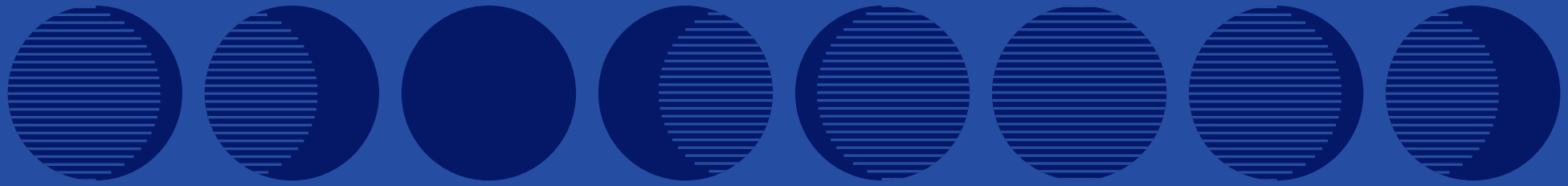
informatica



How to Run a Big Data POC—In Six Weeks.

A practical workbook to deploy your first proof of concept and avoid early failure.

Contents

Tip: Click on parts to jump to the particular section you want

Introduction

The Importance of Starting Small.

The Importance of Starting Small.

At this point there's little doubt that big data represents real and tangible value for enterprises. A full 84 percent of them see big data analytics changing the way they compete[1]. And companies like Uber[2] and Airbnb[3] are disrupting whole industries with smarter operations fueled by big data.

But even though the potential for big data innovation is significant[4], and even though the low cost of storage and scalable distributed computing has shrunk the gap between ideas and implementations, the journey to big data success is a tricky one.

A new set of technological and organizational challenges makes big data projects prone to failure. But if there's one thing that early big data projects have proven, it's that you need a phased approach—not only to make projects more manageable, but also to prove the value of big data to the enterprise.

That means starting with a well-planned proof of concept (POC) and building toward a coherent vision of big data-driven success. The aim of this workbook is to share the advice and best practice needed to run a successful big data POC.

Central to this is ensuring your POC has a short time frame. So we've written this workbook to help you complete your POC in an ambitious but feasible six weeks. By following the lessons shared here, you'll be able to run a POC that is:

- Small enough that it can be completed in six weeks.
- Important enough that it can prove the value of big data to the rest of the organization.
- Intelligently designed so you can grow the project incrementally.

Let's start.

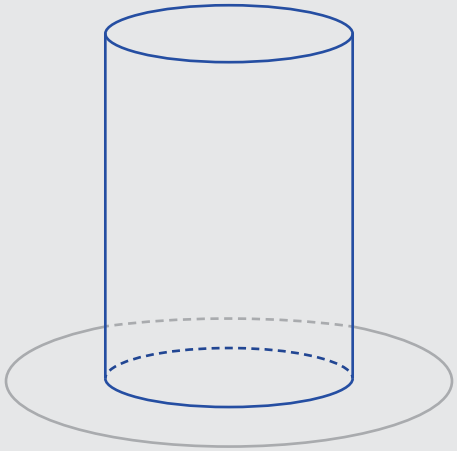
Drinking our own champagne

The marketing team at Informatica recently ran a successful POC for a marketing analytics data lake to fuel account-based marketing. So in addition to examples and stories from our experience with other companies, we'll also be sharing some of the lessons we learned along our own journey.

For a more detailed breakdown of why and how we built this data lake, check out our series 'Naked Marketing' written by Franz Aman, Informatica SVP of Marketing.

Part 1

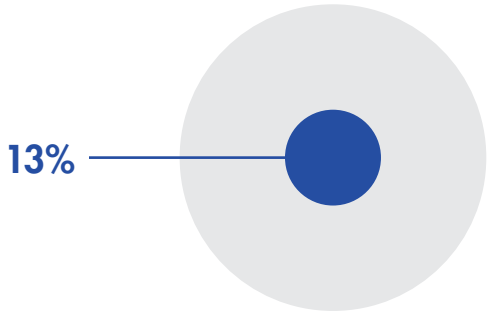
General Lessons.



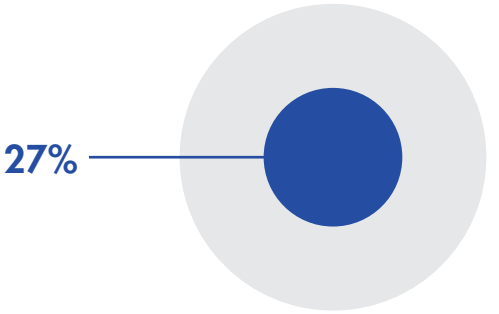
"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."

Abraham Lincoln

Why POCs Fail.



of organizations have big data projects in full-scale production



of executives deem their big data projects 'successful'

Far too often, POCs run over budget, fall behind schedule, or fail to live up to expectations. In fact, only 13 percent of organizations have big data projects in full-scale production[5]. And only 27 percent of executives deem their big data projects 'successful'[5].

So before we get into what it takes to make big data projects succeed, it's important to consider why so many POCs fail.

A lack of alignment with executives
The success or failure of your POC will depend on your relationship with the executives sponsoring the project. So rather than just occasionally reporting to the executive, it's important that you make them an active participant in the project.

That means understanding the outcomes they're interested in championing and making sure that what you build proves their value. In some cases, you may need to manage expectations and in others you may need to show them they can actually aim higher. In either case, it's important that they're treated like the crucial part of the POC process they are.

Without the active involvement of an executive, you cannot prove the value of a POC to the enterprise. And the POC can't be completed in six weeks.

Lack of planning and design
You don't want to spend three of your six weeks just reconfiguring your Hadoop cluster because it was set up incorrectly or for a different project. Even at an early stage, planning, architecture, and design are crucial considerations for big data implementations.

While you may not need your project to have a fully-designed architecture to prove value, (in fact with a time frame of six weeks it's actually wise not to build a complete architecture) you will need to understand the implications of various technical components.

Why POCs Fail.

And POC or not, a poorly rationalized technology stack with component technologies that aren't configured in a coordinated way is a recipe for disaster.

Scope creep
As excitement around the project grows, it's common for new business units, executives, and team members to add requirements and alter your priorities. In fact, as you go through the build and realize what's possible, you may be tempted to broaden the scope of your first use case too.

But scope creep invariably slows down deployments. And a small project that solves a business problem is a lot more valuable than a broad project that doesn't.

