DOMINO

WHITEPAPER

The Practical Guide to Managing Data Science at Scale

Lessons from the field on managing data science projects and portfolios

Contents

Executive Summary

The ability to manage, scale, and accelerate an entire data science discipline increasingly separates successful organizations from those falling victim to hype and disillusionment. While data scientists may have the sexiest job of the 21st century, data science managers probably have the most important and least understood job of the 21st century. This paper aims to demystify and elevate the current state of data science management. We identify consistent struggles around stakeholder alignment, the pace of model delivery, and the measurement of impact. The root cause of these challenges can be traced to a set of particular cultural issues, gaps in process and organizational structure, and inadequate technology.

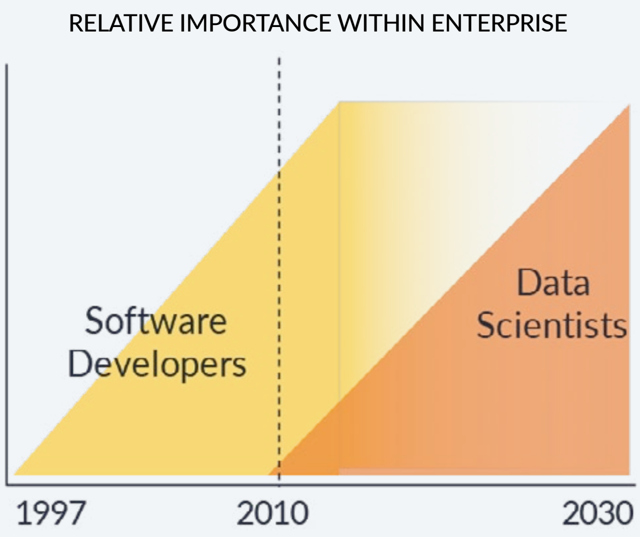
Based on 4+ years of working with leaders in data science, like Allstate, Monsanto, and Moody's, we have observed that the best solution is a holistic approach to the entire project lifecycle from ideation to delivery and monitoring. Organizations that are able to develop a disciplined practice of iterative business value delivery and self-measurement, while utilizing data science platform technology to support a hub-and-spoke organizational structure, can scale data science to a core capability, and accelerate the delivery of a robust portfolio of models. While a complete transformation can take years, we suggest a "crawl, walk, run" approach to build momentum towards the ultimate vision.

Introduction
Where we are today and where we came from

Managing data science is an emerging discipline. It will be critical to achieving the transformational impact anticipated, and increasingly demanded, by business leaders. There is legitimate promise, with companies like Google, Facebook, and Amazon building defensible businesses around the breadth and quality of their models.

At the same time, data science success can be perceived as a dark art to those not familiar with the practice. The pervasive hype has only exacerbated the situation as the risk of disappointment and disillusionment grows. Data science leaders are being asked to 10x their impact without a clear roadmap of how to scale their current capabilities.

Data science is in the throes of a transition from a niche capability leveraged by a few pioneers to a core capability across every enterprise. What was once a "nice to have" has become a survival imperative. As with the evolution of software development, the tooling has advanced dramatically in recent years. But, also like software development, tooling alone is not enough. The hardening of new roles (i.e., people), processes, and technology will be key to cementing data science's position as a core function.



In our experience working with data science organizations large and small, we've seen recurring themes in the management challenges they face; the underlying cause of those challenges; and a set of principles and best practices that have enabled success at scale.

This guide shares market learnings that will help rapidly scaling data science organizations avoid the pitfalls of those who tread before them and deliver business impact. For those already operating at scale, we offer insights on how to measure and optimize your portfolio of models to articulate business impact and cement data science as a core organizational capability.

The intended audience is any current or aspiring data science leader who aims to better understand the current state of play and come away with tangible best practices. The paper has the following structure:

> **1. Goals**
> What are the measures of a high-performing data science organization?
>
> **2. Challenges**
> The symptoms leading to the dark art myth of data science
>
> **3. Diagnosis**
> The true root causes behind the dark art myth
>
> **4. Project Recommendations**
> Managing a data science project to a business outcome
>
> **5. System Recommendations**
> Scaling a good data science project to a business discipline

Each chapter builds on the prior chapter to give adequate context for the recommendations at the end. Note, the final two chapters are more detailed and tangible, representing an aggregated view of actual principles and practices observed in the field. Feel free to jump to those chapters if you're in a hurry, but realize that you'll be missing some context.

1

Goals
What are the measures of a high-performing data science organization?

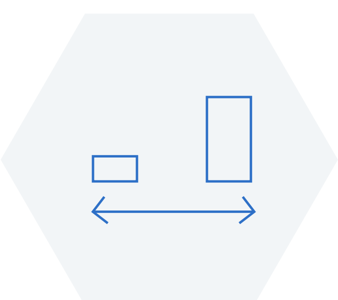
Data science as a widespread organizational capability has not existed long enough to have universally agreed upon measures of success. Typically, data science teams are caught between sharing the goals of their business stakeholders and some nebulous sense of "customer satisfaction" for more centralized, shared service teams. Based on our conversations with leading data science organizations, we believe there are key criteria that should form the cornerstone of any data science team's objectives: measurable, reliable, and scalable impact on the business decisions and metrics that they are charged with improving.

Measurable

Your "quality" indicator. Were business decisions positively changed in an observable and, ideally, quantifiable way? This is easier in certain situations (e.g., automated scoring engines) and more challenging in others (e.g., ad hoc executive analysis). Best practices on measuring impact are discussed in Chapters 4 and 5.

Reliable

Your "hit rate." If I take on five projects, I want 3-4 to deliver business value. An unreliable team will rarely have meaningful impact from their projects. Alternatively, a team with 100% reliability is probably being too conservative and not taking on enough audacious projects.



Scalable

Your "throughput." If my reliability is 80% with five projects and seven people in a quarter, can I expand that to 50 projects and 40 people without a significant degradation in my reliability?

If these are the goals, why are we often struggling to meet them? We discuss that in the next section.

2

Challenges
The symptoms leading to the dark art myth of data science

Data science leaders face a myriad of obstacles on the path to achieving their goals. Everyone talks about data being the problem. "Data scientists spend 80% of their time finding and cleaning data" goes the refrain. We agree that data is a big pain point. But it's a convenient scapegoat that distracts from some of the industry's real problems that can be solved with relative ease, getting you closer to your goals. Below we walk through some of the most common struggles in data science management.

Solve the wrong problem. Over-zealous data science teams often dive straight into the data looking for "something interesting." We've seen large organizations hire 30+ PhDs without clear business alignment upfront. They then emerge from a six week research hole only to realize they had misunderstood the target variable, rendering the analysis irrelevant.

Solve the right problem, but didn't realize it had already been solved or can't reproduce it. Data scientists consistently complain about re-inventing the wheel. Anecdotal estimates put it at 30-40% of their time in large organizations with significant amounts of prior art. In the fortunate situation where a past project can be discovered, reproducing it is often impossible given inconsistent preservation of relevant artifacts like data, packages, documentation, and intermediate results.

