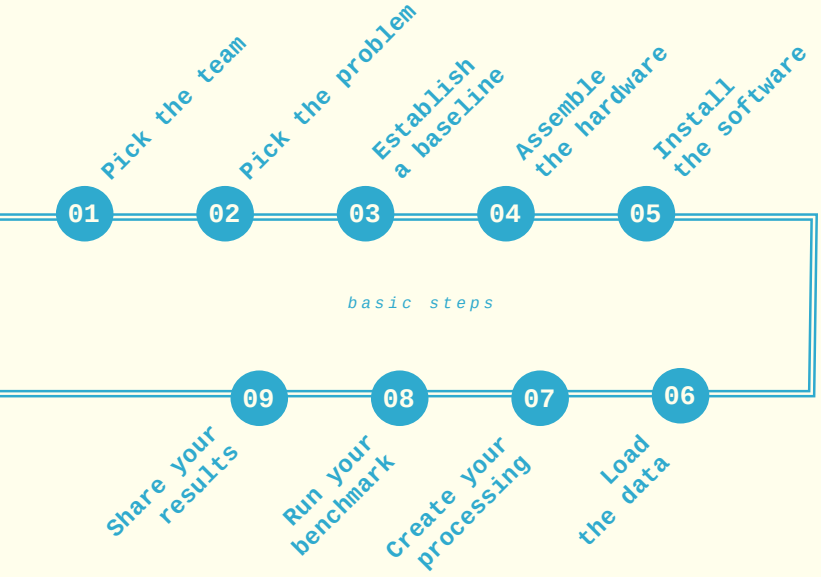
Remember that a proof of concept is about learning. You do not have to get everything right. Failing is learning. Sometimes failing is the most valuable learning experience. For instance, suppose you pick a vendor for the POC that does not work out. Congratulations, you are successful. You now know what vendor you will not work with in future phases. Plus, you probably know a lot more about what to look for and how pick a vendor that will work out.

TIMELINE

The timeline for this phase can vary, but we recommend you plan on three months with six weeks of actual development work. Getting the hardware and network resources you will need will take time. If you are using the cloud, you will still need time for your company to create accounts with the cloud provider, address security concerns and establish safeguards.

PROCESS

A big data proof of concept follows the following 10 basic steps.



We will discuss how to conduct each of these steps.

PICK THE TEAM

Picking the team starts with identifying who in your company will lead the project. The team should be as small as possible. You will need four sets of skills for the project:

- *Subject matter expert, someone who knows the business side of the problem you are tackling*
- *Senior Java or Scala developer, since most of the big data technologies are written in these languages.*
- *Senior database developer, someone who is an expert in SQL and understands how databases work*
- *Senior network and Linux administrator, since big data runs on Linux and is network intensive.*

You may find someone in your company that has all four skill sets. If so, then congratulations, your team is one person. More likely, you will have three or four different employees. Most likely still, you will have only one or two employees on the project and will need consultants to support you in the effort. Having someone from outside who already has experience will help tremendously.

Picking the team also means picking your software partner. You should pick a company that provides a supported distribution of Hadoop. There are several to choose from, including Hortonworks and Cloudera. We will go into why you need a distribution in the *Install the Software* section below.

Each vendor has its strengths and weaknesses, primarily on the software they support and the

services they provide. If the choice is not obvious, then you should pick two or three of the vendors and run the proof of concept with each. It will add time and cost but will lower uncertainty, which is definitely a prime objective of the project.

PICK THE PROBLEM

Success in the proof of concept hinges on defining an objective that is clear, relevant and practical. You do not need to have petabytes of data to process. For instance, you do not need to start processing Twitter feeds just so you have a lot of data. Focus on the processing time, not the data size. We recommend that you pick an existing process that takes several hours to run. We call it a “big enough” data challenge. It should be gigabytes of data if possible. Megabytes is probably too small and terabytes is fine but not necessary.

Another option is to replace an existing data warehouse process. Many companies face the challenge that their data warehouse content and processing loads keep growing linearly or even exponentially while their hardware and processing power grow stepwise or not all. Big data provides a less expensive means to create your analytics.

You will not process the data the same way on big data as you do in the data warehouse. You will not be creating a star schema, for instance. However, you should be able to provide the same end results, that is, the same report table. The data should take much less time to create and be much faster to access. In fact, you should shoot for 10 to 100 times faster.

ESTABLISH A BASELINE

In order to prove success, you need data. You need to know where you are starting in order to know how far you go. If you have picked an existing process, then establishing a baseline is much simpler.

You simply observe the characteristics of what is running now. The three primary characteristics are:

01. DATA VOLUME
02. PROCESSING TIME
03. HARDWARE

Data volume is the size of the data (gigabytes or terabytes) that is being processed. There is the base amount of accumulated data and the amount of data that is added each day. If data is added with files, then you will need the number of files and the distribution in sizes. For example, you may receive 10 to 100 files each day with a range in size from 60 to 120 GB each, with a daily average of 60 files and an average size of 75 GB.

Processing time is both CPU time and end-to-end wall clock time. The extract-transform-load (ETL) process time should not be ignored. In fact, you may find that faster ETL is one of the biggest gains in the proof of concept. Loading data into traditional databases is usually complicated, brittle and slow.