So it's crucial that your POC be tightly focused on the use case it's meant to solve in six weeks' time. That means focusing everyone involved on a single use case that's important enough to see through before you start building.

Ignoring data management
In all the excitement around new technologies like Hadoop and Spark, it's worryingly easy to forget the basic principles of data management needed to make big data work. It's this simple: if the data fueling your big data POC isn't reliable, connected, and easy to manage, it isn't successful.

And if you're hand-coding integrations, transformations, and business logic, you won't have the time you need to actually solve a business problem.

So rather than getting caught up in making hand-coded scripts for 'slowly changing dimensions' work in new technologies like Spark, it's a lot smarter to use tools you already know. More importantly, it isn't easy to scale, deploy, and maintain the code of a small group of experts. So automating crucial data management helps both during and after your POC's six-week time frame.

Big Data Management Lessons.

A phased approach to big data projects is crucial. But the success of a phased approach isn't just about starting small. It's also about having a cohesive vision to build toward. To that end, big data management has a crucial role to play in the success of big data projects for two reasons.

First, because big data management principles streamline a large part of the work that needs to be done during your first six weeks, allowing your developers to focus on the domain-specific problem in front of them.

And second, because once you prove your solution works in a POC environment, you need to be able to scale and deploy it to a production environment in a reliable, flexible, and maintainable way.

So when it comes to building out the solution architecture for your POC, consider the following big data management principles:

Abstraction is key

The key to scale is leveraging all available hardware platforms and production artifacts (such as existing data pipelines). So when it comes to data management, it's important that the logic and metadata you need is abstracted away from the execution platform.

That way, you make sure that no matter what changes in the runtime environment—for instance, if you realize you need to move from MapReduce to Spark—you can still preserve your integrations, transformations, and logic. It gives you the neutrality you need to manage the data and not the underlying technology.

Metadata matters

One of the biggest challenges with large datasets is the sheer amount of data 'wrangling' (cleansing, transforming, and stitching) that needs to occur for data to be usable. In some cases this can take up as much as 80 percent of a data scientist's time[6].

So rather than saddling your scientists with the burden of manually applying the same transformations over and over again, it makes a lot more sense to manage metadata in an accessible way. Global repositories of metadata, rules, and logic make it a lot easier to use a rationalized set of integration patterns and transformations repeatedly.

Big Data Management Lessons.

Leverage your existing talent
Thanks to technologies like Hadoop, it's become incredibly easy to add low-cost commodity hardware. At the same time, it's becoming harder to add the right talent. For instance, if you're using Spark as your runtime environment, you need developers who can code in specialized languages and understand the underlying computing paradigm.

So instead of investing in outside talent for every new skill and technology, it's smarter to leverage your existing talent to integrate, cleanse, and secure all your data using the data management tools and interfaces your team knows how to use. Make sure your data management tool can deploy data pipelines to big data platforms such as Hadoop, and specialty processing engines on YARN or Spark.

Automate to operationalize
Data management in a POC is hard. And data quality is fundamental to the success of a big data project. So while you might be able to complete a POC with a small team of experts writing a lot of hand coding, all it proves is that you can get Hadoop to work in a POC environment.

It's no guarantee that what you built will continue to work in a production data center, comply with industry regulations, scale to future workloads, or be maintainable as technologies change and new data is onboarded. The key to a successful POC is preparing for what comes next.

At this stage you need automation to scale your project. So keep the future scale of your initiative in mind as you plan and design the POC.

Security will be more than a checkbox
At the POC stage, data security is basically treated like a checkbox. That's because in a non-production environment with a small group of users, you might be able to manage risk without a clearly-defined security strategy.

But once you operationalize your project, you have to make sure your big data application doesn't turn into a data free-for-all. At scale, you need to be able to not only apply strict security policies centrally, but also to make sure enforcing them secures your data across users, applications, environments, and regions.

It's no mean feat. For example, do you know which data sets contain sensitive information? And can you de-identify and de-sensitize them in case of a security breach? Even though you may get away with minimal security now, it helps to consider and discuss the various threats to the security of the data that resides in your cluster.

Part 2

Picking the First Use Case.

Limiting the Scope.

The biggest challenge with a big data POC lies in picking the first use case. The key lies in finding a use case that can deliver an important business outcome—while still being small enough to solve within budget and in six weeks. (Note: we'd advise against having multiple use cases for a six-week POC.)

Once you've identified a business challenge you can solve with big data, minimize the scope of the project as much as possible. The smaller and more tightly focused your first use case, the greater the likelihood of success.

Of course, it isn't worth over-simplifying your project and missing out on crucial capabilities. But starting small and having a clearly defined roadmap and phased approach is key.

It's better to spend a few weeks refining the features and functions of your first use case down to the bare minimum required to prove value, before you start the actual POC. That way, you can ensure everyone's on the same page about what needs to be done and avoid scope creep later. And any requirements that fall outside the scope of your first use case can be re-considered during the next phase.

Limit the scope of your POC along these dimensions:

The business unit
In a six-week POC, it's smarter to serve one business unit rather than trying to serve the whole enterprise. For enterprise-wide initiatives, you typically need champions from multiple departments and within a six-week timeframe, it can be challenging to find and manage them.

The smaller the team of business stakeholders you need to engage and report to, the easier it'll be to meet and manage expectations.

The executive
As we've said, executive support is crucial to the success of your POC. But trying to cater to the needs of multiple executives takes much-needed focus and resources away from your six-week timeframe. As far as possible, make sure you're only working with one executive during your POC—and that you're working closely together to deliver the same vision.