Try to solve the right problem but have the wrong tools. Given the explosion of data and tooling functionality, data scientists are still often dramatically ill-equipped to explore the full range of possible domains and solutions. Analysis is still often confined to individual laptops that are easily overwhelmed. We've heard of organizations where it can take 6+ months to approve a widely-utilized open source Python package for research purposes, prompting employees to bring their personal laptops and work under their desks. Furthermore, some organizations have the "coffee experiment" approach which entails running an individual experiment and then going to get coffee as they don't have sufficient computing resources.

> Data scientists spend 80% of their time cleaning data... it's a convenient scapegoat that distracts from some of the industry's real problems that are more easily solvable.

Solve the right problem too slowly for it to matter. We've seen data scientists who will spend an extra two weeks to eek out a bit more AUC on a targeting model, only to realize the marketing team's deadline passed.

Solve the right problem the wrong way, so no one uses it. In one case, we've seen a team build a powerful predictive model for underwriters, wrap it in a standalone scoring front end and realize the underwriters never clicked to the new browser tab from their existing workflow. A data science manager at the large insurer described the problem, saying, "We don't fail because of the math… we fail because we don't understand how people will use the math."

Solve a problem, but don't realize model is being used to solve a totally different problem. Google describes this as the undeclared consumer problem. Results can be thrown "over the fence" and data science teams have little control or even visibility into how those results are used. For example, someone builds a model for predicting the value of California residential mortgages. Then an over-zealous banker uses it to predict the value of Florida commercial mortgages, even though the original model creator knew that would be a bad idea.

Solve the right problem, the right way, but the world changes underneath you. Models are by definition an approximation of the real-world. If you don't keep track of how the world is changing and monitor your models' ongoing performance, you imperil the business and likely leave value on the table. Mission-critical models drift all the time for any number of reasons.

In one situation, a financial firm issuing credit cards expected nulls in their probability of default model if an applicant was not present in the database (typically a sign of elevated risk). The credit agency changed to reporting "no hits" as a "999" code, which was interpreted by the model as a high credit score, leading to a surge in approvals and millions in loan losses.

Solve a few good problems, but not solving enough problems at once. Many teams have had early wins from their low hanging fruit. Working in a tight-knit team on a single business initiative is great. However, teams start to experience negative returns to scale as their existing processes can't cope with a swollen backlog, an influx of new hires, and heightened expectations from the business.

Diagnoses
The true root causes behind the dark art myth

Rigorous problem solving and root cause analysis is a hallmark of good data science. To be successful, it is important to turn our analytical gaze inwards. In Chapter 1 we proposed the generic goals of a high-performing data science team. In Chapter 2, we identified the most common clusters of sub-optimal outcomes observed vis a vis those goals. In this chapter, we propose reasons for those sub-optimal outcomes. The point is not to cast blame, but rather identify root causes of issues, with an eye towards best practices of a well-functioning data science machine.

People and Culture

Stakeholders as an afterthought. Many data science managers spend 80-90% of their time internally focused, often having risen from once being a "hands on keyboard" practitioner themselves. This limits the amount of time spent managing key stakeholders who are critical to the data science organization's success. IT, business consumers, and executives all have a critical role.

Limited culture of introspection. As a discipline, data science remains young (and poorly defined). It's not that surprising that if we can hardly define what a data scientist is, we struggle to measure ourselves and guide internal investment based on our learnings.

Missing some key personnel muscles. The full stack data scientist is dead, if she ever existed at all. The move towards specialization isn't just data engineers, it's a whole host of other roles that cover the concepts of change management, feasibility assessments, rapid prototyping, ROI estimation, training, and stakeholder education. Data science training often focuses on the technical skills, which are necessary but insufficient for driving impact. Increasingly, the role is being partitioned into many roles, as happened with software development over the last two decades.

Data science as a moonshot vs. laps around the track. Many organizations have not built a culture of delivery and iteration. This could be a result of many data scientists' extensive academic backgrounds. It also stems from a confusion between what type of work is really happening: "pure research" and "applying templates to novel business situations." Very few organizations (e.g. large tech companies) are doing significant pure research. Most organizations are applying tried-and-true methods to a unique set of data and stakeholders.

Process and Organizational Alignment

The data science process is different from software development. This is a deep topic and merits its own detailed paper. Key differences should be called out as organizations often struggle because they try to graft their data science process directly onto the software development lifecycle (SDLC). First, software engineering involves building things where what you're building (though not how to build it) is fairly well understood.

Much of data science, however, is fundamentally a research process. A team can try 99 experiments before the 100th yields an interesting insight, or alternatively, there could be no insight at all, which is also a valid outcome.

Disconnected from the business. Teams are often hired into disconnected Innovation Labs without real business accountability to hone their process. This also means they don't have a deep understanding of the target KPIs and the nuances of how a team works today which is critical to ensuring adoption of their results.

Artisan thinking vs. modular system thinking. Data scientists often think of their work as bespoke and highly specialized. While their skillset may be, there are often many intermediate and final artifacts they create that can and should be reused. Whether those are software packages, modeling datasets, features, or anything else. Moreover, many data scientists barely document their development process, much less abstract and modularize the process.

Suboptimal organization and incentive structures. Many data scientists have told us "I get paid for what I build this year, not maintaining what I built last year." Similarly, IT scoffs at the idea that data scientists could write production level code. That leads to huge gaps in monitoring live production models. While IT is responsible for system performance including monitoring live production models, IT doesn't understand if the model is "still right" or being used appropriately. Furthermore, federated IT and data science functions can build shadow IT that create unknown dependencies between production models. No one person or group is responsible for the entire system, leading to sizable operational risk.

Technology and Tooling

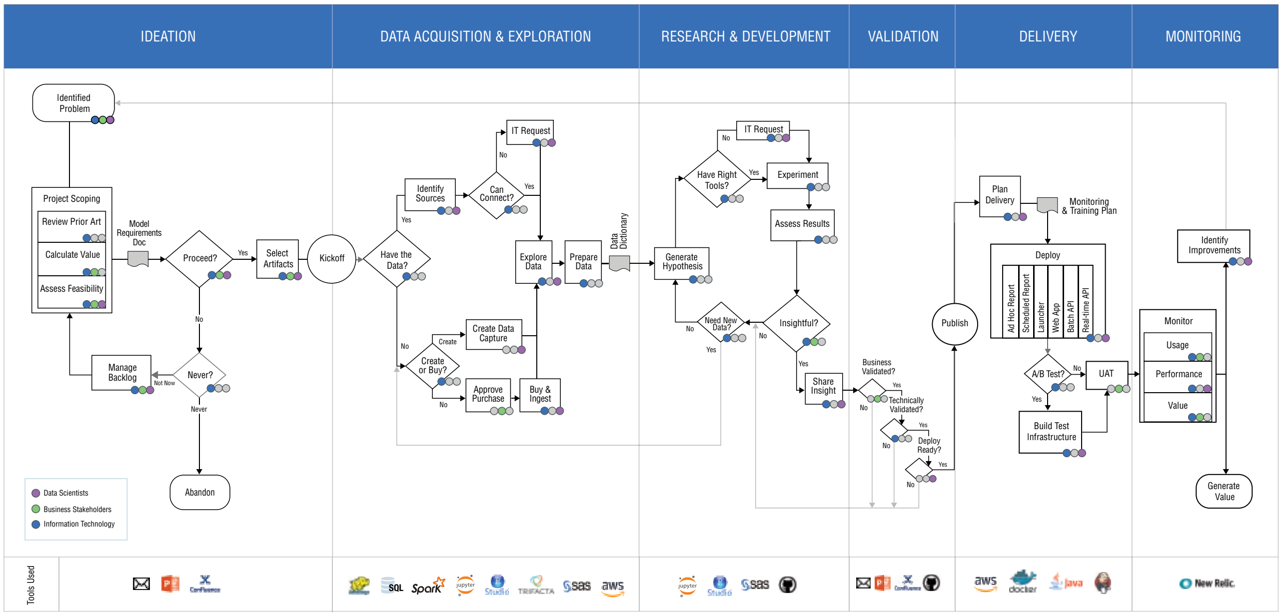
Traditional tooling has limited efficacy. With software, there is a notion of a "right" answer and it's possible to write automated tests that verify intended behavior. This doesn't hold for data science work. There is no "right" answer, only better or worse answers as evaluated by some target outcome. Rather than writing unit tests, data scientists visually inspect outputs, then obtain feedback from technical and subject matter experts to gauge the performance of their model. This has significant implications for validating a model and monitoring its ongoing use. In fact, often unintended results from experiments are the genesis of a new model that drives business value for different problem.

