Data Products

Playbook

equalexperts.com/playbooks/data-products

EQUAL EXPERTS

Contents

INTRODUCTION

4 What is a data product?

7 Who are the users of a data product?

ADVICE

9 Advice to business analysts

11 Advice to product owners

PRINCIPLES

14 A data product should be user-centric and meet at least one clear business problem

16 Data that is not used to create insight is simply a cost

16 A data product must be trusted

18 Data products should be built to be reusable

20 Data products can be multi-modal

20 A data product should meet data governance needs

21 Strive for standardisation across your data products

22 Making and maintaining a data product is a team sport

23 Insights only exist in production data

PRACTICES

24 Data product definitions should have these properties

25 Make your data products easy to find - make them discoverable

27 Deliver your data products incrementally

27 Apply Continuous Delivery and Testing where you can

28 Data Quality is an iterative process

29 What lower environments are for/ can test

30 Data personas can be helpful for improving understanding of users as well as product focus and prioritisation

32 Product visions are a helpful tool to align teams and communicate with stakeholders

32 Entities that cross domains need naming standards first

33 Make metadata manageable

33 Understanding and defining user access roles are key activities of the BA and product owner

34 Build computational guard-rails to enable continuous governance

36 Use tried and tested methods for how you deal with data product changes

37 Start your product with a single user in mind, but build for reusability

37 Provide a channel for users to find out about the product and talk to the data product team

39 When to build a data product platform

EXPLORE

40 Should you provide self-service data access?

41 Letting other teams work on the data product

PITFALLS

44 Building products for engineers rather than for users

44 Treating a technology as a data product

45 Believing data discoverability can be achieved out of the box

46 Insufficient focus on data quality

46 Data products are owned by a centralised enterprise function

47 Don't process and expose data just because you have it

Introduction

Data Products playbook

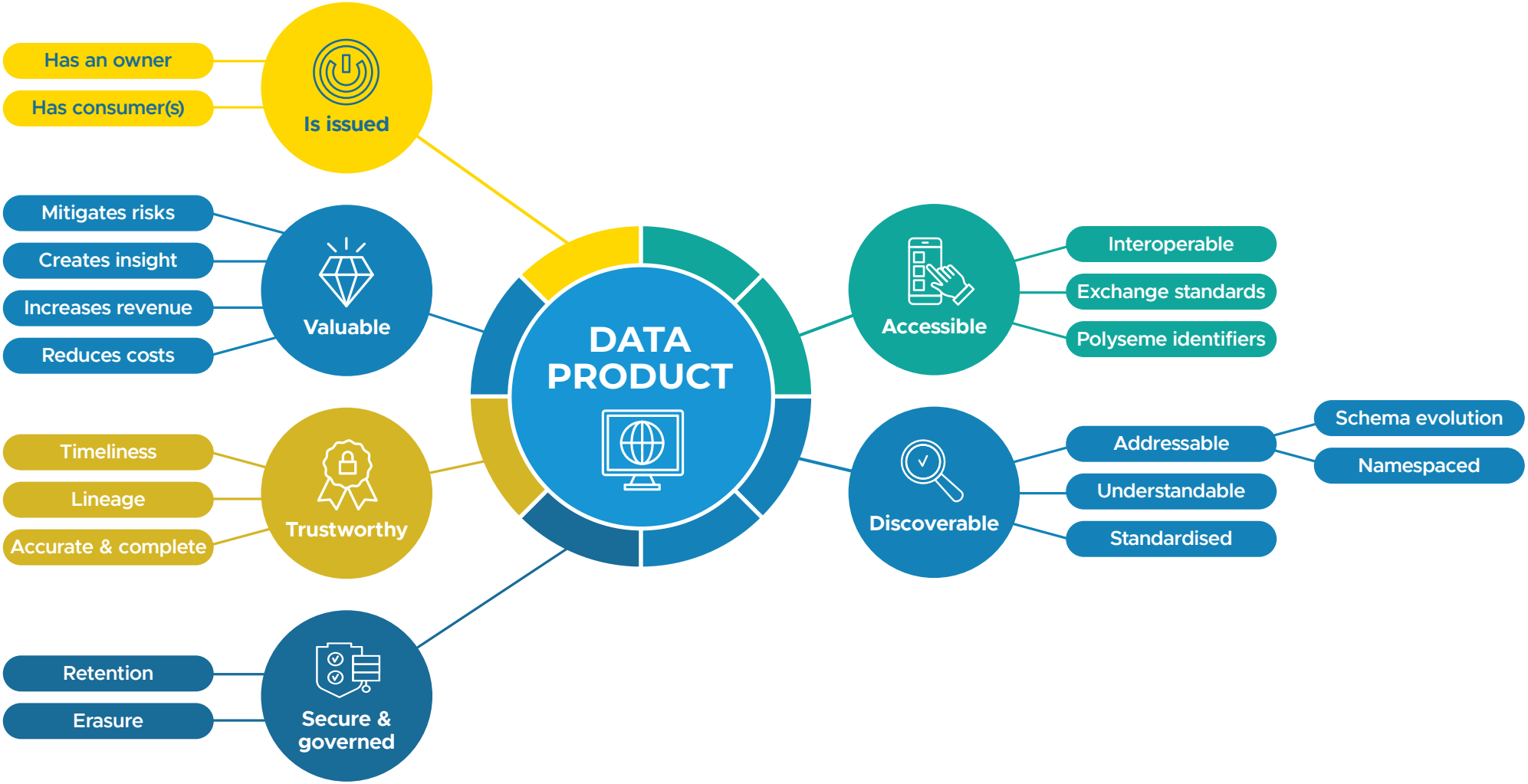
Product thinking has made a huge difference to the development and operation of effective digital products and services. We believe the same approach applies to products that provide data or depend on data. Thinking about data as a set of products with their own domains rather than as an amorphous mass across a business brings many benefits. It explicitly and clearly makes the link between data and its users - reducing the time to insight. It also helps create a manageable scope for development teams, and enables scaling.