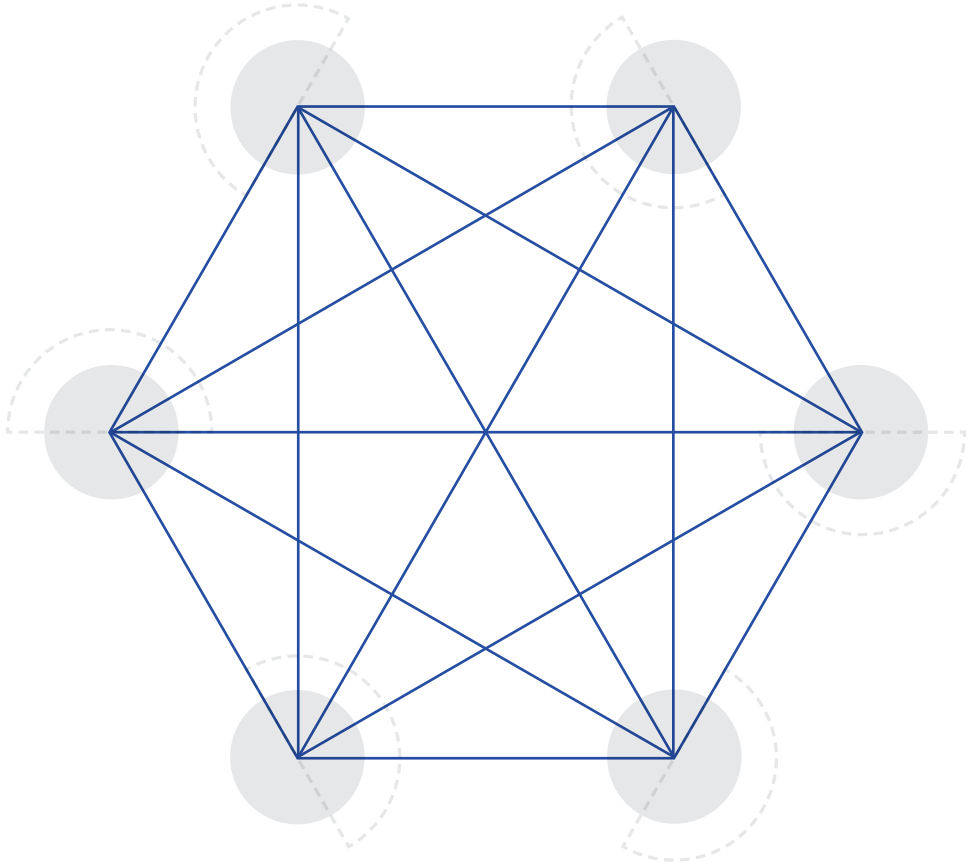
When the Informatica marketing team ran its POC for a marketing data lake, our Digital Marketing Analytics Manager Anish Jariwala found that working with one SVP of Marketing—Franz Aman—was actually crucial to completing the POC in time:

"If I had to work with three different Franz's, it would've taken us a year to deliver."

Engaging your Integration Competency Center (ICC)

A lot of the data management technology (from data quality tools to data pipelines) and standards (from business logic to glossary terms) you need may already exist within your Integration Competency Center or Integration Center of Excellence.

It makes sense to engage these experts and show them what you're working on. If your group and theirs decide to work together, you get the benefit of pre-built standards and tools. And even if you don't work together for the POC, you'll have established a relationship that will be useful when you operationalize your project.



Limiting the Scope.

When it came to building out the marketing data lake at Informatica, our team focused on only the most essential data sources needed to gain a consolidated view of how prospect accounts engaged with our website: Adobe Analytics, Salesforce, and Marketo.

Once we'd proven the value of this use case, we were in a position to add new sources such as Demandbase and ERP data, as well as other techniques such as predictive analytics.

The team
Effective collaboration between IT and business stakeholders is crucial if you're going to complete your POC in six weeks. So involve only the most essential team members needed to prove value in six weeks.

Depending on the nature of your project and the way your legacy stack is set up, this team is likely to consist of a domain expert, data scientist, architect, data engineer, and data steward—though one person may wear many hats.

The data
Aim for as few data sources as possible. Go back and forth between what needs to be achieved and what's possible in six weeks until you have to manage only a few sources. You can always add new ones once you've proved the value.

In terms of the actual data itself, focus only on the dimensions and measures that actually matter to your specific POC.

If your web analytics data has 400 fields but you only need 10 of them, you can rationalize your efforts by focusing on those 10.

Similarly, if you're attempting to run predictive analytics based on customer data, it might be better to validate a model based on a smaller but indicative subset of attributes so as to not over fit your analytic models.

Cluster size
Typically, your starting cluster should only be as big as five to 10 nodes. We recommend you work with your Hadoop vendor to determine the appropriate size of cluster needed for your POC. It's smarter to start small during the POC phase and then scale your development and production clusters since Hadoop has already proven its linear scalability.

And the six weeks you have for your POC should really only serve to prove your solution can solve your domain-specific business challenge.

Sample Use Cases.

Now let's take a look at some use cases from successful POCs that limited their scope to prove value and grow incrementally.

Business Use Cases:

Fraud Detection in Hadoop

Problem: One U.S. bank found that managing its general ledger data on a mainframe was proving to be far too expensive. Even worse, it made it incredibly time-consuming to use that data for risk analytics.

Use case: The team took the general ledger data off the mainframe and stored it in Hadoop to reduce costs. This allowed them to identify anomalies and trends with a new and better approach to fraud detection.

Customer 360 Analytics

Problem: Disparate data sources and a fragmented view of customers across systems made it impossible for marketers at a global payment services company to detect trends and patterns in their customer data.

Use case: The team decided to process a subset of customer data for a customer 360 initiative that would allow them to analyze and predict purchasing behavior and consumer preferences across different channels.

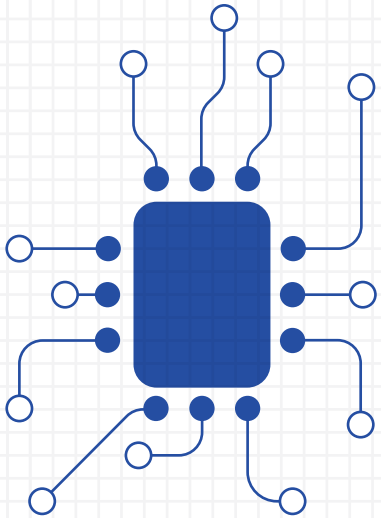
A Marketing Data Lake for Account-Based Marketing

Problem: The Informatica marketing team lacked a consolidated view of how known and unknown prospects from target accounts interacted with the Informatica website and content.

Use case: The team integrated site traffic data from Adobe Analytics with lead data from Marketo and Salesforce, as well as reverse-IP lookup from Demandbase to understand what users within specific accounts were looking for.

Looking forward: Now that the team has proven the value of both the solution architecture and its account-based approach to marketing, we're extending its capabilities to include cohort analysis, attribution modeling, and identifying influencers.

Sample Use Cases.



Technical Use Cases

In some cases, IT teams have the resources and authority to invest in technical use cases for big data. This may be to identify technical challenges and possibilities around building an enterprise data lake or analytics cloud, for instance.