Fixation on tools. The explosion in tooling has, for the most part, been a blessing to the industry. However, it has created a culture of "silver bullet" thinking where some data scientists believe just getting Spark Streaming and TensorFlow will solve all their problems. This has carried over into the community. Reddit blogs on the optimal data science organizational structure don't get the same traction as throwdowns about Python and R. In many cases, data scientists' wear their tool wrangling as a badge of honor and the badge is wrapped up in their identity.

4 Project Recommendations
Managing a data science project

Now that we've established the goals, suboptimal outcomes, and underlying causes of those outcomes, it's time to discuss how to modify our data science machine to achieve the promising results we know are possible. In this chapter, we synthesize the successful project practices from dozens of leading data science organizations spanning many sizes and industries. This chapter is deliberately more detailed and tactical than earlier sections specifically so that readers can take away actionable insights for their own organizations.

Before jumping into the details, Domino Data Lab's overall lifecycle methodology can be viewed in this **aggregate flow chart** (Appendix) that encompasses the people, process, and technology we see across leading organizations. The approach can be summarized as: Imagine your ideal process for a single data science project, then consider how to manage a portfolio of those projects, and then think about the *types of people, tools and organization structure* you need to best achieve your goals.



See Appendix for larger version

Finally, we would be remiss if we didn't mention the existence of other project frameworks for data science, notably **CRISP-DM**. What follows is inspired by CRISP-DM and other frameworks, but based more on practical realities we've seen with leading data science organizations, like Allstate, Monsanto, and Moody's. We step through the key stages that we've seen consistently emerge across many organizations' data science lifecycle: ideation, artifact selection, data acquisition and exploration, research and development, validation, delivery, and monitoring. However, the methodology and best practices here are broader than the process to manage a single project.

In **Chapter 5**, we take a more system-level view that includes the many people, processes, and tools across a full portfolio of projects.

Overall Lifecycle Principles

Before jumping into the specifics of each project stage, below are a few guiding principles.

Expect and embrace iteration. The flow of a project is highly iterative, but, by and large, nearly all projects pass through these stages at one point or another. It is normal for a project to get to validation and then need to go back to data acquisition. It is also normal for a single project to open 10 additional avenues of future exploration.

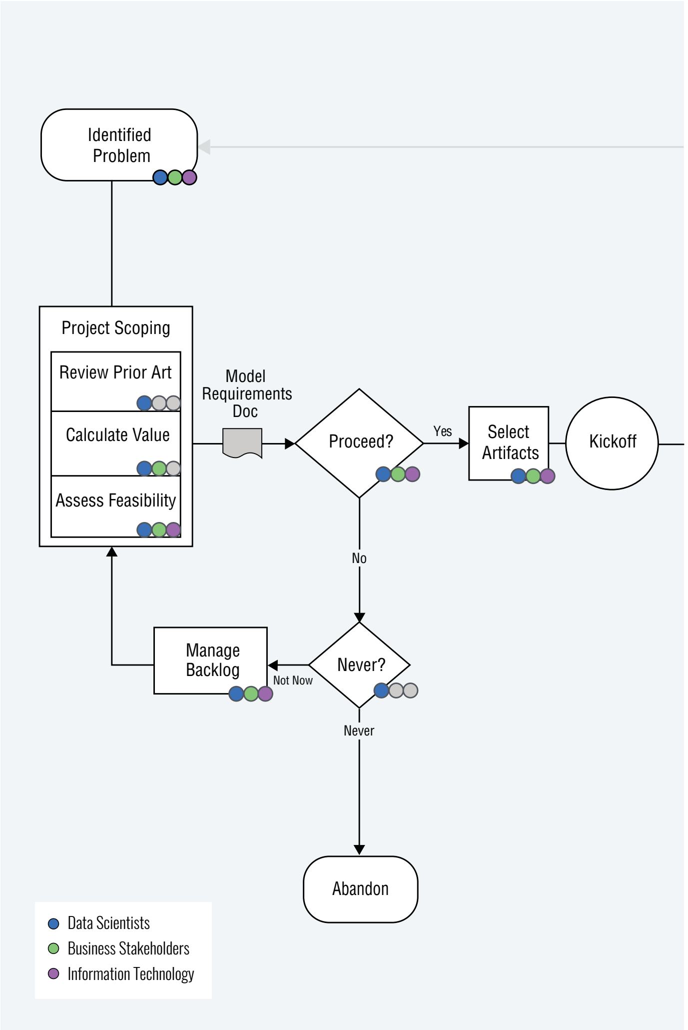
What separates leading organizations is their ability to prevent iterations from meaningfully delaying projects, or distracting them from the goal at hand. One leading organization includes an "areas for future exploration" in all project deliverables and has educated business stakeholders in "lunch-and-learns" to expect many loops through the process.

Enable compounding collaboration. High-performing data science teams realize they work best when they stand on the shoulders of those before them. One data science leader even goes so far as to track component reuse as a KPI. Data scientists who create widely used components (e.g. a great dataset diagnostic tool) are given visibility and credit for their contributions to the community's success.

Anticipate auditability needs. As more and more critical processes incorporate data science results, it is essential to be able to audit and inspect the rationale behind the model. The financial industry is formally regulated under "**model risk management**."

Yet other industries are also taking proactive steps to build model risk expertise and preserve all relevant artifacts associated with the development and deployment of a model. More recently, there is speculation that **technology firms** could follow suit to preserve model integrity.

Stage 1: **Ideation**



Some of the most important work in the overall lifecycle happens before a line of code is written. If done well, the ideation stage dramatically de-risks a project by driving alignment across stakeholders.