For hardware, you need the number of processors, the CPU speed, the amount of memory (RAM) and the amount of disk space. Moving from a traditional system to a distributed system like Hadoop means that you will move from one server to multiple servers, but these metrics will be important when presenting the results.

Depending on the particular problem, other metrics may be relevant. However, this covers the basics.

ASSEMBLE THE HARDWARE

Big data is distributed processing, which means that it runs on multiple servers. With most new technology, a developer just downloads the software to his laptop and starts hacking. You can't do that with Hadoop. It requires a minimum of four servers to work properly. Most developers do not have four servers lying around that they can play with to learn a new technology.

You should consider using cloud computing for your POC. Hadoop distributors have made getting a cluster running in the cloud very easy. However, for some companies, cloud is not an option or another POC in itself.

We recommend, as a minimum, 8 computers with 8 GB RAM, dual processors and 1TB hard drives. The more hardware you have, the better. The servers should be identical if at all possible.

Note that Hadoop requires Linux. Check the Hadoop distribution for which Linux distributions it supports. Picking the right Linux distribution for your Hadoop will make installation much easier.

You will need to get your network administrators involved. You will need fixed IP addresses for the servers and names assigned in your domain name service (DNS).

You will also need to have the servers on the same router, otherwise you will choke the office network with traffic. Hadoop is very network intensive. Your servers also need internet access to download all the various software needed. Hadoop is actually many different projects, so automated download is essential for easy installation.

INSTALL THE SOFTWARE

You should use a Hadoop distribution provided by a vendor. Yes, Hadoop, Spark and the whole zoo of technologies are open source, but building, installing, configuring and integrating all of those packages yourself can be challenging. And, once you have it running, you would need to upgrade the software with each release, which seems to occur constantly. It would require a great deal of expertise and time that can be better spent on solving business problems specific to your company. A Hadoop distributor such as Hortonworks manages this pain for you. Plus, they offer limited-time free versions of their distributions specifically for proof of concepts.

Hadoop distributors have made deploying a cluster much easier. Even with using a distribution, however, installing a Hadoop cluster can be complicated, especially when doing it for the first time. There are literally hundreds of parameters to set in Hadoop. We recommend that you bring in help or have access to support from your Hadoop distributor in case you run into problems.

Do not be afraid to wipe it all out and start over again. Better to experiment and learn how to configure a cluster during the POC. It becomes much more expensive in the later phases.

LOAD THE DATA

If possible, load all your data as-is into HDFS. Avoid any transformation before loading. There are three reasons for this: traceability, flexibility and speed. First, if the original files are in HDFS, you can trace how you transformed the data from one form to another all the way back to the source. Second, you have the flexibility

to transform the source into other formats at a later time. Third, you have the full speed of Hadoop parallel processing to do all these transformations.

Keep track of how long it takes to load the files. This will be helpful later when calculating your benchmarks and presenting results. Consider that a huge time cost in traditional systems is the actual ingest of data into the database. If your source data is delimited text files, you can declare a folder with the metadata in Hive (one of the databases in Hadoop), upload the files as-is to an HDFS folder and have the data queryable in seconds.

Be aware that some Hadoop data ingest tools can overwhelm a traditional database you might be reading data from. Hadoop by nature wants to operate in parallel. With a simple command line instruction, you could have every processor on every server in your cluster requesting data from the database, which could easily flood the database's connection pool.

CREATE YOUR PROCESSING

Use the technology that is most comfortable for your team. If your team is strong in relational databases, the use Hive and write your jobs as SQL joins. If your team is strong in scripting languages such as Python, then use Pig to write your data processing jobs. If you have Java or Scala programmers, then use classic map-reduce or better yet, Spark to write your processes.

Keep the data processing focused and as simple as possible while still meaningful. You are proving the technology, not building an application. Your process does not need to be production grade code. It just needs to get the data from source to target at a speed and scale that is a significant improvement for your business.

RUN YOUR BENCHMARK

As mentioned earlier, consider the data ingest time as part of your benchmarks. It is a matter of seconds to load megabyte text files into HDFS and have the data immediately queryable in Hive. That is not the case for traditional databases. Highlight that advantage to your audience.

Your processing time should scale completely linearly. For example, if it takes 2 hours on 4 servers then it should take 1 hour on 8 servers. Demonstrating this linear scalability should be easy during the POC. Seeing is believing.

SHARE YOUR RESULTS

Do not hide your light under a bushel. Share your results with the rest of the company as broadly aspossible. Publish on an internal wiki or social media if you have it. Have presentations with question and answer sessions.

Some lessons learned from other POCs:

- Spend at least half of your time explaining the basics of what Hadoop is and the business problem you were solving
- Present solid, specific metrics regarding the servers used, the amount of data and the time to process
- Do not belittle the technology or the solution you are replacing.
- Emphasize that Hadoop does not replace all other existing technologies. It does, however, make certain types of problems easier, faster or even possible to solve.
- If you have worked with a software or consulting vendor, have them participate in the presentation but do not let them turn it into a sales pitch.

Remember that the POC is just the first step of your journey. The presentation is vital for you to muster the internal support necessary to move on to the next phases.

CHALLENGES

There are four common challenges that arise in conducting a big data proof of concept:

four common challenges

01
Underestimating the time to get servers

02
Picking the wrong problem

03
Letting the IT department lead

04
Underestimating internal opposition

We will quickly discuss each of these challenges.