It's important to note that while we've been describing the first phase of a big data project as a Proof of Concept we're really talking about a Proof of Value. The difference being that while a POC aims to demonstrate the capabilities of features and functions in big data technology, a Proof of Value goes one step further and demonstrates the business impact.

Even if you have the license to explore big data, it's still important that you have a clear view of the value you need to demonstrate for business stakeholders.

Some of the most common technical use cases are:

Data warehouse extensions

- Building a staging environment to offload data storage and ETL workloads from a data warehouse.
- Extending your data warehouse with a Hadoop-based ODS (operational data store).

Building data lakes and hubs

- Building a centralized data lake that stores all enterprise data required for big data analytic projects.
- Building a centralized data hub with a logical data structure over Hadoop to perform aggregations and transformations on the fly.

Building an active data archive

- Managing archived data with full data lineage to increase auditability.
- Managing all archived data in a single platform to make it easier to search and query.

Part 3

Proving Value.

Defining the Right Metrics.

A successful POC needs to prove a number of different things:

- That the big data project will deliver a measurable return.
- That the technology stack used for the project will help business stakeholders achieve that return in a reliable, repeatable way.
- That the solution architecture is worth further investment once you start to operationalize the project.

The starting costs for big data POCs can be misleadingly low and costs can spiral once you operationalize. So your POC needs to have comprehensively proved its value before this point.

To that end, even before your POC has started, it's important that you begin translating expectations of success into clearly defined metrics for success.

Based on your first use case, list out as many of the most appropriate metrics for success as you can. If the metrics you're listing aren't likely to prove this is a project worth further investment, it's worth redesigning your project to achieve more relevant metrics.

Quantitative measures (examples)

- Identify 1,200 high revenue potential prospects within one month.
- Identify anomalies in general ledger data 30 percent faster than in the enterprise data warehouse.
- Make customer product preferences available within top-fold of Salesforce interface for 20,000 leads.

Qualitative measures

- Agility: Allow marketers to ask more sophisticated questions such as who are my most influential customers.
- Transparency: Improve visibility into provenance of atypical data entries leading to better identifying fraudulent patterns.
- Productivity and efficiency: Reduce number of man-hours spent by data analysts on stitching and de-duplicating data by 50 percent.

Capturing Usage Metrics to Prove Value

One big indicator of success for Informatica's marketing team was the number of people actually using the tools they created. By recording usage metrics, they could prove adoption of the solution by field marketers and sales development representative to make sure they were on the right track.

Additionally, it helps to analyze why more users are flocking to your new interfaces and tools. For instance, how much faster can sales qualify a new opportunity compared to previous approaches? Are lead conversions increasing? What fields can sales development representatives see now that they have a consolidated view of different systems?

SL No	Job Title	View ID	View Name	Last Viewed	Viewed
1	Product Marketing	1096	Dashboard	12/17/15 14:02	71
2	Sales Director	1096	Dashboard	01/04/16 11:07	217
3	SDR	1096	Dashboard	12/21/15 11:06	1
4	Team leader	1096	Dashboard	12/10/15 12:19	40
5	Team leader	1096	Dashboard	10/26/15 7:34	7
6	VP of Sales	1096	Dashboard	11/03/15 12:39	42
7	Field Marketing	1096	Dashboard	01/04/16 14:20	322
8	SDR	1096	Dashboard	10/08/15 11:29	15
9	Field Marketing	1096	Dashboard	12/17/15 11:30	232
10	Solutions Director	1096	Dashboard	01/04/16 12:40	24
11	VP of Sales	1096	Dashboard	12/17/15 13:38	235
12	Solutions Director	1096	Dashboard	01/04/16 07:27	16

Part 4

Getting Technical.

The Three Layers of a Big Data POC.

For the most part, the hype around big data technologies has focused on either visualization and analysis tools like Tableau and Qlik, or distributed storage technology like Hadoop and Spark.

And for the most part, the hype has been justified. Visualization tools have successfully democratized the use of data in enterprises. And Hadoop has simultaneously reduced costs, improved agility, and increased the linear scalability of enterprise technology stacks.

But big data projects rely on a crucial layer between these two—the data management layer. Without it, data isn't fit-for-purpose, developers get mired in maintaining spaghetti code, architectures become impossible to scale for enterprise-wide use, and data lakes become data swamps.

So when it comes to building out the solution architecture for your POC, it's important you consider all three of the following layers:

Visualization and analysis
Includes analytic and visualization tools used for statistical analyses, machine learning, predictive analytics, and data exploration.

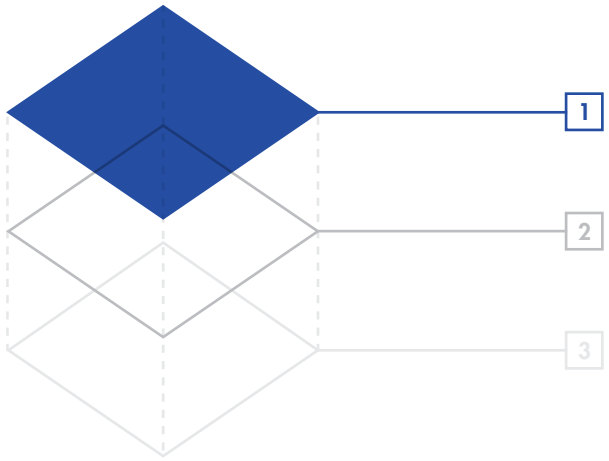
Here, user experience is key. Business users, analysts, and scientists need to be able to quickly and easily query data, test hypotheses, and visualize their findings, preferably in familiar spreadsheet-like interfaces.

Big data management
Includes core tools to integrate, govern, and secure big data such as pre-built connectors and transformations, data quality, data lineage, and data masking.

Here, the emphasis should be on ensuring data is made fit-for-purpose and protected in an automated, flexible, and scalable way that speeds up your build.

Storage persistence layer
Includes storage persistence technologies and distributed processing frameworks like Hadoop, Spark, and Cassandra.

Here the emphasis should be primarily on cost reduction and leveraging the linear scalability these technologies enable.



Should Big Data POCs Be Done On-Premise or in the Cloud?

Just over half (51 percent) of Hadoop clusters are run on-premise today[7]. That's a surprising amount considering Hadoop's association with the rise of cloud computing.