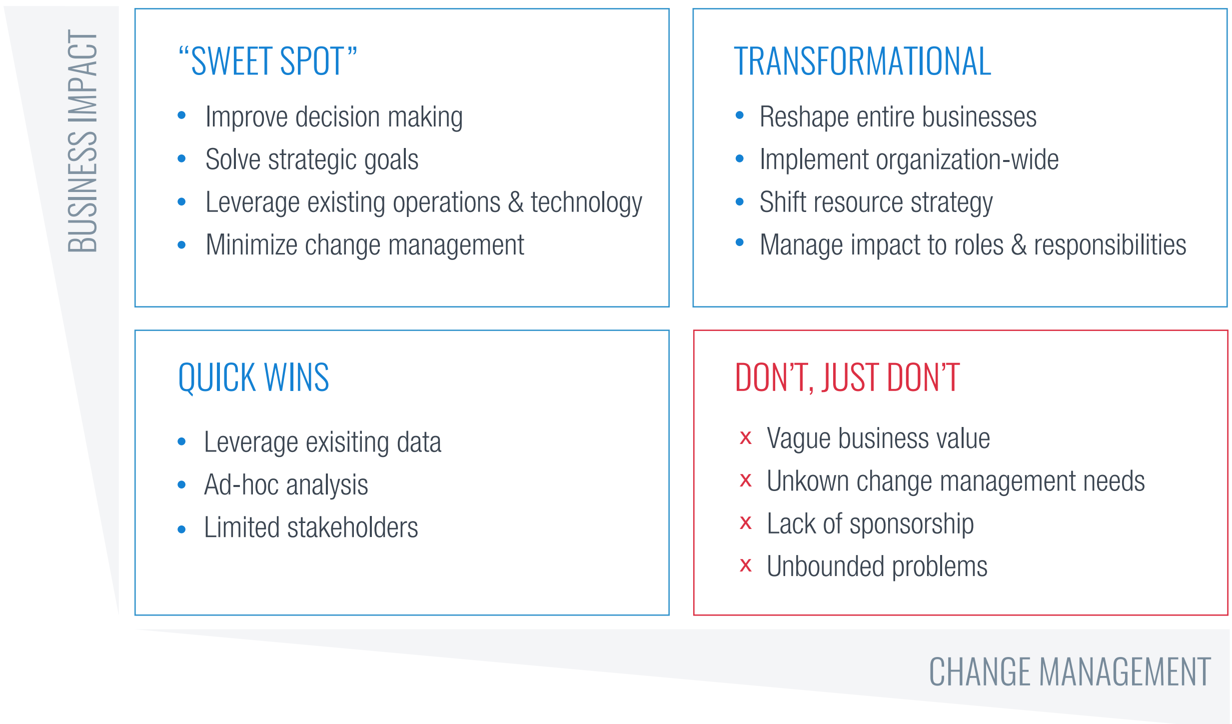
This is where the business objective is identified, success criteria laid out, prior art is reviewed, and initial ROI calculations are performed. Ideation is when feasibility is assessed, both in terms of "Does the data even exist?" and "Can we realistically change how the business process works?" It is also where prioritization happens relative to other potential projects. Below are some best practices observed that get to the root of many of the problems discussed earlier.

Problem first, not data first. Many organizations start with the data and look for something "interesting" rather than building a deep understanding of the existing business process and then pinpointing the decision point that can be augmented or automated. The genesis of a project need not only come from the business, but it should be tailored to a specific business problem.

Map existing processes. Leading organizations map existing business processes in tools like Vizio, PPT, or LucidChart and then circle on that map the exact points that data science could drive business impact. Doing so ensures they aren't missing the opportunity to target more impactful parts of the process. It also ensures they're speaking the language of their stakeholders throughout the lifecycle to minimize change management down the road.

Practice and master order of magnitude ROI math. The ability to estimate the potential business impact of a change in a statistical measure is one of the best predictors of success for a data science team. However, despite much mathematical prowess, data science teams often shy away from back-of-the-envelope calculations. The point is not precision to hit some CFO-mandated internal hurdle rate, but rather to aid in the prioritization process. For example, a large insurer asked, "If we reduce fraudulent insurance claims by 1%, how much would we save?" They then asked, "What is a conservative estimate of how much improvement we can expect by the data scientist's efforts?"

At the same time, they considered all of the project costs: time spent by the data science team, potential data acquisition costs (e.g., either from a vendor or internal instrumentation costs), computing resources, implementation time for IT, and training/adjustment time for stakeholders. Finally, they settled on a rough number based on past experiences but erring on the conservative side. The table below captures two useful dimensions for this exercise.



Maintain a hub of past work with business domain and technical experts. As teams grow, no one person can be an expert in everything. It's critical to have a way to search to see who is most familiar with the latest version of TensorFlow or who has done the most work in the marketing attribution space. Even better than search is the ability to reproduce this work (e.g., data, packages, code, results, discussions) which will give a substantial head start in subsequent steps. In one large tech organization, this hub also provides information into downstream consumption of work product to help assess quality. For example, a project that feeds an internal application with thousands of internal users is likely more trustworthy than a prototype that hasn't been used in months.

Create and enforce templates for model requirements documents. Documentation is often viewed as a chore, but high-performing organizations find that documentation upfront saves heartache down the road. Create a template for 80% of cases, knowing there will always be exceptions. Keep track of who is using the templates to see if it leads to productivity lift over time.

Ideally, bake this into your actual infrastructure rather than in disparate systems which often fall out of sync ("the curse of Sharepoint"). Key components of a good Market Requirements Document (MRD) include: problem statement and objective, target KPIs, estimated ROI, target data sources, known risks, relevant stakeholders, prior art (with links), and a stakeholder-centric timeline.

Maintain a stakeholder-driven backlog. Your stakeholders should always be able to see what's in flight and what's been put in the backlog. Like any product org, they don't necessarily get to change it. Yet, you should have recurring check-ins with stakeholders to ensure priorities or constraints haven't shifted.

Artifact Selection

This is where the shape of the final deliverable is agreed upon. It's always possible to amend the agreed upon deliverable or to have multiple, but visualizing the ultimate consumption medium and working backwards is key.

Ask yourself: "Are you building a one-off answer to support a strategic decision, a standalone light-weightapp for stakeholders to use, or a real-time data product that integrates into other systems?" The best organizations start simple and get the result into the business. Learn and measure before updating the model with a more sophisticated approach (e.g., more features, more complex algorithm, deeper integration).

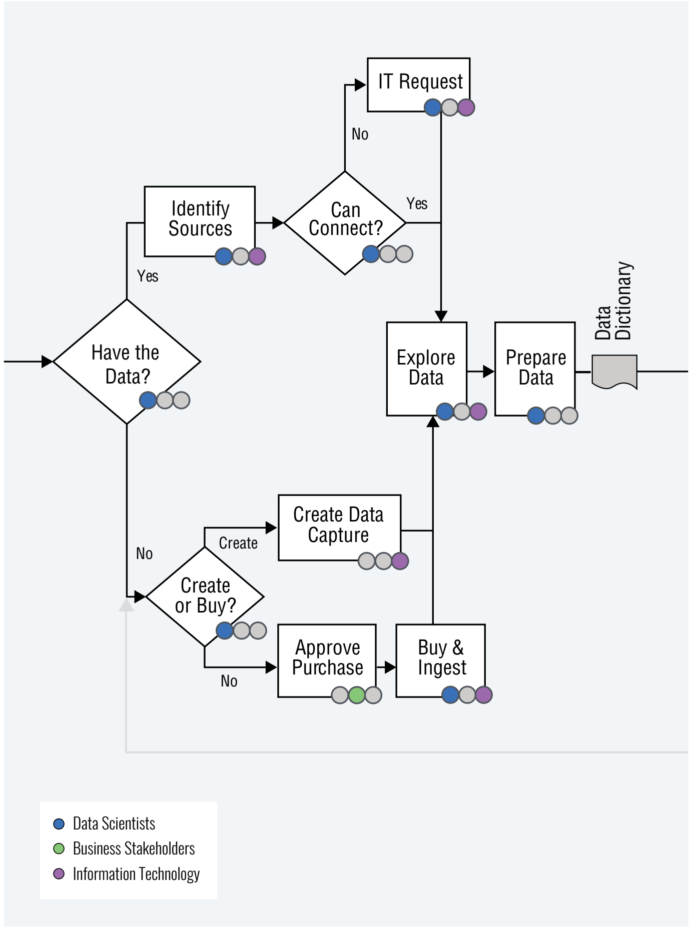
Create multiple mock-ups of different deliverable types. A leading e-commerce company creates 3-5 mocks for every data science project they take on, even bringing in a designer to make it feel real. For example, they discovered exposing their model as a HipChat bot was the most user-friendly way to leverage the model. By iterating on design possibilities before they get data, they ensure they've surfaced any previously undiscovered requirements and maximize their odds of adoption.