Many aspects of product management and thinking remain the same for data products, but there are some important differences. We have written this playbook to bring out some of these nuances in the hope that it will be useful for data product owners and managers, business analysts and anyone involved in the creation and operation of data products. It is a sister playbook to our data pipeline playbook https://playbooks.equalexperts.com/data-pipeline , which focuses on how to implement and engineer the data pipelines which are the critical foundations for any data product, and our ml ops playbook https://playbooks.equalexperts.com/mlops-playbook , which focuses on creating and operating data driven services.

What is a data product?

A data product is a piece of software that provides, consumes or interprets data, has users and provides value to business operations. It provides value by helping the business understand its operations or customers better. It requires support and maintenance and should have a product owner whose role is to make sure the data product meets the needs and quality requirements of its users and of the business. Products have a lifecycle - they need to be iterated, maintained and eventually sunsetted, retired or replaced.

Data products are valuable, trustworthy and well governed. They are accessible and discoverable.

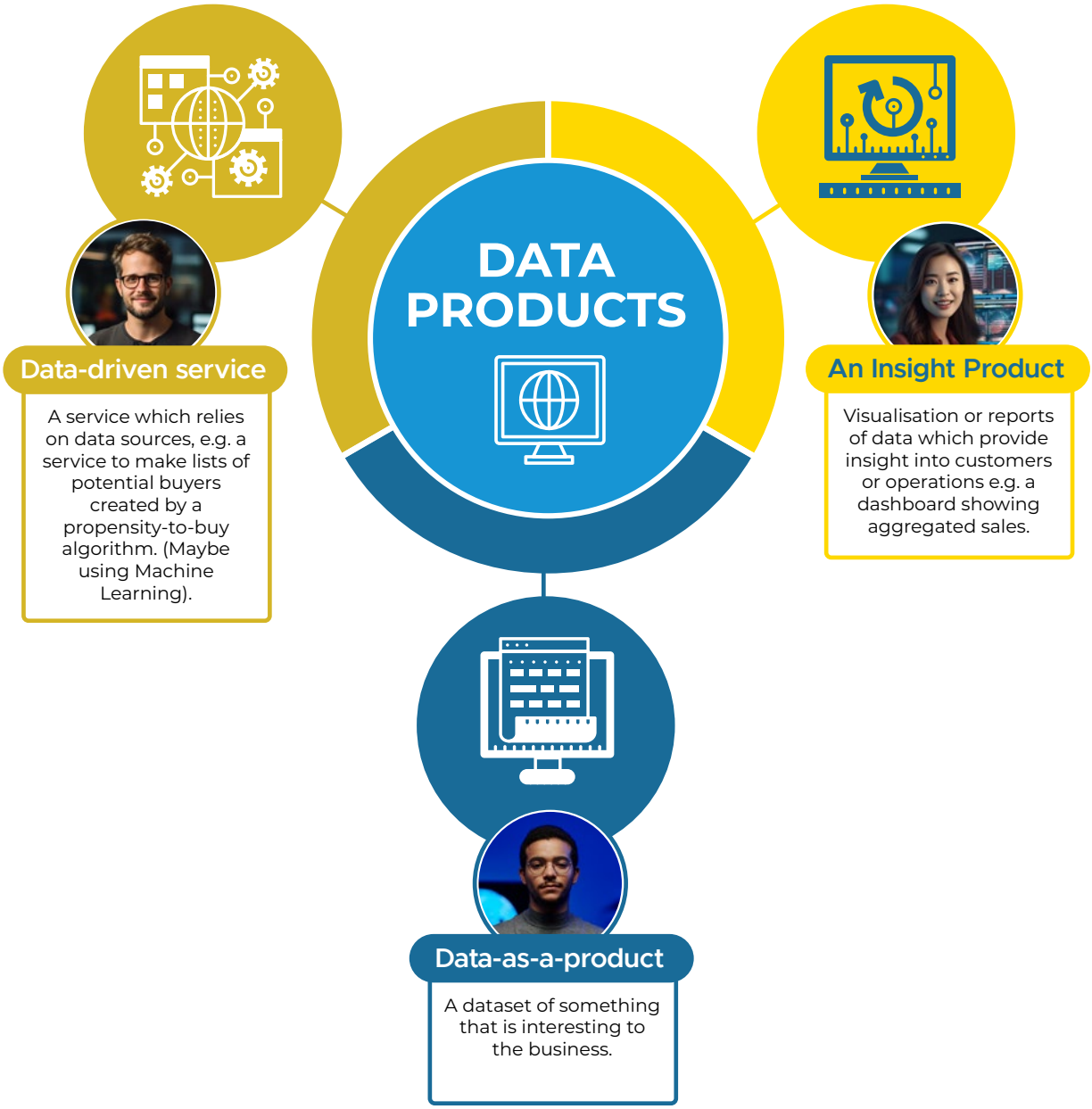

DATA PRODUCT
Is issued
Has an owner
Has consumer(s)
Valuable
Mitigates risks
Creates insight
Increases revenue
Reduces costs
Trustworthy
Timeliness
Lineage
Accurate & complete
Secure & governed
Retention
Erasure
Accessible
Interoperable
Exchange standards
Polyseme identifiers
Discoverable
Addressable
Schema evolution
Namespaced
Understandable
Standardised

We typically see three distinct categories of data products:

- **Data-as-a-product:**
  A dataset which is interesting to the business, e.g. a table in a database containing sales information.