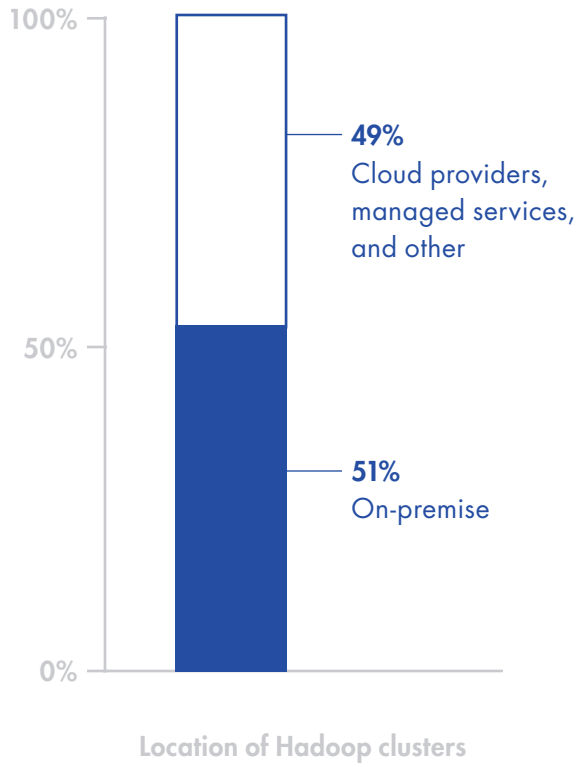
But it does raise an interesting question about whether or not you should run your cluster in the cloud or on your own hardware.

On the one hand, cloud storage is cheaper, offers elasticity, and makes it easier to access commodity hardware. In fact, cloud vendors have the resources and expertise to actually provide significantly better security than most companies can afford.

At the same time, you have to be careful about your organization's standards and mandates, as well as perceptions about the cloud.

Additionally, if you've already sunk the costs of hardware and have the staff for 24/7 maintenance, it might actually be easier to get started on your own premises.

In either case, it's clear that as projects scale and evolve, you'll likely change the runtime environment under your stack. So make sure you manage your metadata strategically and abstract away transformations, logic, and code so it's easier to deploy elsewhere.



The Three Pillars of Big Data Management.

When it comes to building the big data management layer of your POC, you need your stack to account for three key areas:

1. Big Data Integration
The first challenge with any big data project is about ingesting the data you need. Big data means huge volumes of data from multiple data sources in multiple different schemas. As a result, you need high-throughput data integration and, if necessary, real-time streaming for low-latency data.

In the more specific context of your six-week POC, it's important that you aim for the right things as early as possible.

- Leverage pre-built connectors, transformations, and integration patterns to speed up the development of your data pipelines. They'll make it easier for analysts and developers to connect to key sources and ensure no one has to reinvent the wheel every time they need a new connection.
- Versatility is key. Once you've completed your POC, you'll need to be able to grow your platform incrementally. Aim for universal connectivity across data sources, data types, and dynamic schemas so you can onboard new types of data quickly.
- Abstraction matters. As we described in Part I Section 2, it helps to abstract your code and logic away from the runtime environment. That way, no matter what changes you make to your underlying technology stack, you still have a global repository of patterns and transformations to use.

2. Big Data Quality (and Governance)
Without fit-for-purpose data, it doesn't matter how sophisticated your models and algorithms are. It's essential that you proactively manage the quality of your data and ensure you have the tools you need to cleanse and validate data.

At the scale of big data, managing data quality is all about maximizing IT's impact with consistent standards and centralized policies for effective governance.

It's about making sure you can:

- Detect anomalies quickly: You can't eyeball all your data. And you can't attempt to manually correct every error. For this to be scalable, you need to be able to track and monitor data quality with a common set of rules, standards, and scorecards acting as a frame of reference across the environment.
- Manage technical and operational metadata: A global metadata repository of data assets, code, and logic about transformations enables every user to easily find datasets and deploy the cleansing and validation they need. Most important, it gives you the ability to define those rules once and govern anywhere.

The Three Pillars of Big Data Management.

- Match and link entities from disparate data sets: The more data you're managing the harder it becomes to infer and detect relationships between different entities and domains. By building in entity matching and linking into your stack, you're making it easier for your analysts to uncover trends and patterns.

- Provide end-to-end lineage: An important function of a global repository of metadata is to make it easy to uncover the provenance of data. For one thing, this streamlines compliance processes because it increases the auditability of all your data. But it also makes it easy to map data transformations that need repeating.

3. Big data security

Data security can be fairly light during POCs. But the second your project succeeds, security becomes a whole other ball game. This is particularly challenging with Hadoop implementations where it's hard to monitor and control different users from different teams and even regions.

For instance, if you have a team in India accessing certain data that was generated in California, you need to make sure you have the tools in place to control what they can and cannot see.

While it may not directly apply to your POC project, it's important to start thinking (and talking) about your security strategy for a production environment. At scale, you need to be able to:

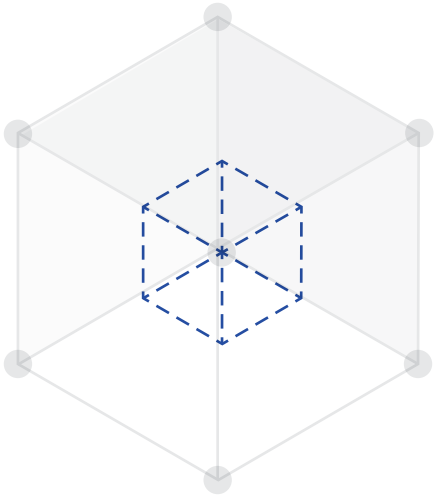
- Monitor sensitive data wherever it lives and wherever it's used: That means risk-centric security analytics based on a comprehensive 360 degree view of all your sensitive data.

- Carry out risk assessments and alert stakeholders: Security breaches occur all too frequently. So you need to make sure you clearly understand where your sensitive data resides, who's using it and for what purpose. Risk scorecards based on trends and anomalies are essential to detect exposure to breaches when they happen.

- Mask data in different environments to de-identify sensitive data: Encryption and access control can only go so far. You don't want your security controls to be prohibitive and intrusive. So it makes a lot more sense to use persistent and dynamic data masking to make sure different users can see the data they need, while never exposing the data they don't.

Four Dimensions of Big Data Security

When most of your data no longer lives behind the firewall, you can't afford to rely on authentication and authorization alone for your big data security strategy. In fact, your big data security should ideally encompass four types of protection:



1. **Authentication & Authorization**
   Tools like Kerberos, Knox, and Sentry are essential for perimeter control when multiple users are accessing your cluster.