UNDERESTIMATING THE INFRASTRUCTURE LEAD TIME

In a large firm, getting the servers and networking you need to conduct the proof of concept is usually the hardest and longest task. Large firms have strict network policies and longer purchasing cycles. Ordering servers and getting them installed can take weeks.

If you decide to use a cloud provider such as Amazon, Microsoft or Google, you will still face lead times, especially if your firm has never used cloud before. You will need to set up accounts and policies. You will also face security issues on what data can be uploaded to the cloud.

PICKING THE WRONG PROBLEM

Do not pick a problem that is too big. You do not want to get bogged down in a lot of development work. You should keep your team small and your deadline tight.

Do not confuse a proof of concept with developing a product. Your objective is to learn the technology and what it can do for your business. The proof of concept is only the first step. You will have time to get a working beta product in the next phase.

However, do not pick a problem that is too trivial. If you do, then your results and benchmarks will be meaningless or dismissed as irrelevant to the business.

LETTING IT DEPARTMENT LEAD

If you are a large firm, do *not* use your information technology (IT) department to lead or even conduct your big data proof of concept.

Yes, the IT department has the skills in software, hardware and networking needed. Yes, you will probably need their help with getting hardware and networking setup. However, the primary purpose of the IT department is to keep production systems up and running smoothly. All other purposes are secondary. Keeping production systems running smoothly requires minimizing or eliminating change, disruption and risk. Successful IT departments are *risk averse* by the very nature of their mission.

Creating a proof of concept is a research and development effort. A successful proof of concept is all about change, disruption and risk. You need a research and development culture to conduct a successful proof of concept. That culture is nearly the polar opposite of an IT department. It is not just about knowing the technology; it is about having culture that embraces risk.

UNDERESTIMATING INTERNAL OPPOSITION

Prepare for the naysayers. There are people within your firm who identify their value with the technology that they support. They can easily perceive big data technology as a threat to themselves and to the firm. It is important to emphasize that big data technology does not replace all other existing technology. It is also important not to belittle the solution that you are benchmarking against in the POC. Focusing on the positive will help the company accept and embrace that big data technology makes certain types of solutions easier, faster or even possible.

RESEARCH & DEVELOPMENT

Pearl Zhu, Digital Master

“We are moving slowly into an era where Big Data is the starting point, not the end.”


FUTURE

Big Data Pocket Guide

RESEARCH & DEVEL-OPMENT

OVERVIEW

OBJECTIVES

The second phase of big data adoption is research and development. You have completed the proof of concept. You have learned more about the technology and proven to yourselves as a team that big data has the potential to provide significant improvements for your business. Now it is time to solve a real business problem.

The objective in the previous POC phase was to prove the technology to yourself. The primary objective of this phase is to prove the technology to the business. Some other secondary objectives are:

- *Learning the technologies in depth*
- *Getting use to managing a cluster, including installing, configuring and monitoring services*
- *Building a core team that can create big data solutions*
- *Solve a real business problem*
- *Have business users beta testing the solution*

If you are taking on an existing problem, then you will build your solution in parallel. Most likely, you will consume the same data as the existing business application and process it on the new platform. At the end of the phase, you should be able to demonstrate that the big data platform produces the same or better results in less time.

If you are taking on a new problem, then at the end of the phase you should have a minimum viable application that is demonstrable to the business.

TIMELINE

The research and development phase should take 3 to 6 months, depending on the scope of the problem and the size of your team. You should not attempt to solve a problem that will take more than 6 months. The team will be discouraged and the business needs to see results.

PROCESS

A big data research and development project follows the following seven basic steps.

01. Pick a business problem
02. Have beta users
03. Expand the team
04. Expand the cluster
05. Learn the technology
06. Build the solution
07. Run the process in parallel

These steps build upon the work and experience your team gained during the proof of concept project. As such, there will be fewer details to cover.

We will discuss how to conduct each of these steps.

PICK THE BUSINESS PROBLEM

As with the POC, your success is determined to a large degree by which business problem you chose to solve. If the problem is too large or complex, then you are setting yourself up for failure or, at the very least, expense and delay. If the problem is too small, then the business will be underwhelmed by your results.

The R&D project does not have to be the same problem used in the POC, but it helps. It should have a larger scope than the POC. You will want to pick a project that you can take into production. However, do not treat this phase as a production development. It is still too early for that.

Start with a batch process. Real time streaming is the cutting edge for now and it will eventually become mainstream, but it is not there yet. Manage your risk and start with batch processes or micro-batch such as Spark. You will have time to evolve as you learn.

HAVE BETA USERS

In this phase, you need an engaged customer. Picking your beta users goes hand-in-hand with picking the business problem. One can usually find a technology enthusiast in the business side, or someone who seeks out and embraces change.

Your internal customers should be your best champions. On the other hand, an unsatisfied customer can kill your project. You will need to manage customer expectations. Hadoop is not a panacea, a solution to all your technology woes. Hadoop does not replace all existing technology. You may be surprised that the legacy relational database does some things remarkably better than Hadoop. Your customer should know not only the strengths but also the weaknesses of Hadoop.

EXPAND THE TEAM

If you found one person who had all the skills you needed in the POC phase, lucky you, but that one engineer will not be enough for R&D. You will need a team. There are three distinct roles in big data development: data administrator, data engineer and data scientist.