- **Insight products:**
  Visualisation or reports of data which provide insight into customers or operations, e.g. a dashboard showing aggregated sales.

- **Data-driven service:**
  A service that relies on data-sources, e.g. a service that identifies likely buyers of a product based on a propensity to buy algorithm, which was in turn developed using historical sales data.



Who are the users of a data product?

There can be many different types of people who use a data product.
We often see:

- **Data analysts** - use the data to provide reports on an aspect of the organisation or as a means of identifying or diagnosing issues in the organisation.
- **Data scientists** - use the data product to create models (machine learning, etc.) which become incorporated into business operations.
- **Business users** - use the data product for detailed understanding of their organisation operations or customers. This can be a diverse group of people who want to monitor specific business processes. They can be passive consumers of the data or power users who are keen to access and manipulate data for their own purposes.
- **Decision makers** - require aggregated data to understand how their area of the organisation is performing or to meet regulatory requirements.
- **Other engineers** - use combinations of data products to build new data products that provide value to the organisation.

A given data product may be used by more than one of these users, or aimed at a single category. Different users have different data skills. Some (like data analysts and scientists) are skilled at manipulating and understanding data and may appreciate data products that have a wide set of uses. Others may not have the skills or time and will prefer a more targeted product that addresses a specific need - and will probably value visualisation very highly.

I want to forecast the profitability of the new children's range for Summer 2024

Data scientist
Discovery team

Uses data daily and is comfortable working with raw data. They need access to fresh data and to be able to transform it and use it for daily campaigns and reporting.

What % of customers with a credit card account spend over £5000 per year?

Product manager
Financial Services team

Uses data to inform decision making for product strategy and to monitor performance of the digital ecommerce experience. Would be using dashboards and analytics tools to consume and interrogate data.

How many customers have bought 1 or more items from our mid-century range in the last 30 days?

Head buyer
Retail & Trading team

Uses data to map buying trends and understand purchasing behaviour in customers. Uses market data and needs quantitative data to inform decision making as a buyer.

Shared core needs

Easy to use	Access	Reliability	Timely	Clear governance	Secure
I need to be able to present data to others in a way that is easy to consume	I need to make sense of the data without the help of a data analyst every time	I need to analyse data at the appropiate level of fidelity and detail	I need data that is up to date to be able to blend or aggregate data to illustrate a point of view	I need to expose data provenance for reference and credibility	I need to customise the view to show me the detail I need

Advice

Advice to business analysts and product owners

We hope this playbook will be useful for many different people, although we have written it to help people used to developing more standard digital products - in particular, business analysts and product owners who need to create the product vision, define the product and understand and prioritise user and other stakeholder needs.

Advice to business analysts

In some ways, data products are similar to other software products. In both cases, you are usually trying to understand the user need. A lot of detail might be required to understand that need, and it may be tough to find a stakeholder who is happy to commit to making assumptions in the name of progress. Often, a lot of face time is needed to capture and validate all the necessary information. Like any other product, a data product does not need to be feature-complete to be valuable, and defining how good is 'good enough' is an important part of the process.

In other ways, data products are different from other software products, from the point of view of a business analyst. In software development, we are often trying to specify a feature or system, and it's more important for the BA to understand the concept first, then work with technical and business experts to develop relevant stories. In data products, the details are incredibly intricate and specific, so it's important to conduct a thorough cross-examination of business users and stakeholders. Compared to other digital products, understanding the data journey is at least as important as understanding the user journey. With data products, it's essential to understand how data flows.

For example, let's consider how organisations define a client. A retailer might consider anyone who has made a purchase to be a client, before breaking down this group by guest shoppers versus logged-in clients, or other characteristics. A service-based company that offers a combination of paid-for and free services might define a client as anyone who has consumed free advice, regardless of whether that person has made a purchase. A business might then classify active customers according to a wide set of parameters such as:

- anybody who has browsed a certain section of the website in the last three months.
- someone who has made a purchase in the last six months.
- someone who has purchased something in the distant past, but still engages with marketing content.

This type of data insight demands extensive technical analysis, so data products often require BAs to undertake technical data discovery and even archaeology. This will likely include investigating and querying existing databases, and requires a good understanding of common data terminology (such as joins) to help identify which features can be used to combine datasets for a given data product.

Some other tips our business analysts have found useful are:

- Because of the extensive analysis requirement, it can be a good idea to tee up a few requirements before the development effort begins to allow the BA to keep on top of adding to the development pipeline.
- Data scientists and analysts usually know a lot about the data; sometimes they are happy to act as the BA for technical aspects. In any case they are usually very useful sources for understanding data.
- Data users are often internal users and may be small groups or even one individual. You probably need to worry less about addressing the needs of very varying user types.
- Dashboards are usually implemented in tools such as PowerBI or Tableau, which have limitations in what they can and cannot do. Many, but not all UX ideas are achievable.
- Lots of the work is understanding where to get and how to access source data - is it an API or SFTP? Does it arrive as a CSV or parquet file?
- Understanding the taxonomy of data elements is important. There may be more than one way to classify and arrange data depending on user needs.

Advice to product owners

Data products are similar but different to normal products. Data product owners still have the key roles of developing and defining what the data product needs to be, then communicating that to the various stakeholders and prioritising work on the product. As a product owner, you still need to translate users' desires into product features. You still need to remove dependencies, and strive for independent modules.

Like any other product, a data product has roadmaps. You should think about data like a microservice - it will need to evolve. Like a microservice, a data product may touch multiple teams, and you will need to find out who the users are, and agree on how to change the product. Also like microservices, many of the building blocks of data products can be built in parallel, so dependencies are broken down.