Bring IT and engineering stakeholders in early. A model may work spectacularly in the lab, but not have any hope of ever working in production the way envisioned by the business. IT and engineering stakeholders need a seat at the table this early in order to identify constraints like, "We only backfill that data monthly from the vendor, so we can't do a real-time scoring engine."

Consider creating synthetic data with baseline models. Some organizations even create synthetic data and naive baseline models to show how the model would impact existing business processes. A leading agriculture company devotes an entire team to creating synthetic "perfect" data (e.g., no nulls, full history, realistic distribution) to establish potential value with the business before they go contract with expensive satellite data providers to get "real" data.

> The best organizations start simple and get the result into the business. Learn and measure before updating the model with a more sophisticated approach.

Stage 2: **Data Acquisition and Prep**



Data is rarely collected with future modeling projects in mind. Understanding what data is available, where it's located, and the trade-offs between ease of availability and cost of acquisition, can impact the overall design of solutions. Teams often need to loop back to artifact selection if they discover a new wrinkle in data availability.

Extracting the most analytical value from the available data elements is an iterative process and usually proceeds in tandem with data understanding. Below are some best practices we've seen streamline an often painful process.

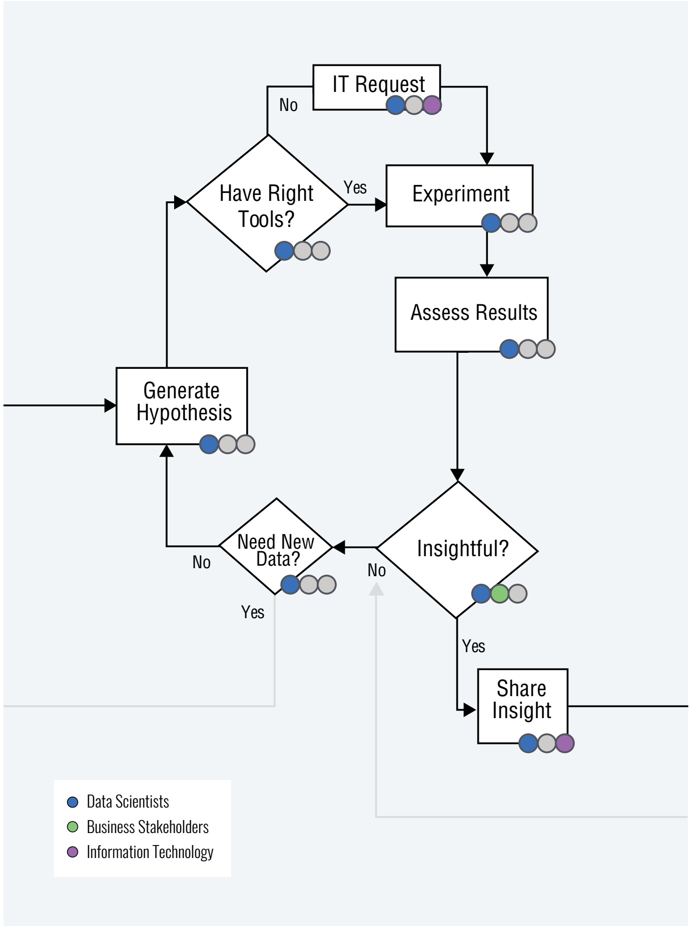
Check stakeholder intuition. Stakeholders often have solid intuition about what features matter and in what direction. Many high-performing teams extract this intuition to help point them to relevant data and jump start the feature engineering process.

Datasets as a reusable component. Given the time spent acquiring and cleaning data, it's critical that output becomes reusable by others. Many organizations create analytical or modeling datasets as key entities which are shared, meaning that they only need to interpolate nulls and exclude outliers once. Some organizations are starting to move towards "Feature Stores" to ensure people can build on past work. Regardless of the title, the work of creating these datasets should be interrogable and auditable for future research and also streamlines eventual production pipelines.

Track downstream consumption of data. Many organizations spend significant funds to acquire external data or devote internal resources to collect data without knowing if it's useful. A leading credit ratings agency tracks how many projects and business-facing apps utilize each external dataset to help guide their data investment decisions.

Develop a "play" for evaluating and incorporating external data. Increasingly teams are turning to alternative datasets to better understand their customers, whether it be social data, location data, or many other types. Organizations that have streamlined the process of vendor selection, data evaluation, procurement, and ingestion eliminate a key bottleneck. This often requires coordination with procurement, legal, IT, and the business to agree on a process. One hedge fund has reduced its evaluation/ingestion time from months to weeks, helping maintain its edge in a highly competitive space.

Stage 3: **Research and Development**



This is the perceived heart of the data science process and there are numerous guides on technical best practices. Below are a number of best practices that address many of the key reasons data science organizations struggle.

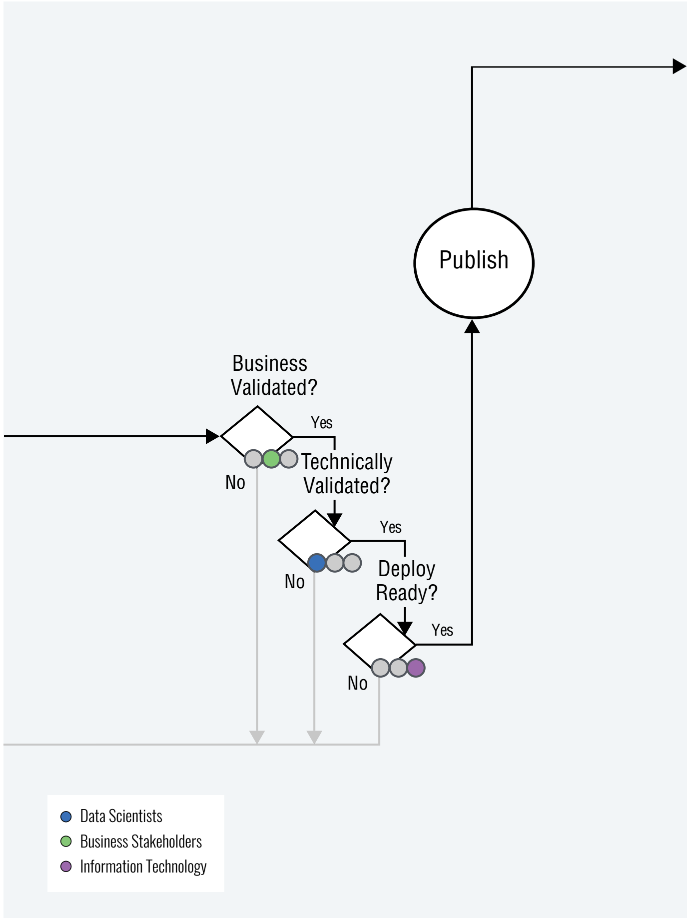
Build simple models first. Resist the temptation to use 500 features. One company spent weeks engineering the features and tuning the hyperparameters. Then they learned that many of them were either a) not collected in real time so couldn't be used in the target use case or b) not allowed for compliance reasons. They ended up using a simple five features model and then worked with their IT team to capture other data in real time for the next iteration.