- **Data administrators** are responsible for creating, configuring and monitoring the cluster. A data administrator must have Linux administration experience and should have experience in networks and database administration. They will need specialized knowledge with Hadoop. There are hundreds, perhaps thousands, of parameters to set in configuring a cluster. Some can have significant impact on the performance of your data processes.

- **Data engineers** are responsible for the loading, transformation, storage and processing of data on the cluster. A data engineer must have experience with Java or Scala and should have experience with SQL databases. They will need specialized knowledge with Hadoop, mapreduce, Spark and NOSQL.

- **Data scientists** are responsible for finding meaningful information in the data. A data scientist must know R or Python and should know machine learning libraries.

If you did not get contract support from a Hadoop distributor during the POC phase, then you should give it serious consideration for the R&D phase. A single phone call or email can provide the answer you need within a matter of hours which could otherwise cost your team days of frustration trying to research and solve on their own. The Hadoop technology base is huge and constantly changing.

No one can know it all. Most distributors now provide pro-active help. They monitor the performance of your cluster and provide recommendations on how to improve it.

EXPAND THE CLUSTER

The cluster used in the POC is not going to be big enough. You will need more resources. You will want to rebuild the cluster with high-availability. Create separate control nodes and worker nodes. Your Hadoop distributor should be able to provide you reference hardware implementations.

In this phase, we recommend that you use cloud computing for your R&D cluster. It is not a matter of direct cost. In fact, cloud computing will probably be more expensive. Rather, it is opportunity cost. Your organization is still learning. You could take weeks to buy 10 expensive servers only to find out in 3 months that they are the wrong configuration. It happens. On the other hand, you could start up a cluster in the cloud in a day and three months later change the hardware configuration completely with a click of a mouse.

LEARN THE TECHNOLOGY

During the POC, your team only became acquainted with Hadoop technology. Now is the time to dive deep and learn.

If you have experienced Hadoop developers, then have them cross train your other team members. Otherwise, spend the time and money to send your engineers to training. A week of hands-on instruction is worth four weeks of web searching, trial and error. Do not get me wrong. You want your team to conduct trial and error, but their effort will be much more valuable if it has a foundation of knowledge from which to expand.

If you are an agile development shop, you can still run your R&D project in sprints. You will not have user stories for much of your work. Instead, you can have hypothesis stories. A user story enables a developer to focus on solving a customer requirement as quickly and succinctly as possible. Likewise, a hypothesis story enables a developer to focus on solving a technology problem or question as quickly and succinctly as possible. It is all too easy to get lost testing things simply because they are there. A hypothesis tells you what you are testing and why you are testing it.

BUILD THE SOLUTION

Learning the technology is the research part; building a solution is the development part. Having a business problem to solve will focus your efforts and make the learning more practical and effective. It also communicates the results of your learning much better to the business. Remember, though, that you are not building production code. This is an R&D project. What time you would spend hardening your code you should spend instead experimenting.

When building your solution, take the simplest, most brute force approach first. See if it works. Establish a baseline then evolve your solution with more efficient processes and structures. Distributing computing requires a different way of programming. Sometimes an approach that would be more efficient in a traditional object-oriented application will be highly inefficient in a distributed process. You will only learn by trial and error.

Not all of the work is data science. Much of the work will be getting the data into the cluster and making it usable. You should treat these processes as first-class engineering problems and not some side distraction. Working in an enterprise, you are accustomed to extract-transform-load (ETL) processes and probably have commercial tools with

the firm to handle such jobs. As noted in the POC before, you should load data directly into Hadoop as is. In other words, you should build extract-load-transform (ELT) processes, not ETL processes. There is a significant difference.

In the POC you probably ran all your processes manually. In this phase, you will need to automate your processes. There are several Apache projects for handling the scheduling and automation. Oozie is the most widely supported. Each has its advantages and disadvantages. You should allot time to find which one meets your needs.

RUN PROCESS IN PARALLEL

There are two paths to take in your R&D effort:

two paths

create a new process that was impossible or impractical to solve before	*replace an existing process in a much faster or more efficient way*

If you took the second path, then you will need to run your new process in parallel with the old process in order to validate your success with the business. They need confidence in the process so that you can move on to the next phase. You should build automated reconciliation between the new data and the old data. Of course, you should run that reconciliation in the Hadoop cluster. Reconciliation is a natural mapreduce problem.

Once your beta users are satisfied with the results, it is time to move on to the next phase: first production.

CHALLENGES

FINDING EXPERIENCED ENGINEERS

Experience in big data is in extremely high demand. If you can locate consultants or employees available, you will find that they are expensive. One alternative is to grow your own, that is, pick smart employees and pay to have them trained in the technology. If cost and time are constraints, then you should consider having offshore development teams that have on-shore leadership. The offshore team should be as close to your time zone as possible to enable team interaction. We recommend 4 hours of overlap between the onshore and offshore teams. Otherwise, there will be too many delays as issues take more than one business day to resolve.

KNOWING WHEN THINGS GO WRONG

When your beta users see data that does not match their expectations, they will be left to wonder what went wrong.

- *Is the source data corrupted?*
- *Is there a bug in the processing?*
- *Is the model or expectation wrong?*

Human nature being what it is, users will usually assume that there is something wrong with the data or the code before assuming there is something wrong with their model.