However, there are some differences compared to standard digital products. The main challenge for many data products is connecting the producer and consumer of data, rather than providing a user journey. This means that the UX focus is about understandability and discovery. Information architecture and user experience design are more important than service design.

It can be hard to give shape to a data product because it may seem more abstract than an application. Defining the domain of your data products is challenging - which parts of the data should be part of the product? Every data product should be accessible to users, but being driven by other needs can lead to users being confused by changes not driven by their needs. Some products provide a lot of data, some are very specific. One challenge is to know when you need a new product, or need to split a new product out.

The relationship with stakeholders and consumers is different and, unlike standard products, data products do not necessarily have a simple market and fit. When working with consumers, you segment primary and secondary users, and orient your development to their needs. Stakeholders are interested in how well the product meets these needs or delights the users. When it comes to data products, you might have a small number of internal users who have similar needs. Your challenge is to understand and meet those needs while not being so closely aligned that other data users won't benefit from the product.

With data products, you won't always see the end-to-end experience, or how users actually engage with a product (although you should definitely try). This is especially true for data-as-a-product, where you may be providing a product that is a small part of a larger pipeline.

Data products often depend on other products. For example, insight products like a dashboard with an important UI aspect, will depend on a dataset (data-as-a-product). Or placing an API in front of a dataset can surface that data-as-a-product in a new way.

Experience Report

Nathan Roche | Data Product Manager

Being a data product owner is very rewarding for people who gravitate towards structure. Although it can sometimes seem quite nebulous or daunting, you get to define rich rule sets that make it easier for other people to understand a part of the world. It can be hugely satisfying to take something vast and complex, and present it in a way that is simple and self-evident. A good data product should almost describe itself; it should be obvious how it can be used without requiring the user to know how it was shaped.

One important thing I have noticed about data products is that it's important that everyone signs up to a definition of a data product, distinct from the tooling that serves it. What is the thing of value (the product) and separately what are the tools that help us deliver that product. Otherwise everyone will be trying to navigate a soup. For some reason this is more difficult in the data space.

Principles

A data product should be user-centric and address at least one clear business problem

In our data pipeline playbook https://playbooks.equalexperts.com/data-pipeline we were keen to point out that data has users. Since then many data leaders have told us that this was helpful to them when planning and implementing data pipelines and platforms. So we'll say it again - like any other software product a data product must have a user who needs the data to help solve an organisational challenge. Designing and implementing with a user in mind who is involved in the process means that the data will be useful, you will be able to consider genuine needs for things like information security and discoverability, and you are much more likely to provide something of business value than simply making data available in a void.
You will also deliver value much earlier.

Experience Report

Munish Malik | Lead Product Consultant and Coach

When I was the product coach for a data lake initiative, my product point of view made it clear to me that the right way to go was to 'start at the end.' I started by understanding the needs of the consumers of the data and then working backwards. I picked one vertical - the financial risk team - and asked, 'What analytics do you need? What should a dashboard look like?' The end users were very comfortable talking in this way and talking about what they needed visually. They could easily describe what data they needed to make a decision and what format it should be in. They were clear on the kinds of graphs that they wanted. What are the decisions that you want to make? What kind of data do you need for it? This is a very natural conversation for people to have.

Then we could talk about what we needed to enable those analytics. What validation did the data need? When would you think something was not right in the data? These naturally led to data quality specifications. These discussions allowed us to naturally build out epics and requirements.

It takes lots of teams to create a dashboard. I found that involving the product owners of the other teams early on really helps, especially with understanding data governance and data quality.

Data without a purpose is simply a cost

In itself, data has no value. It only becomes valuable if it does something meaningful for people or processes in your organisation, typically improving understanding and insight into your operations or customers. This could be raising awareness, providing metrics to measure progress, supporting decision making or helping resolve issues.

A data product must be trusted

Belief that data is accurate and reliable is critical to a data product. If data becomes untrusted, then it will not be used for decision making and has no business value. Or worse, competing data products that show different results lead to lots of effort working out which is the true answer - and create negative business value. Many organisations have found to their cost that once trust in a data product is lost it is hard to regain.

Data trustworthiness is positively correlated to data quality. Data quality includes the following kind of characteristics:

- Accuracy and completeness - the records should accurately report the real world, and we should know how accurate any dataset is. The more complete the data about a domain, the better the insights will be, so we should know how much of the data we have collected.
- Timeliness - data freshness: time from business/customer event occurring to the data being available for consumption.
- Provenance - the lineage of the data from source datasets, source events, including enrichment and synthesis with other datasets/product should be clear.

Reliability is an important foundation of trust. Users need to know that data will be available in the way they expect and is up to date. Users can perceive data as unreliable for many reasons. For example, if someone renames the column in a data product that I query, then all the analyses, insights, models etc, that I and other people depend on will fail. It doesn't matter how accurate it is, you just broke the downstream analyses for me and every other consumer who touches that column. Likewise, if the results for the same query suddenly change, then people will lose trust in the data. (So if you find that new data or a change in the calculation of data is going to create a big change, let your users know beforehand and explain why the data has changed).

You need this kind of trust from day one, and we believe it is best addressed by adopting good engineering principles (see the data pipeline playbook https://playbooks.equalexperts.com/data-pipeline).

Experience Report

Paul Brabban | Lead Consultant

'Trust' and 'accuracy' are more nuanced than just looking for 100% accuracy. A perceived need for unachievable accuracy can paralyse teams. Clickstream-style data and log data springs to mind - it's horribly messy, and often not exactly trusted by anyone! But it is also arguably some of the most useful and valuable data that a company has, packed with potential insights about your customers and their behaviour - and it's the only place you're going to get it.