Set a cadence for delivering insights. As discussed earlier, one of the most common failure modes is when data science teams deliver results that are either too late or don't fit into how the business works today. Share insights early and often. For example, one leading tech company has their data scientists share an insight every 3-4 days. If they can't publish a short post on incremental findings in business-friendly language, then chances are they are down a rabbit hole. The insight can be as simple as "this experimental path didn't pan out." This lets the manager coach more junior team members, plus gives an easily consumable medium for business stakeholders to offer feedback, spark new ideas, and gauge progress.

Ensure business KPIs are tracked consistently over time. Too often, data science teams lose sight of the business KPI they are trying to affect and instead focus on a narrow statistical measure. Leading teams ensure that the relevant KPI is never far from their experimental flows. One hedge fund tracks the overall performance of its simulated investment portfolio across hundreds of experiments and then shows this to its Investment Committee as proof of data science progress.

Establish standard hardware & software configurations, but balance the flexibility to experiment. Data scientists can often spend the first eight weeks on the job configuring their workstations, rather than exploring existing work and understanding their stakeholder's priorities. Having a few standard environments gets people onboarded faster. Yet, it's important they retain flexibility to try new tools and techniques given the breakneck pace of innovation. Cloud computing resources and container technology are well-suited to address these demands without compromising on governance.

Stage 4: **Validation**



WHAT INFLUENCES A RESULT?

Results
↓ *depend on*
The statistical analyses selected
↓ *depend on*
The R scripts that implemented the analyses
↓ *depend on*
The R libraries that implement the statistical functions
↓ *depend on*
The C libraries that perform the mathematical computations
↓ *depend on*
The operating system running the computational framework
↓ *depend on*
Reduced data
↓ *depend on*
Scripts that reduce the data
↓ *depend on*
Raw data

Validation is more than just code review. Rigorous evaluation of the data assumptions, code base, model performance, and prediction results provide confidence that we can reliably improve business performance through data science. Validating results and engaging with stakeholders are equally important in this phase. Ultimately receiving sign-off from stakeholders is the goal: the business, any separate model validation team, IT, and, increasingly, legal or compliance.

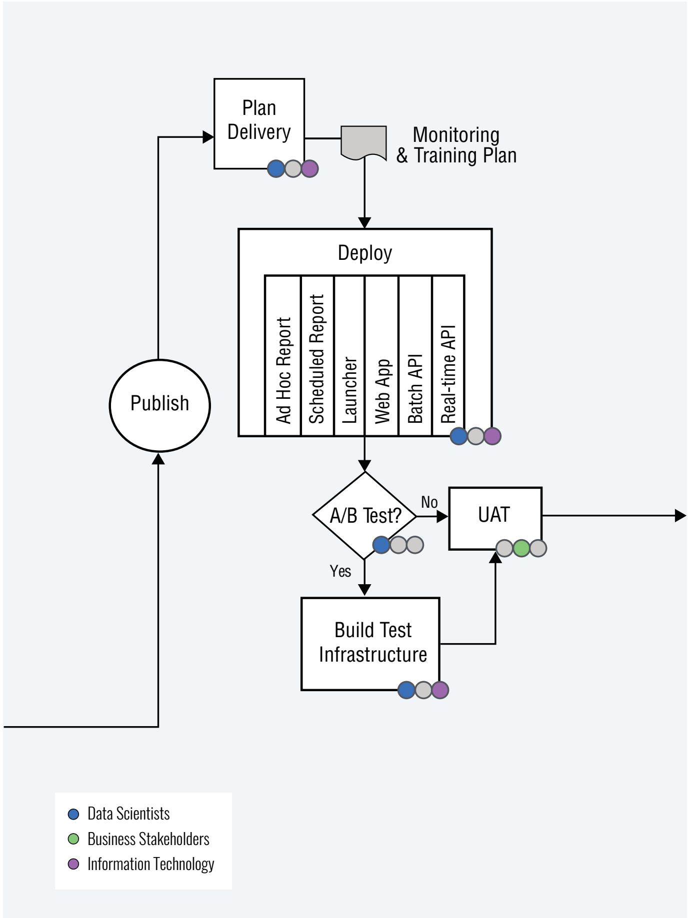
Ensure reproducibility and clear lineage of project. Quality validation entails dissecting a model and checking assumptions and sensitivities from the initial sampling all the way to the hyper-parameters and front-end implementation. This is nearly impossible if a validator spends 90% of their time just gathering documentation and trying to recreate environments. Leading organizations capture the full experimental record, not just the code. One large enterprise customer captured this well in the following diagram.

Use automated validation checks to support human inspection. While data science's non-deterministic nature means that unit testing does not directly apply, there are often repeated steps in a validation process that can be automated. That may be a set of summary statistics and charts, a portfolio backtest, or any other step that could turned into an automated diagnostic. This lets human validators focus on the critical gray areas.

Maintain record of discussion in context. The model development process often requires subjective choices about data cleansing, feature generation, and many other steps. For example, when building a home price forecasting model, the feature "proximity to a liquor store" could increase predictive power. However, it may necessitate significant debate amongst multiple stakeholders about how to calculate it and if it was permitted from a compliance perspective. Leading organizations have set up their infrastructure and process to capture these comments and discussions and preserve them in context rather than scattered across countless email chains.

Preserve null results. Even if a project yields no material uplift and doesn't get deployed into production, it's critical to document it and preserve it in the central knowledge library. Too often, we hear that data scientists are re-doing work someone explored without knowledge of previous inquiries.

Stage 5: **Delivery**



The delivery path taken depends on the initial artifact type determined. This is when a mathematical result becomes a "product." Deploying into production can be as simple as publishing the results as reports or dashboards, incorporating the modeling pipeline into batch processes (e.g., new direct marketing campaigns), or deploying models into production systems for automated decision making (e.g., online credit decisions or product recommendations).