2. **Encryption**
   Initiatives like Intel's Project Rhino allow you to encrypt the data living in your data stores and manage keys. This is crucial when you need to ensure outsiders can't access sensitive data but insiders need to be able to decrypt and access the same data. And format preserving encryption can be used to get as granular as the field level.

3. **Tokenization**
   This is all about substituting sensitive data elements with non-sensitive equivalents or 'tokens'. It allows you to preserve data types and formats and protect against stolen keys.

4. **Data masking**
   To persistently or dynamically protect data from certain users (e.g. protecting credit card data in dev and test environments), data masking can be used to de-identify data while still ensuring it looks like the original data. Unlike with encryption, the 'masked' data element can't be decrypted by insiders to reveal the sensitive information being protected.

A Checklist for Big Data POCs.

I. Visualization and Analytics

1. List the users (and their job roles) who will be working with these tools.

2. If you're aiming for self-service discovery, list the types of visualizations your users need (e.g. bar charts, heat maps, bubble charts, scatter plots, tree maps).

3. What types of analytic algorithms will you need to run (e.g. basic or advanced statistics, predictive analytics, machine learning)?

4. If you're aiming for reporting or predictive analytics, which tools will you be deploying?

5. Do your tools need to access multiple systems? If so, how will they access all of these systems?

6. What is the format of the data you will deliver to the visualization and analytics tools?

A Checklist for Big Data POCs.

II. Big Data Integration

1. Will you be ingesting real-time streaming data?

2. Which source systems are you ingesting from?

3. Outline the process for accessing data from those systems (Who do you need to speak to? How frequently will the data be transferred? Will you just capture changes or the entire dataset?)

4. How many data integration patterns will you need for your use case? (Try to get this number down to between two and three.)

5. What types of transformations will your use case call for (e.g. complex file parsing, joins, aggregation, updates)?

6. What type of data (e.g. relational, machine data, social, JSON, etc.) are you ingesting? (Review sample data in the required format.)

7. How many dimensions and measures are necessary for your six-week timeframe?

8. How much data is needed for your use case?

9. How will you infer join keys for disparate datasets?

A Checklist for Big Data POCs.

III. Big Data Quality and Governance

1. Will your end-users need business-intelligence levels of data quality?

2. Alternatively, will they be able to work with only a few modifications to the raw data and then prepare the data themselves?

3. What tools are available for end-users to do their own data prep?

4. Will you need to record the provenance of the data for either documentation, auditing, or security needs?

5. What system can you use for metadata management?

6. Can you track and record transformations as workflows?

7. Will your first use case require data stewards? If so, list out their names and job titles.

8. How will you match entities (e.g. customers) between datasets from different systems?

A Checklist for Big Data POCs.

IV. Big Data Security

1. Do you know which datasets contain sensitive data and on which systems they reside?

2. Will you need different data masking rules for different environments (e.g. development and production)?

3. Which techniques will you use to protect sensitive data (masking, encryption, and/or tokenization)?

4. Have you set up profiles for Kerberos authentication?

5. Have you set up your users to only access data within your network? What datasets will they have access to?

6. What security policies do you need to work within your POC environment?

A Checklist for Big Data POCs.

V. Big Data Storage Persistence Layer

1. What systems will you use to store your data (e.g. data warehouse, Hadoop, Cassandra, or other NoSQL platforms)?

2. What type of data do you plan to store in each of these systems? How do you plan to share data between these systems if there is more than one?

3. Will you be using Hadoop for pre-processing before moving data to your data warehouse?

4. Or will you be using Hadoop to both store and analyze large volumes of unstructured data?

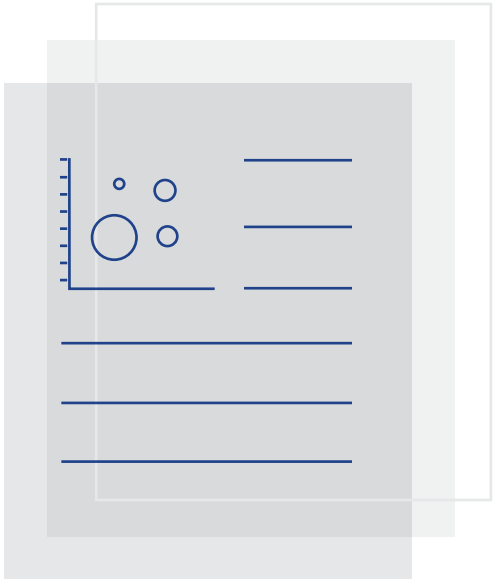
5. Which Hadoop sub-projects will you be using (e.g. Hive, HBase, Kafka, Spark)?

The Importance of Documentation

Documentation is vital, especially if your POC is successful. You'll need to be able to share the details of it once you decide to expand the scope of your project. You'll also benefit from being able to socialize your new capabilities, what they enabled, and why they matter.

Instead of trying to reverse-engineer the documentation you need after the project, it makes sense to document your progress along these lines:

- **The goals of your POC** as defined by the desired business outcomes.
- **The success of your POC** as defined by your metrics for success.
- **How data flows between different systems;** crucial transformations; mappings; integration patterns and end-points.
- **Your solution architecture** and explanations about important design choices the team made along the way.
- **Development support:** It should be said that with the right selection of tools, you can minimize the amount of code that needs to be documented.



A Solution Architecture for Big Data Management.

No two POCs will have the same solution architecture. But when it comes to determining what technology you need for your big data management stack, it's important to consider all the necessary layers.