Data products should be built to be reusable

In our experience, a well-built data product will often be usable for more than the initial use-case. This is especially true for data-as-a-product. A dataset on customer habits may turn out to be useful for many departments including marketing, pricing or stock ordering. Data products that are reference or master data (e.g. product catalogues, customers etc.) are particularly likely to find lots of interested users.

Organisations should recognise this and provide means by which products can be easily surfaced, and made available to other, unanticipated users. At the same time, care is needed to avoid YAGNI issues https://martinfowler.com/bliki/Yagni.html. Build what the user in front of you needs, not the hypothetical one next year.

Experience Report

Michele Dallachiesa | Data Products & AI

When I was working on a recommender system, a really important part of the solution was getting the data for the content that we were going to recommend. The data came from a variety of sources and the raw data was not clean. So I needed to create something to clean and consolidate the raw data into a table we could use for the recommender.

This was really helpful for me, but I also found that it was helpful for other people working on recommenders. We had a sort of epiphany one day when we realised that 'we should have a separate pipeline for this data' so that managing the data logic was separated from the logic of the recommender. The data product emerged from talking to our colleagues and recognising this shared need.

After that, because more people used the data, we found further issues. Because we were thinking of it as a product, the fixes had a wider impact and improved quality for all the users. As people used it we got requests for new features like additional columns. It was interesting how, once we recognised that other users shared my needs and we started thinking of the data as a product in its own right, it became a more useful and better dataset.

Data products can be multi-modal

The same product may be surfaced into a data warehouse for dashboard users, via API for application integration, or in cloud storage for further transformation by other teams. The core data product schema stays the same throughout, with different technology applied on top without performing any modifications.

A data product should meet data governance needs

The end users are not the only people who have an important stake in the use of data products. Data is intimately tied to governance needs. However, governance is often an ambiguous term and is used to address a wide range of needs, or point to concerns which need further unpacking. In our experience, data governance can mean any of:

- Ensuring the data is secure and privacy concerns are met.
- Ensuring the data meets compliance requirements.
- Ensuring the data quality is high and the data is trustworthy.
- Making data findable.
- Providing consistent master/reference data.
- Having a strategy for data.
- Knowing what is in the data estate.
- Ensuring data is modelled, well-managed and named consistently.
- Preparing data for AI and/or automation.
- Does your organisation have the appropriate operating model and team to support data governance? Do you have appointed data leaders and data stewards?

Understanding which of these governance needs apply is an important part of data product definition.

Information security is one need which will always have to be met. We want products to be available to users but we also don't want them to leak information or lead to data breaches. Security should be a first-class citizen of any data product, and promoted across data products through governance as code. Security classifications such as public or confidential are important parts of a data product's definition, along with which user groups have permission to access it.

See also https://www.equalexperts.com/blog/our-thinking/what-is-data-governance/.

And https://www.equalexperts.com/blog/our-thinking/how-to-achieve-effective-data-governance/.

Strive for standardisation across your data products

These principles require a level of best practice across your data products. By having patterns and standard tooling for developing data products (usually driven by a data product platform), teams can deliver and support them faster, and leverage the built-in knowledge and maturity of the data products that have come before.

Start simple and build out as required, focusing on the essential and moving towards maturity over time.

These standards are usually focused on:

- Security
  - **Simple:** Reusable IaC module that embeds cloud-native policy controls around data storage.
  - **Enhanced:** Propagated data storage layer access controls (e.g. Lake Formation).
- Governance
  - **Simple:** Choose a schema format (Avro, SQL) and embed into all data product repositories.
  - **Enhanced:** Notifications of schema changes when product is deployed.
- Quality
  - **Simple:** Choose a standard data quality framework.
  - **Enhanced:** Create an alerts framework derived from data quality framework outputs.
- Interoperability
  - **Simple:** Choose a standard storage format for all exposed data products.
  - **Enhanced:** Auto-detect new products on publication, and expose to users and data catalogue tooling.
- Build & Deployment
  - **Simple:** Choose standard CI/CD and orchestration tooling.
  - **Enhanced:** Custom handling of dual version running or A/B deployment of data products.
- Naming and taxonomy
- Consistent visualisation
  - Standard chart usage.
  - Application of themes.

Making and maintaining a data product needs a cross-functional team

Creating a data product that is valuable to the business, accessible and well governed is best achieved with a small cross-functional team working closely together. You will probably also need to interact with some important stakeholders. A typical team is:

- Data users - the data analysts or business users etc. who are going to use the product.
- Data engineer(s) - the person(s) who is going to create and maintain the product.
- Business analyst - the person who understands the business needs and transfers them into requirements for the engineers.

Depending on the nature of the product and your organisation, you may also need to work with:

- Data Platform team - if you are lucky your organisation will have tooling to rapidly create high quality, secure data products. The platform team will be in charge of creating this tooling and helping you to work with it. (They shouldn't be the people to make it though - that's your job!)
- Enterprise Architecture functions - in larger organisations, the architecture function approves solution designs. Typically they will be concerned with making sure the data is properly protected and that designs meet wider technology patterns and agreements such as using the preferred cloud provider. There's a great blog at https://www.equalexperts.com/blog/our-thinking/how-to-achieve-effective-data-governance/ that describes worries that architects commonly have over data governance.
- UX professionals - if you have an insight product like a dashboard then applying User Experience techniques to the design and development of the product can improve the usability of the product.

Insights only exist in production data

We create data products because we want to find out more about our operations and our customers. In many development activities there is a clear separation between the data that is used in development (often test or sample data) and production data (the actual data). Strenuous efforts are made to restrict access to this real 'production' data, such as only allowing access for a short time after a logged service request. This is good practice in development, but as insights only come from accessing the real data, these security practices cannot be applied to the end users of the data. Analysts will not be able to find out about operations if they are only allowed access to test data.