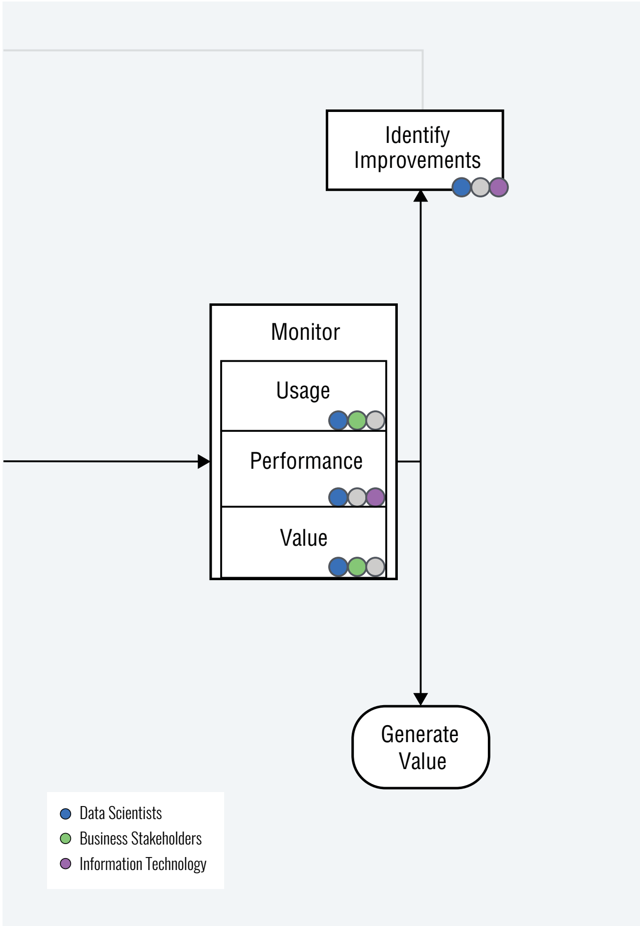
Preserve links between deliverable artifacts. While real-time scoring gets all the glory, the vast majority of models will at one time or another be reports, prototype apps, dashboards, or batch scoring engines. It's important to keep a link between all those deliverables because it saves time and avoids risk that key feedback is lost if something goes awry.

Enforce a promote-to-production workflow. As a result of incentive structure and responsibility alignment, data science teams often stumble in the last mile. If you establish the workflow ahead of time, you reduce the bottlenecks for delivery without adding operational risk. For example, pre-determine which packages and tools are permitted in critical path production workflows and ensure consistency of results relative to a dev environment result. Determine if and how much human inspection of early results is necessary in staging as automated testing may not be sufficient.

Flag upstream and downstream dependencies. A model is at its most risky when it finally makes it to production. Ensure that you know the upstream dependencies: what training data was used, what transformations were done with what tools, what modeling packages were used, etc. Also make sure you know the downstream dependencies (e.g., this nightly batch model is stacked on another model).

Anticipate risk and change management burdens. High-performing teams anticipate the human component and proactively address it with training. One leading insurer has a dedicated role that helps train business units on how a model works, why they did it, and acts as a single point of feedback for future iterations on the model. This has dramatically increased adoption across non-technical stakeholders like claims adjusters.

Stage 6: **Monitoring**



Models are at their most impactful when they are actually "live" and affecting people's behavior, whether internal workflows or customer engagement. Given the non-deterministic nature of data science "success", it's critical to have a rich monitoring suite that takes into account the semantic and statistical measures in addition to traditional system measures covered by application performance management (APM) tools like New Relic.

Consider control groups in production. While it is hard to convince business stakeholders that the fantastic model you've just completed shouldn't be applied universally, it's often critical for long-term success. One leading organization established a global holdout group from all of their customer segmentation and price elasticity models. After a year, they compared the average revenue from the holdout group to the customers whose experiences were guided by the predictive models. The overall lift was more than $1 billion, which gave them the credibility to dramatically expand the team and push models into more steps of the customer journey.

Require monitoring plans for proactive alerting, acceptable uses, and notification thresholds. The data scientist who created the model is the person best positioned to know what risks are inherent from their approach. Rather than wait for the business to notice something is wrong or a metric to drift, codify that knowledge into your monitoring system. Do you expect certain input types and ranges? If it's outside of those, what should you do? Rollback? Stop serving predictions? What if someone in a totally different department starts consuming the model in a way that may be risky or outright wrong? Working collaboratively with IT or engineering, data scientists can put the appropriate guardrails on their creations.

Integrate monitoring with tools where people spend most of their time (e.g., email, Slack). High performing teams realize that monitoring is only good if someone acknowledges, inspects, and changes behavior if necessary. We've seen organizations build alerts into chatbots or email systems to ensure they can keep up with the alerts as their number of production models scales.

5

System Recommendations
Scaling a good data science project to a business discipline

Early data science wins drive more project requests and more hiring, which generates a whole new class of problems for managing the growing portfolio of data science work. Adopting and adhering to a single project framework can help address many of the underlying reasons for failure, but data science managers must consider how they scale if they are to cement their place as a core capability. Below are a few best practices used to manage teams of 100+ data scientists and 300+ simultaneous projects.

Measure everything, including yourself. Ironically, data scientists live in the world of measurement yet rarely turn that lens on themselves. Tracking patterns in aggregate workflows helps create modular templates, disseminate best practices from high-performing teams, and guide investment to internal tooling and people to alleviate bottlenecks. For example, by examining their complete body of work over multiple years, one large tech company realized they only have 15-20 canonical problems to solve and then planned to just apply templates where appropriate. Another organization does quarterly reviews of the aggregate state of hundreds of inflight projects and realized they were consistently blocked in ETL, so re-allocated budget to increase data engineering hires.

Focus on reducing time to iterate. Many organizations consider model deployment to be a moonshot, when it really should be laps around a racetrack. Minimal obstacles (without sacrificing rigorous validation) to test real results is another great predictor of data science success. Leading tech companies deploy new models in minutes, whereas large financial services companies can take 18 months.

Socialize aggregate portfolio metrics. Even if it's not precise, it's critical to socialize the impact of the whole portfolio of data science projects. Doing so addresses data scientists' concerns about impact and helps address executive level concerns about investing further in data science. Let stakeholders know how many projects are in the backlog and aggregate pipeline. Many are shocked to realize that dozens of projects are in flight as they typically had line of sight into only a few. Importantly, many successful data science managers don't claim the credit for themselves, but as a collective achievement of all the stakeholders.

People

A successful data science machine comes from the coordination of people, process, and technology. We've gone in depth through the processes we've seen drive success, but in this section we examine the people. As mentioned above, the full stack data scientist no longer exists and the data science roles are increasingly specialized. This is a natural evolution that we expect will continue as data science becomes ingrained into the fabric of how organizations function.

The most consistent feedback we've heard is the increasing demand for a "product manager" type role as most organizations move from delivering mathematical results to stakeholder-facing apps. In large tech organizations, data science sits peer with product management to drive strategic priorities and ongoing optimization of engagement and impact. Below is a list of the types of roles across successful organizations. This is by no means definitive and actual titles can vary but this represents the broad shape of responsibilities we've heard.