You should also have unit tests for your code. It is not straightforward with mapreduce code, but it is possible. You will also need to create some ad hoc means to check your data integrity. We will discuss these issues more in the next phase.

OVERWHELMED WITH THE TECHNOLOGY

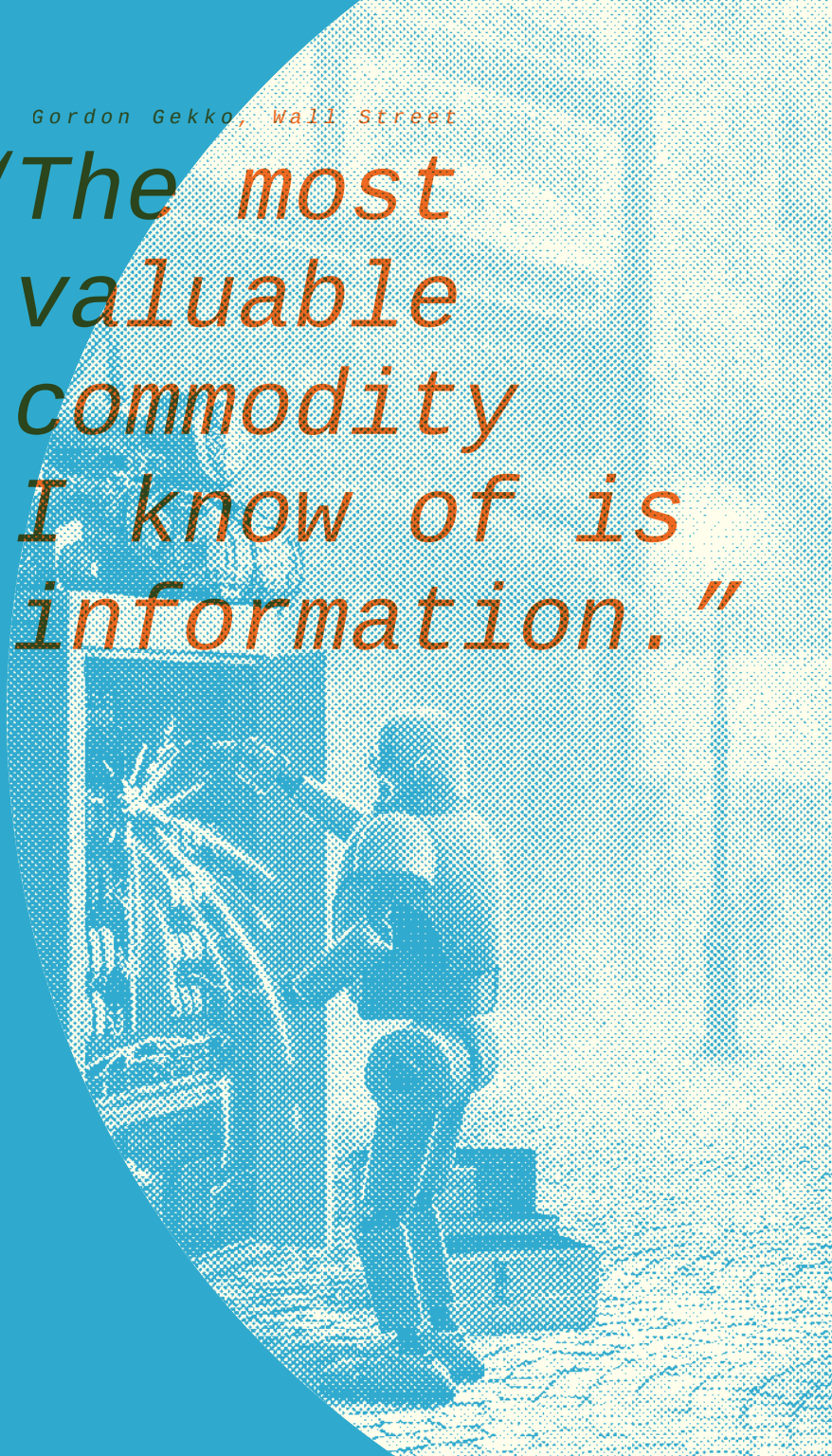
Apache Hadoop is a very large and ever growing ecosystem of technologies. Hortonworks provides support for 22 Hadoop related open source projects (or applications) but that is only a subset of all the projects. In many cases, there are two or more projects that serve the same or overlapping purposes. That is just the projects adopted by the Apache Foundation. There are dozens of other open source projects released and supported by large technology companies that are not managed in the Apache Foundation. New projects are announced every month. These various projects are interdependent and each of these projects is releasing new features all the time. It can be overwhelming.