This does not mean that anything goes. We may still need to protect aspects of the data, such as removing personal identifiers like names and addresses, or treating what is accessible somehow e.g. by only providing aggregated data for some datasets. But the security posture needs to be reasonable and understand that access to production data for the users is essential for data products.

Practices

Data product definitions should have these properties

For discoverability and management purposes a data product should have a description so that it can be found by other data users. The description should make clear what it is, how to gain access to it and how to report problems with the product. It should include at least:

- Description - what is the data about?
- How to access the data (e.g. the database view or API call).
- Who owns the data? Who do you ask to get permission to access? How do you talk to them?
- Who supports the product? Who do you email/message if you see something wrong?

We have also found other properties become more essential as the data estate grows and matures:

- Lineage information - which sources does the data come from?
- Cost tags - so you can understand and improve what your product is costing.
- Latency/speed/details of refreshing or scheduling.
- Delivery mechanism (SFTP, etc.).

The definitions should be in a place where users can search to find the data they need, such as a data catalogue.

Make your data products easy to find - make them discoverable

It is common for a data product to be useful to more than just the initial user base. Most data products developed for one particular use are found to be applicable for others. Sometimes this happens by word of mouth, when a data analyst recommends data to a colleague. But this limits the scale to which the data can be used. It is much more powerful if potential users are able to easily find data products that can help them.

This function can be provided by a data cataloguing tool. There are many tools available, including some good open-source ones. Bear in mind that if the metadata for the catalogue is manually provided it can quickly go out of date, so where possible, automate the creation of data catalogue entries. For example, we have found it useful to embed publicly consumable documentation in the codebase, which can then be linked to business terms to make the documentation discoverable.

Data catalogues can be as simple as a static website showing data sources in markdown - which may be the place you start. Other solutions we have used include Google Data Catalog (https://cloud.google.com/data-catalog/docs/concepts/overview) and Datahub (https://datahub.io) which is an open source catalogue.

A data catalogue is itself something that should meet the needs of a variety of different users. In a complex data environment, applying product thinking to the catalogue and its users can be beneficial (see https://www.equalexperts.com/blog/tech-focus/lessons-data-governance-product-thinking/).

Experience Report

Andrew Taylor | Data & Product Manager

At the start we had minimal formalised data governance and no data cataloguing at all. Documentation about data was stored in various places, leading to people reinventing the wheel and developing SQL code to get data because it was easier than finding pre-existing data products. The closest thing to a data catalogue was an audit report of hundreds of known critical reports used across the business.

The business realised the shortcomings of this approach in terms of both data governance and treating data as an asset. A data governance function was set up and a third party tool procured to help with metadata management and data quality. I think it's really important to remember that data governance is not just a technical problem and requires the organisational structure, support and ownership to implement it properly. That said, you still need to add the metadata into the tool, take care of the data quality and make it easy for users to find the data they need. We found that we needed to build repeatable processes around the tooling so that we could easily and reliably import the metadata.

Deliver your data products incrementally

Incremental delivery - the practice of delivering software in small chunks of end-to-end functionality - has brought many benefits to software delivery in other areas (see https://www.rebelscrum.site/post/the-value-of-incremental-delivery-in-scrum). We have found that it also works very well for data products. In an incremental approach, the product team starts with identifying a first thin slice of the data that will provide some value to the users. This may be part of the intended dataset (such as a table with enough data columns to answer one business question) or a dashboard showing some high-level data. We often refer to this as the 'steel thread.' New product features such as additional data features or improved observability are then added piece by piece, rapidly improving the product capability.

As the team develops the first simple version of the product, they uncover any unknown technical and organisational challenges and resolve them early on. A quick feedback cycle gives data users early sight of the products, so that misunderstandings and unknown needs can be identified quickly. Incremental delivery means we are continually ensuring the correct direction of travel with the users and other stakeholders. It also means the team is able to build confidence in its ability to change robustly right from the start.

Incremental development also provides value quicker. Rather than waiting for the whole product to be ready before any insights can be generated, your understanding of at least part of the business is improved from the first iteration.

Apply continuous delivery and testing where you can

All data products depend on an end-to-end data pipeline. These are hidden from the users but are in fact complex pieces of software. We believe continuous delivery techniques are necessary to create robust and trustworthy pipelines, and thus robust and trustworthy products. (See our data pipeline playbook https://playbooks.equalexperts.com/data-pipeline for more details.)

Data products with a strong visual element, such as dashboards, typically use third party visualisation tools such as Tableau, PowerBI etc. Including these within a continuous delivery or CI/CD build environment can be challenging. These visualisation tools typically utilise point-and-click interfaces with some coding to make more complex features. As point-and-click tooling, they do not support the full level of traditional testing you'd expect from code in a CI/CD pipeline, even with expensive licensing costs.

This creates a tension between where the processing should be managed with code in a CI/CD pipeline and where you should use the visualisation tool - trading off testing capabilities against better visualisation features.

If you are building a dashboard that needs complex calculations for business metrics across multiple products, you could do all the metric calculation and aggregation in your ETL pipeline, but this means that your visualisations are limited to just displaying data rather than empowering users to explore that data. However, if you do all the calculations in your visualisation tool then errors are harder to track down and have to be manually tested at each change of the dashboard.

It is difficult to give hard and fast rules for this, but in general we recommend:

- Complex calculations should be done where possible in code deployed in a CI/CD pipeline where its functionality can be properly tested.
- Aspects like aggregations should be done (where practical) in your visualisation tool to enable dynamic filtering and drill throughs.

Finding the balance between these two tools is vital so that users can explore data themselves, whilst maintaining data integrity and confidence in metric calculations.