Role	Responsibility	Pitfalls without them?
Data Scientist	Generating and communicating insights, understanding the strengths and weaknesses of algorithms and features.	Naive, or low power insights.
Data Infrastructure Engineer	Building scalable pipelines and infrastructure that make it possible to develop and deploy models.	Insight generation is slow because data scientists are spending their time doing infrastructure work.
Data Product Manager	Responsible for clearly articulating the business problem, at hand, connecting through domain knowledge about the business problem and translating that into the day to day work. Also, ensuring training and ongoing engagement with deployed models.	Projects miss the mark. The data scientists spend their time playing with math, model/features selections are mathematically valid but ultimately domain-divorced.
Business Stakeholder	Responsible for vetting the prioritization, ROI, and providing subject matter expertise throughout.	ROI decisions aren't made sensibly, not knowing when to pull the plug, non-actionable results.
Data Storyteller*	Creating engaging visual and narrative journeys for analytical solutions. Somewhat analogous to a designer.	Low engagement and adoption from end users.

*Note, the data "storyteller" is a role we're just starting to hear more about, though remains rare in the market.

Organizational Design

The final question often heard from managers is how to organizationally configure their teams once they've established the people to hire and process they want to follow. Typically, there is an almost religious debate between centralization and decentralization. We believe this is a false dichotomy and instead offer two best practices we've seen in the field. But first, the table below summarizes the arguments on each side.

	Centralization	Decentralization
Pros	Community and mentorship, easier transparency for managers and IT, more passive technical knowledge sharing, more easily attained governance and regulatory compliance.	Deeper understanding of business processes and priorities, easier change management, fewer bottlenecks.
Cons	Greater risk of low adoption of results, fewer proactive project ideas, loss of credibility with business.	Less technical knowledge compounding, harder to codify best practices, risk of shadow IT.

Prepare to evolve as the team scales and as business demands shift. Many organizations start with a centralized "Center of Excellence" for data science to build their core technical infrastructure before evolving to a hybrid structure, which we call the "hub and spoke" model. In this structure, the centralized team are the keepers of best practices and focus on building templates (documentation, software environments, project stage flows). Meanwhile, embedded groups sit next to each major business line to address the "bookend" problems of identifying the right data science problem and maximizing adoption of solutions. Sometimes a full data science guild (to borrow Spotify's term) exists and regularly meets regardless of their day-to-day functional department to foster community cohesion.

Deploy tech to address the pains of decentralization. A data science platform can facilitate technical knowledge sharing, encourage or enforce best practices, and provide transparency while still allowing data scientists to be closer to the business. It is much harder for current technology to overcome the challenges of centralization, like isolation from the business. In particular, a "hub and spoke" organizational structure supported by a data science platform can benefit from the proximity to the business in ideation and delivery/monitoring phases (the particularly challenging "bookends" of the lifecycle). At the same time, consistent infrastructure can encourage reuse and compounding to minimize time to delivery and establish a flywheel effect across the entire portfolio of work.

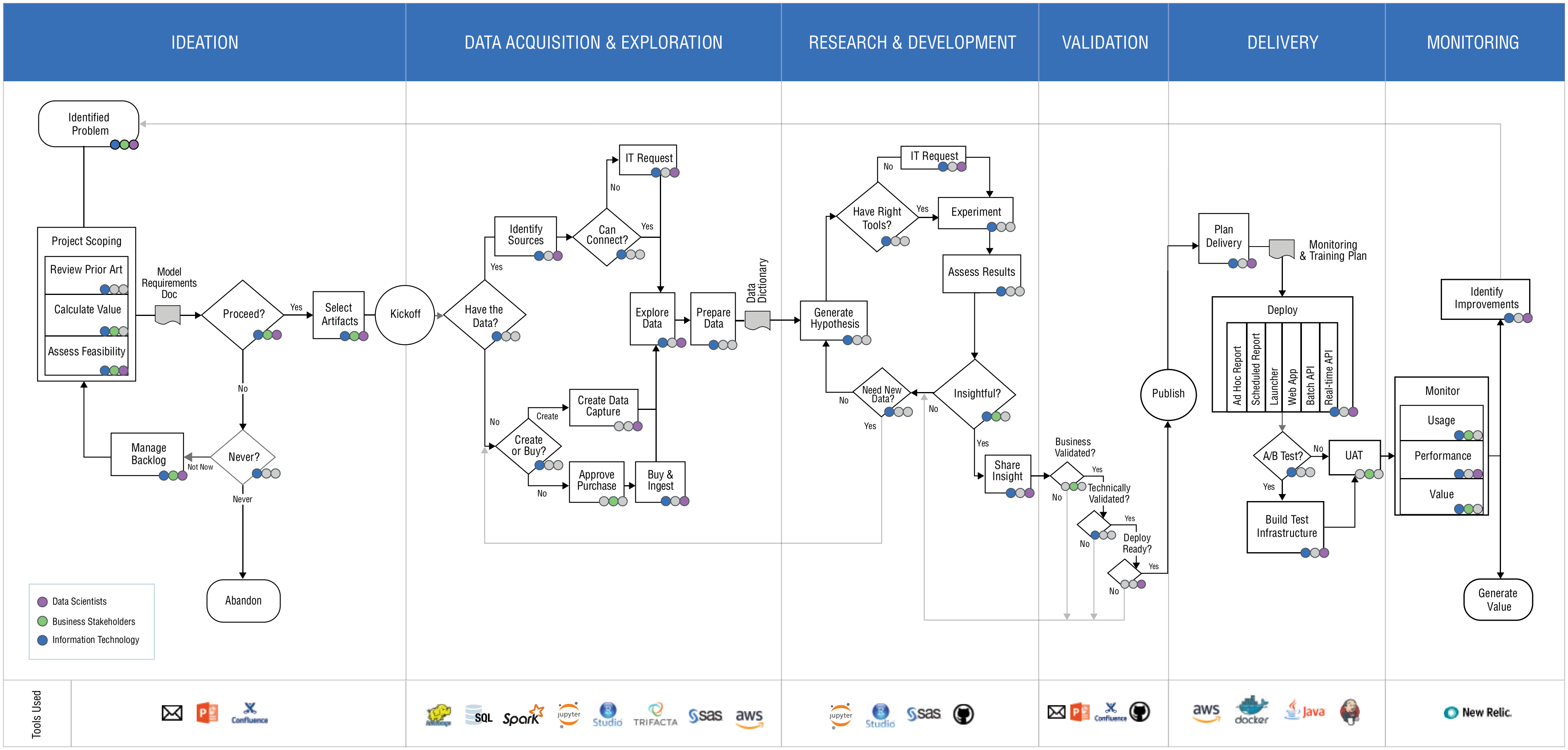
Conclusion

Data science success at scale is not as easy as bringing in a single "silver bullet" technology. It requires maturity and vision across many dimensions: hiring people, implementing processes, and acquiring technology to support those people and processes. While this can seem overwhelming, we've seen that organizations do best with a "crawl, walk, run" approach. For many, the most important next steps are:

> **Conduct an assessment** of your current project lifecycle
>
> **Gather data scientists and stakeholders** from different groups to share their pain and wins
>
> **Inventory** your current models

Over time, work with the different groups of stakeholders to establish the goal, determine where you are missing that goal, honestly assess the reasons for those misses, and then take tangible steps to improve the system. Unless we're able to examine the messy guts of today's process, we won't be able to achieve long-term glory. The data science leaders that navigate this journey successfully will not only set the standards for everyone else, they will reap the benefits of imbuing their entire organization with a powerful head start in the next wave of computing.

DATA SCIENCE LIFECYCLE



DOMINO