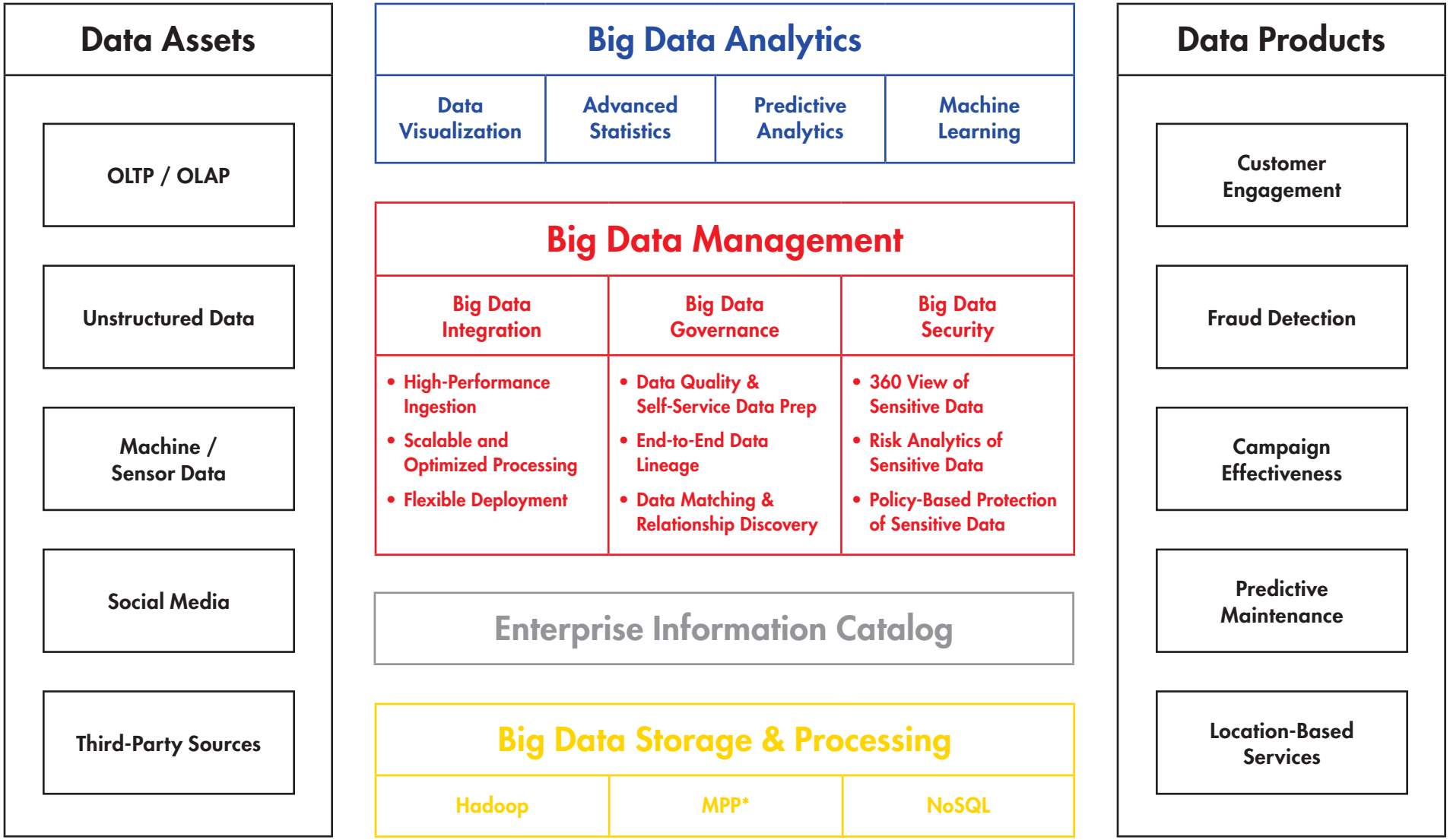
In the following solution architecture, we've used a general set of technologies worth considering that span:

- The visualization and analytics layer
- The big data management layer
- And the data storage layer

Additionally, you'll notice that we've also included a horizontal layer to represent the necessary universal metadata catalog we've described throughout this workbook.

Use this reference architecture, along with the checklist from the previous section to plot out your technology needs for an efficient, reliable, and repeatable big data infrastructure.