Data quality is an iterative process

The quality requirements for a data product depend on the needs of the users. Understanding these needs and specifying how they should be met are important parts of data product specification and an important part of the data product owner's role. Data quality challenges can be a result of poor practices anywhere in the pipeline from poor data collection to problems in transforming values within the data pipeline.

It is usually very difficult to identify all the problems with the data up-front, so do not expect everything to be defined at the start or in an initial phase. Improving data quality is usually an iterative process in which initial samples of the data show common data issues that are resolved in the first or early versions of the product. You should expect to learn about your data. For example, as the product is used, rarer quality issues are encountered. These can be resolved depending on the impact. So it's okay to go to production early and learn about your data as soon as you can!

Some typical data quality issues are:

- Variations in formatting of common items (e.g. phone numbers with and without country codes or dates in a variety of formats).
- All values set to a single value.
- Values are out of expected bounds (e.g. prices which are in millions of dollars).
- Null or default values (e.g. system time - 1 January 1970).
- Duplicate data.
- Mixed data types or changes of data types in the middle of a dataset.
- Impossible sequence or non-monotonic data (e.g. death dates before birth dates).
- Lack of indexes or changes in identifiers/matching terms.
- Incorrect data entry.

In mature environments you can apply tools to profile data so that changes are spotted early on and data quality issues can be identified and resolved quickly. In a similar vein, it is common to find tests in production environments that check for changes in the data. (Something that is different from standard software development.)

What lower environments are for/can test

We recommend using a continuous delivery approach for data, with testing in lower development and test/QA environments. We use these to test our implementations of:

- Infrastructure - does our terraform code work how we expect?
- Dependencies - does this step in the data flow integrate with upstream data sources or downstream users?
- Transformations - the scripts used for data manipulation e.g. conformation to a data model can be tested with test data.

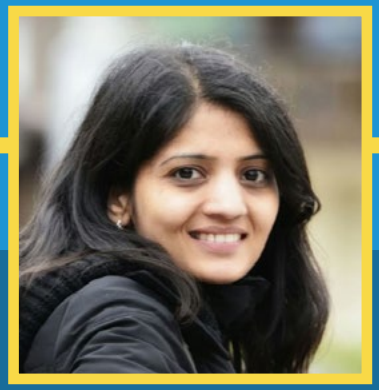
As noted above, this won't capture all the issues because the data can change, so testing in the production environment is common for data products.

Data personas can improve understanding of users as well as product focus and prioritisation

During product development, personas are a great way to understand who the users are. They are an effective tool to understand different user groups and their needs, and for communicating which types of user a feature is being developed for. Creating data personas has the added benefit of helping identify unexpected stakeholders who may not have been otherwise engaged, as well as understanding wider user needs such as accessibility.

Data products can have different types of user (see who are the users of a data product? on page 7) and creating personas for these types is a great way to understand and communicate their various goals and needs. For example, a senior manager might need an aggregated view of a dataset, such as weekly sales or year-on-year comparisons, while an analyst might use the same dataset to see when and where sales happened, or which stores are performing well. Understanding different user groups and how they use data helps to define data better and can even guide you to the creation of separate products.

Experience Report



Chandan Pawar | Technical Business Analyst

We needed to create reports for a large number of metrics across the business to a tight timescale. We had to find out which reports we should focus on first, so we created a number of personas around different users in the business. We identified that exec level metrics had the most impact and were the most critical need. Focusing on this persona really helped me find the right scope and have the right conversations with the stakeholders, so we could focus on something achievable. It also really helped us explain these decisions to our stakeholders.

Product visions are a helpful tool to align teams and communicate with stakeholders

Product visions are a very useful tool when creating software products and we have found them useful when creating data products. At the start of product development they help cross-functional teams understand all aspects of the data product and what the benefits will be, which helps with alignment and prioritisation. A sample product vision based on a real data product is:

> *The daily sales data product will be more sustainable and enable us to reduce data warehousing costs and support time. It will provide a highly secure, resilient environment giving great access and visibility to all users. The daily sales data product will give analysts more opportunities to understand sales patterns and lead to improved revenue.*

Product visions are also helpful in communicating work to wider stakeholders. You can see that the vision includes non-functional requirements in a way that brings them to life for non-technical stakeholders - a data product does not just share the dataset, it must also be secure and resilient.

Entities that cross domains need naming standards first

Aspects of your data product are cross-domain. For example, a shop location might occur in a data product for daily sales but might also occur in a data product for logistics. These entities are critical for joining datasets. Users may want to combine data on sales with data on logistics to improve logistic deliveries or to see if availability is impacting sales, for example.

Those parts of the data product on which joins are made (in this case shop location) should be standardised. They should have the same names, or at least have a standard format, so that they can be joined. At a later part of the life cycle, if the names need to be changed, they will have a bigger impact radius.

Make metadata manageable

When building data products for the first time, new ways of working will probably need to be established. Since these will likely be iterative in nature, it is very easy for metadata to start living in a number of different places. For example, a metric definition may start life in a spreadsheet, then be part of a user story in a tool, be developed and documented in the software tooling of choice, and then the final version gets surfaced to an end user. You may also have metadata in a data catalogue and possibly even in an enterprise data management solution.

Maintaining consistency across these disparate systems can be a difficult exercise, especially as definitions are updated, code is refactored and so on. A suggestion is to have a master source for the metadata and to assign a metric identifier against these. This way, definitions can change, but the identifier remains the key across any systems where the data point has been referenced. As per database design, the key should exist, but to avoid misuse, should not have meaning. Furthermore, the ideal is to have a loose coupling to any other systems so that any updates in the master source can be programmatically updated elsewhere to prevent maintenance headaches.