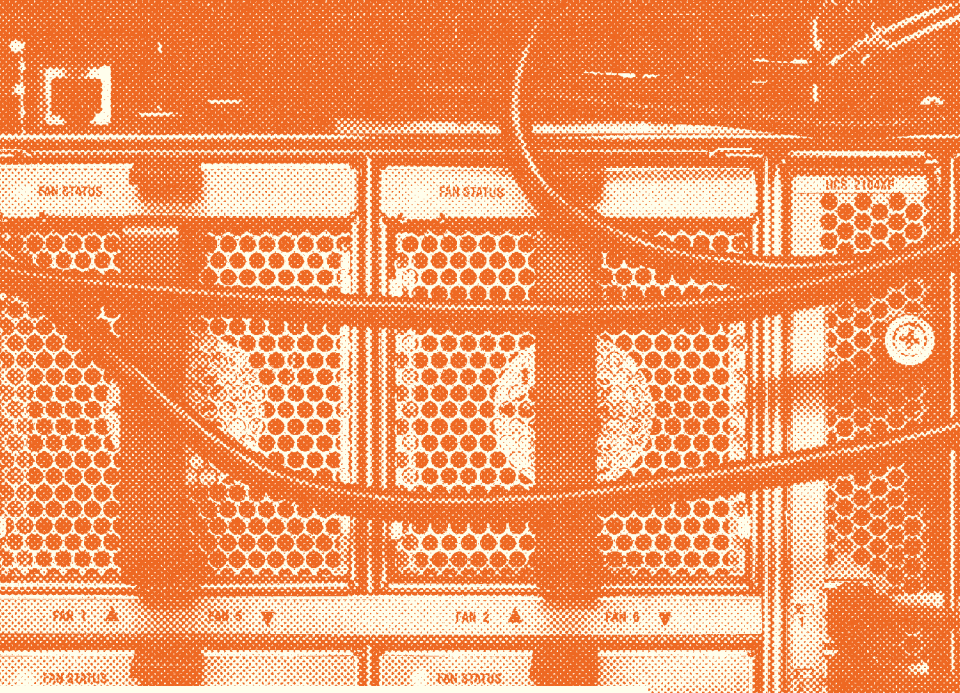
Pick and choose your scope. You cannot learn everything at once. Be aware of the various projects but focus on the ones that you need. It is very easy to chase after a new project simply because it is new. Remember that it usually takes a year or longer for these projects to mature and stabilize into something you can deploy into production.

In order to keep pace with changes, consider allocating 10 percent of your team’s time (four hours per week) to continually read up and test new technologies.

Gordon Gekko, Wall Street

“The most valuable commodity I know of is information.”





Big Data Pocket Guide

FIRST PRODUCTION

OVERVIEW

OBJECTIVES

The third phase of big data adoption is deploying your first production application. You have completed your research and development project. You have a data process running in parallel with a legacy process, or you have a completely new data process that was unattainable with previous technology. The primary objective of this phase is to enable users to run the business relying on the data produced by the cluster.

Up till now, your Hadoop cluster has been a sandbox that you can play in. From now on, the business will be relying on the data. If a process fails or the cluster goes down, there will be consequences.

TIMELINE

The research and development phase should have reduced the risk and complexity of this phase. The first production phase should take 3 to 6 months, depending on the scope of the problem and the size of your team.

PROCESS

A big data first production project follows the following seven basic steps:

01. BRING IN IT SUPPORT
02. SPLIT THE CLUSTER
03. ESTABLISH SECURITY
04. BUILD QUALITY CHECKS
05. BUILD THE USER INTERFACE
06. IMPROVE THE CLUSTER
07. IMPROVE THE PROCESSES

Most of these steps can proceed in parallel. These steps build upon the work and experience your team gained during the proof of concept project and the research and development project. As such, there will be less details to cover.

We will discuss how to conduct each of these steps.

BRING IN IT SUPPORT

We strongly recommended that you not let the information technology (IT) department run the POC or R&D phases of your big data development. The reason is that successful IT departments reduce risk, while R&D is about embracing risk.

You have now entered the phase of big data adoption when you want to begin lowering risk. It does not mean that you will stop experimenting. Big data technology is still young, so it will continue to evolve at a frantic pace. However, from now on, your production big data will be separated from your experimental big data.

As such, it is time to get the IT department involved in your big data efforts. You will need to have IT support trained as data administrators in order to monitor and maintain your production cluster.

SPLIT THE CLUSTER

You will have noticed during the R&D phase that the cluster seems to gradually grow in size. That is natural and will continue. At this point, however, you will need to make a big increase in the size of your Hadoop infrastructure. You should create two clusters: a development (DEV) cluster and a production (PROD) cluster. There is no practical need for creating a test, user acceptance testing (UAT) or other stages that you might have in your other application environments. It is overkill. Hadoop is a distributed resource sharing platform. Your DEV platform can serve all of those other roles. What is important is that you have one cluster that your engineers can break and another that your internal customers can rely on to be always up and available.

You have two options. You can either keep the existing cluster as DEV and create a new cluster for PROD, or you can designate your R&D cluster as PROD and create a new DEV cluster. There is no right or wrong approach. For the most part, it will depend on your IT policies.

ESTABLISH SECURITY

Your production cluster will need to be secure. Hadoop came out of a laboratory, so security was not a pressing issue in its first versions. Security in a distributed processing environment is not trivial. However, Hadoop has been in the enterprise for some time now, so there are excellent enterprise-strength security options available. For the most part, these solutions are built on Kerberos.

Your IT department will need to drive this process. Most Hadoop distribution vendors provide special professional services to help with setting up a secure cluster.

Your security should focus on enabling two things: monitoring resource usage and controlling data access. In this phase, you probably only have one department using the cluster. However, Hadoop is a shared resource. You should treat the first production project as the first tenant of many in a multitenant environment.

BUILD YOUR QUALITY CHECKS

In the R&D phase, we discussed the challenge of knowing when things go wrong. In a production environment, it is critical that you monitor the quality of the data. You will be loading data from different sources, from different departments within the company or different data vendors. The source of the data is out of your control, so you cannot assume that it is correct.