Big Data Reference Architecture



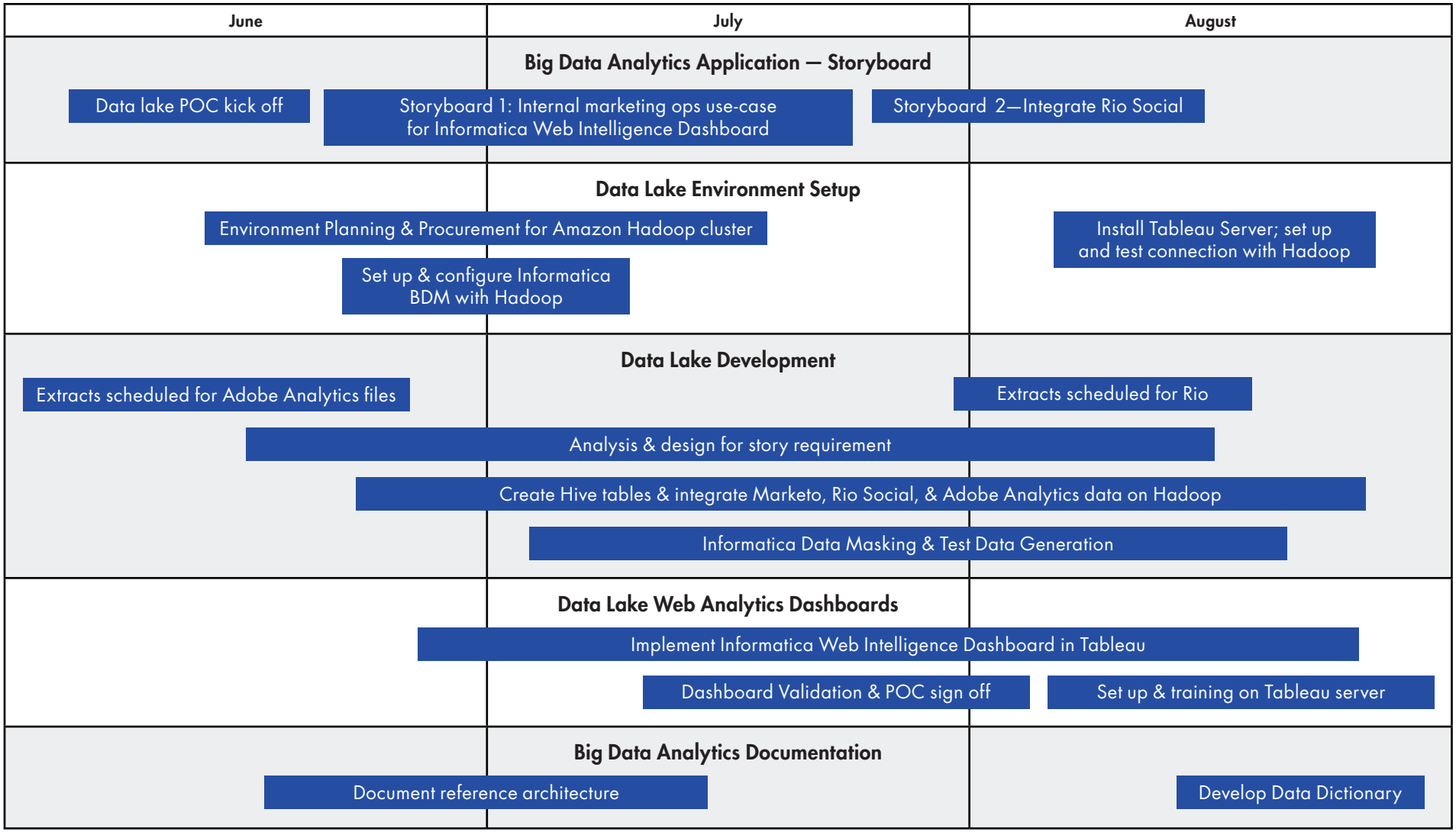
*Massively Parallel Processing

A Sample Schedule for a Big Data POC.

The key to completing a POC in six weeks is having a schedule that prioritizes and organizes everything you need to do. The specific components and layout of your schedule will depend on the nature of your own POC.

But consider the following schedule from the Informatica marketing team's data lake project for an idea of how to structure your efforts. Bear in mind, the timeline for this deployment was two months so your six-week POC will need to be carried out in a slightly tighter timeframe.

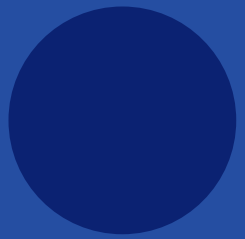
Big Data Analytics POC Schedule



For more context about the details of this project, read our blog series 'Naked Marketing'.

Conclusion

Making the Most of Your POC.



Making the Most of Your POC.

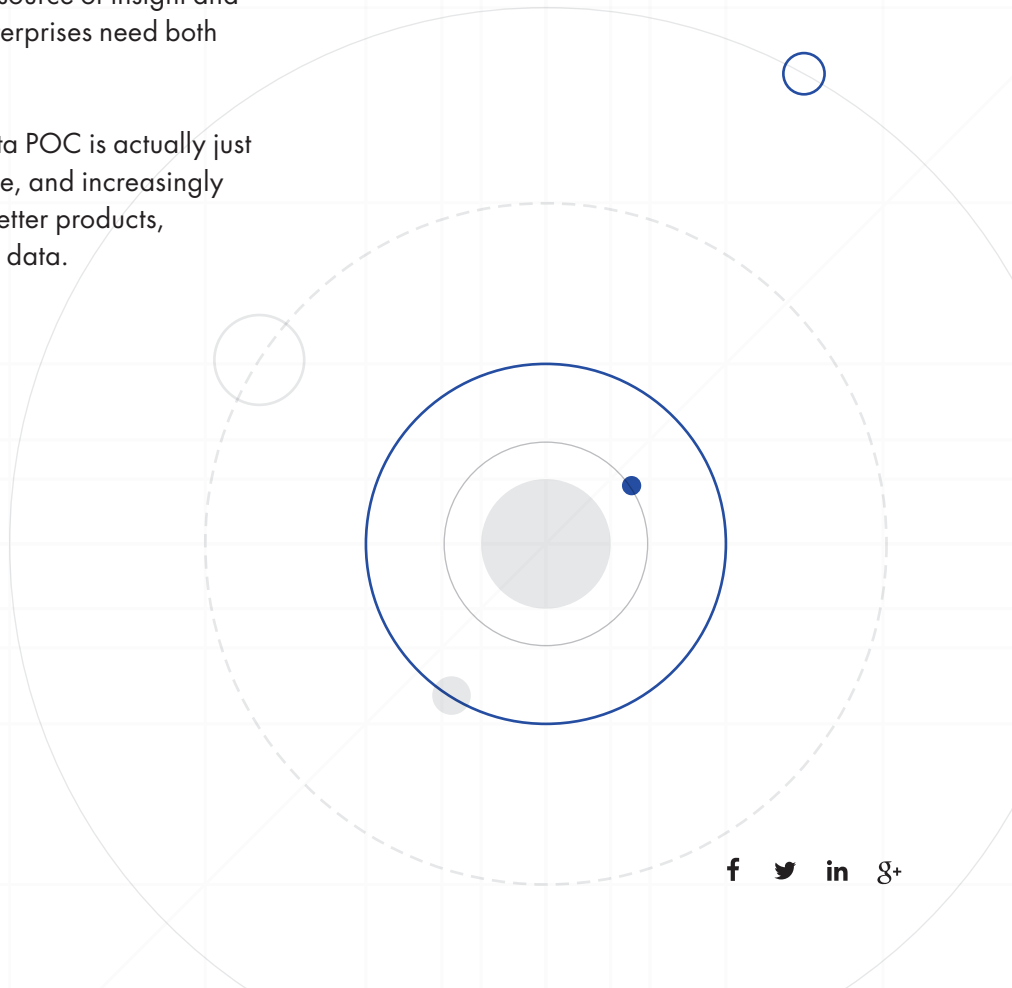
At the start of this workbook, we explained that our aim was to help you build and run a POC that is:

- Small enough that it can be completed in six weeks.
- Important enough that it can prove the value of big data to the rest of the organization.
- Intelligently designed so you can grow the project incrementally.

We hope the advice, lessons, and examples we've shared in this workbook help you achieve all these things. But more than that, we hope they can help you kick start a bigger, more important journey for your company.

Successful POCs don't just prove the value of a specific solution architecture. They prove the value of big data as a viable source of insight and innovation at a time when enterprises need both more than ever.

So really, a successful big data POC is actually just the start of a constant, iterative, and increasingly interesting journey towards better products, smarter analyses, and bigger data.



Sources

1. **Forbes**, 84% of enterprises see big data analytics changing their competitive landscape, 2014

2. **Data Science Central**, The amazing ways Uber is using big data, 2015

3. **Forbes**, How Airbnb uses big data and machine learning to guide hosts to the perfect price, 2015

4. **Fast Company**, The world's top 10 most innovative companies of 2015 in big data, 2015

5. **Capgemini**, Cracking the data conundrum: How successful companies make big data operational, 2015

6. **New York Times**, For big data scientists, janitor work is the key hurdle to insights, 2014

7. **TDWI**, Hadoop for the Enterprise: Making data management massively scalable, agile, feature-rich and cost-effective. Second Quarter, 2015