There are basic monitors, such as alerting if files do not arrive by a certain time. There are basic checks, such as checksum or parsing to insure the files are not corrupted. You should go beyond these basics and check the data itself with statistical analysis. For instance, if it is account data, then there should be a distribution for the number of accounts and transactions received each day. Your quality check processes should automatically alert if there is a significant deviation.

BUILD YOUR USER INTERFACE

Customers focus on what they understand. You should not expect your business users to understand big tables and mapreduce, so do not expect much feedback on the Hadoop technology itself. They will understand the data and the user interface, however, so expect plenty of feedback on that.

There are several applications on the market that provide self service data analytics for Hadoop. Work with business users to build the data sets that work best with these tools. Business users should be able to visually explore the data to find trends and anomalies.

There are also analyst notebooks such as Jupyter and Zeppelin for the power users. The clever business analyst that builds complex models in Excel will love these applications. The analyst notebooks put machine learning algorithms in the hands of your users. As you progress towards becoming a data driven enterprise, you should encourage each department to have their own data science capabilities.

IMPROVE THE CLUSTER

Improving the cluster can mean increasing the capacity by adding more worker nodes. Hadoop is linearly scalable, so you should see incremental and proportional improvements in performance for every server you add to the cluster.

Improving the cluster also means tuning the configuration settings. There are literally a thousand configuration properties you can set and adjust that alter the behavior and performance of the cluster. You will need to adjust the parameters to match the the time of jobs that you run on your cluster. You should rely on your Hadoop distribution partner to help in this effort; it is a key part of the value that they add.

IMPROVE THE PROCESSES

We recommend that is most cases you take the simplest, most direct approach in your first solution of a big data problem. In other words, use the brute force approach and get something working. In massively parallel processing, the brute force approach often works surprisingly well. You may waste hours of development work creating an optimized or elegant solution that does not really significantly improve the performance of the application. Go with “good enough” first. You can always make it better later.

It is the nature of big data to constantly grow in size and complexity. A process that may have finished quickly enough with 1 terabyte (TB) may not meet the service level agreement (SLA) at 10 TB. As such, you will need to refactor your solution periodically over time. For instance, partitioning the data can have significant impact to performance in certain scenarios.

Your performance over time should look a saw blade: sharp immediate improvements with gradual degradation over time.

CHALLENGES

Some of the challenges that you faced in the R&D phase, such as finding experienced engineers, will persist. You will also face new challenges in the first production phase.

RESOURCE CONFLICTS

Your big data application is now in production, so a delay or failure has consequences for the business. It is easy to guarantee resources when you have only one tenant on the cluster. Things become a bit more complicated as you expand the usage of the cluster and add more tenants and more processes.

With YARN, Hadoop has evolved to enable multi-tenant usage of cluster. Enabling the functionality and avoiding all conflicts, however, are not the same. There are always finite resources. You will face the need for setting priorities between tenants and even between processes. This will require negotiation, leadership and some accounting.

KNOWING WHAT IS IN PRODUCTION

A major transition for the first production cluster is how changes are deployed. Up till now, your engineers probably worked directly on the cluster, editing and testing in an ad hoc fashion. Now that has to change.

You must control how code changes are introduced into the cluster. Do not allow developers to load scripts into the production cluster directly. Use DEVOPS practices of deploying new code by script and auditing all changes. You should be able to know what version of what code was running at any given time.

CREATING A DATA-DRIVEN CULTURE

Our discussion has focused on technology and its implementation, but the technology is not an end in itself. The data produced and processed by the technology is not even an end in itself. They are but means to an end, which is to change the way you conduct business, to embrace data as a core to decision making.

At the end of it all, big data is about culture change, which is much harder than mere technology implementation. Big data is a tool, a powerful tool and, yes, even an expensive tool, but unless you change the way you do business and embrace the power the tool provides, your success will be limited. Without culture change, you will not achieve the full benefit of the costs you have incurred.

CONCLUSION

CONCLUSION

LET'S SUMMARIZE WHERE WE ARE.

The first phase is proof of concept. It usually takes 1 to 3 months. The objective is to prove the hype. Can big data technology actually deliver an order of magnitude improvement over traditional technology?

The second phase is research and development. It is usually 3 to 6 months in duration. The objective is to learn how to build a big data solution as a team.

The third phase is first production. It usually takes another 3 to 6 months beyond research and development. The objective is to move the solution from beta to a proper production system that business users can rely on.

We had several objectives in this short playbook. We wanted to show you how to get started with big data. We wanted to layout what the first 12 critical months or so will look like and how to be focused and successful in that short period of time. We wanted to give you a simple step by step guideline to get moving now and know where you are going.

I hope that we have achieved those objectives for you.

These first three phases in 12 months or so are just the beginning. Once you have completed the phases described in this book you will be ready to begin building an enterprise big data platform - an infrastructure for the whole company to use.

Big data is just a means to a much bigger end goal: to become a data driven enterprise. You want your firm to evolve to a higher level of analytics, from descriptive to diagnostic to predictive to prescriptive. You want to reach prescriptive analytics, the ability to answer how to make something happen. You want to close the gap from data to decision to action.

Becoming a data driven enterprise is not a technology change; it is a culture change empowered by a technology change. The technology is not enough. It is not a “build it and they will come” scenario. Without leadership, training, incentives and all the other aspects that make culture change, the technology investment will never meet its full potential.

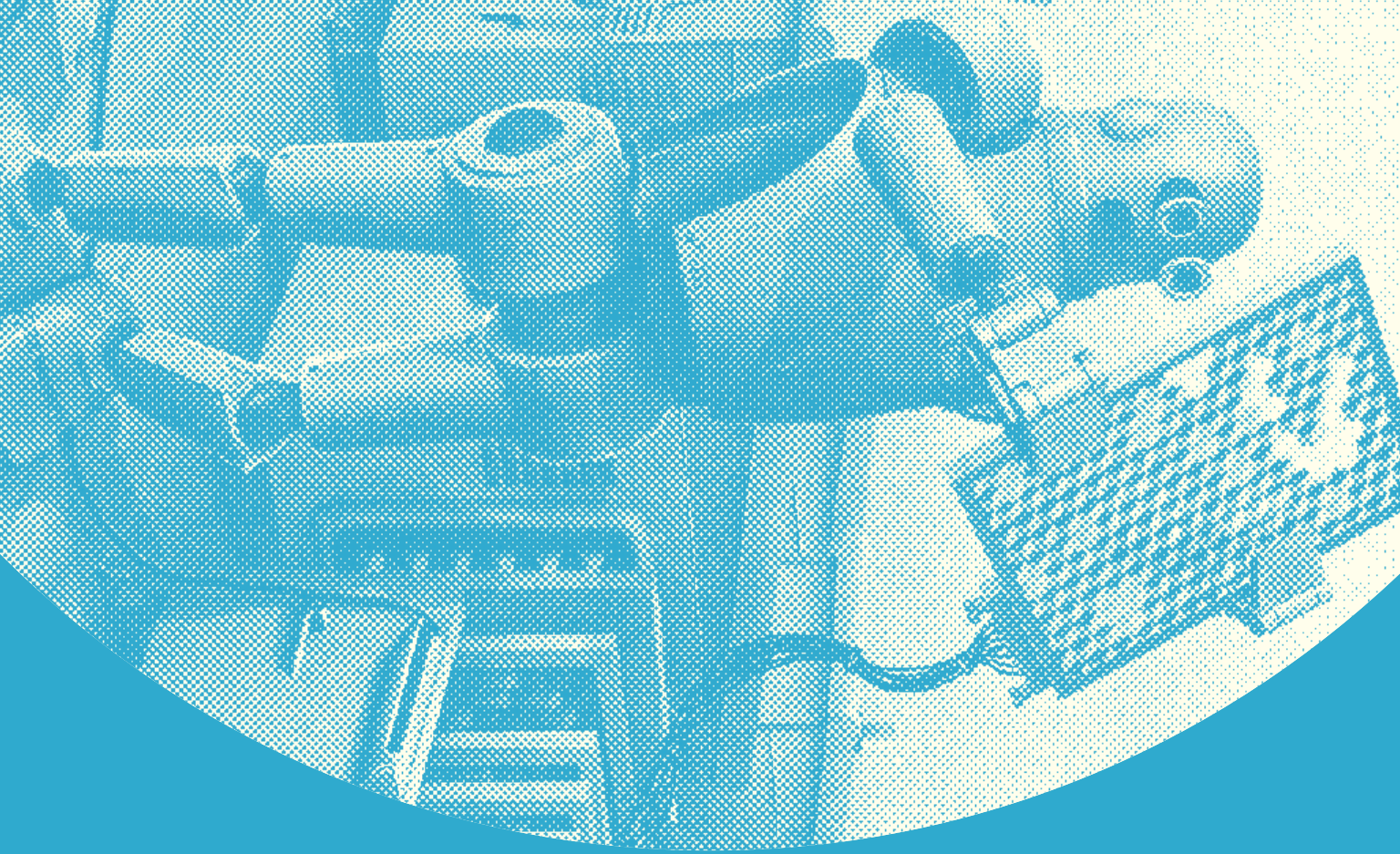
Finally, not to be too cliche, big data is a journey, not a destination. You will never be done. The technology itself is changing constantly. The change itself is accelerating. The opportunities for change are constantly expanding.

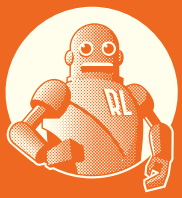
So go and **GET STARTED.**

And let us know how we can help.

"In God we trust.
All others must
bring data."

W. Edwards Deming,
statistician, professor,
author, lecturer,
and consultant.





RICKER
LYMAN
ROBOTIC

https://rickerlyman.com

RICKER LYMAN
ROBOTIC

ABOUT US

Ricker Lyman Robotic is a US-based company focused on streaming big data and the Internet of Things. We are a complete solutions firm, providing software, hardware and professional services.

We have a wholly owned subsidiary in Lviv, Ukraine, a picturesque city with a large world class talent pool of engineers.

Robotic means expanding the use of automation as far as practical into all aspects of our lives.

Previous generations had a vision of the future, where technology ubiquitously serves our needs, where we ceaselessly conquer the frontiers of space and knowledge. Our company is dedicated to this vision. Technology can be simple yet powerful, intuitive yet awesome.

We intend to make it so.

www.rickerlyman.com