Understanding and defining user access roles are key activities of the business analyst and product owner

Protecting information security is critical for any data product. A lot of complexity around implementation is down to ensuring that data is secure and well protected. Defining user access is a really important part of this. The data product owner should ultimately decide who can access the data and is also responsible for defining which user access roles can access the product. Don't pass this off to a separate team or make all user access definitions the job of one person. They will not have the context to understand who needs to use the data, or if there are any particular sensitivities around access.

Rather than giving access directly to individuals, adopt a role-based access approach. Work out what roles (e.g. customer team) may want to access the data and assign individuals to those roles. This makes access management more manageable. Personas can be very helpful in defining roles and user groups.

Build computational guard-rails to enable continuous governance

We have noted that a data product should meet data governance needs on page 20, and that these needs vary from product to product. We believe that many of these needs are best met by creating guard-rails and enabling governance as code. For example:

- Creating the product involves making a data definition, which can then be published as metadata to catalogues and similar tools.
- Data quality rules and validation can be met through specific dq tooling (e.g. AWS DeeQu, Great Expectations).
- Personally Identifiable Information - PII leakage should be protected against. There are many tools that can help with this (e.g. AWS Macie, GCP DLP https://cloud.google.com/dlp).
- Applying continuous delivery techniques and using a build pipeline with tests is essential for ensuring that the data is well managed and that changes to data products do not reduce data quality or reliability.
- Automated data deletion - governance may well specify needs around how long data can be held and when it should be deleted. This is something that can be easily automated.

Experience Report

Michael Czerwinski | Security Principal

When creating data products you want to ensure a secure path to production, which defines how data products reach production and their subsequent lifecycle. Create guard-rails that implement these security policies as code. This baseline should encode your standard security practices such as vulnerability scanning, secret detection, static and dynamic security testing etc.

Establishing common patterns makes it much easier for infosec to evaluate the risk for new data products. Exposing interfaces for common needs also makes it easy for developers to implement these policies.

Use tried and tested methods for how you deal with data product changes

An important, but sometimes overlooked, aspect of data products is that they usually change over time. Users ask for new features from the data, or upstream data sources change - and these changes impact the interface to the product. If this is not managed well, then downstream users are affected and may lose some or all access to the data. Fortunately, changes to interfaces happen all in software systems and these techniques can also be applied to data products:

- Data contracts - in software development it has been very helpful to think about products as having a contract with downstream software or users. This has lots of benefits including making clear and explicit what data is transferred, and being able to test against the contract. This approach can be applied to data products and means that developers can validate whether a change will break the contract and whether you will need to alert downstream users (see https://tempered.works/posts/2023-05-19-dbt-contracts-in-sql/ for a description of using SQL and DBT to manage data contracts).
- Usage monitoring - allows you to see which user groups or downstream services are using your product. These are the people who need to be alerted ahead of any changes.
- Public interfaces vs. private implementations - know what is exposed to the users and is part of the contract with them, and what is an implementation detail, which can be changed without impacting end users.
- Expand/contract - is a well known and proven pattern for managing change of interfaces in software. In the expand phase you create a new version of the interface that runs alongside the old one. Users of the original version of the interface are migrated onto the new version. (This requires knowing who is using the interface - see usage monitoring above). When this is complete the old interface is stopped - this is the contract phase. (See https://martinfowler.com/bliki/ParallelChange.html for more details on expand/contract.)

Start your product with a single user in mind, but build for reusability

Thinking of data in product terms brings lots of benefits. It underlines the fact that data has users and that the product owner's role is to meet the needs of these users. It also promotes reusability of the data - the discipline of developing and operating a true product makes something more robust and more easily used by unanticipated data users.

On the other hand, like any product, if you start trying to meet the needs of all possible users, you will create features which no-one in fact needs. The development will take much longer (and may even never finish) and will be much slower to get to value. We strongly believe that you should avoid the hypothetical user who needs all the features, focus on actual users with real needs, and start your product with a single user in mind. If you are unable to find someone who will be the user of the data product, then it's time to question whether there is any need for it at all.

There is a risk with this approach that the product becomes so tightly tied to the needs of a single user group that it cannot be reused. You should try to create the product so that it can be reused.

Some practical ways to achieve this are:

- Try to use names that resonate with initial users but can be understood more widely.
- Make sure access to the data is integrated with wider organisational access control mechanisms.
- If you can, discuss names (variables, columns etc.) with a content designer. Ask them - how does this resonate with someone who is not a domain expert?
- Invest in tests - this allows you to amend the model whilst ensuring you still meet the needs of the initial users.

Provide a channel for users to find out about the product and talk to the data product team

Throughout the life of the data, users will have questions for the team about what is in the data and how it's accessed, and they may want to report problems with the data. Create a front door where (potential) users can talk to you about these things - you can use a Slack or Teams channel for example. You can also help users by documenting early (providing descriptions, etc.) but don't obsess over it. It doesn't have to be complete to be useful.

Experience Report

Liz Leakey | Head of Design

During an inception for a new data product, we took a user centred approach and created user profiles for data scientists and data analysts as well as the engineers who were supporting the product. We created journey maps for our users to understand the uses cases, or how they used the data in their day to day work. It was interesting to see that, in order to create the maps, we needed to understand the journey of the data as well as the user experience. We discovered that it arrived in AWS S3 and needed to be manually checked every morning. This really highlighted an unmet need around reliability of having fresh data - there were lots of manual checks to see if it was up to date or not. With limited alerting in place, the user was often the first to know if there was a problem, before the team owning the data, which is not ideal.

Alongside the profiles and maps, we ran a product vision exercise - why were we creating this product? This led to a really good articulation of the business benefits - the why, and helped reframe the work from being purely about migration and data engineering, and being clearer about the business outcomes it supported. The product vision created alignment for the team, giving them an eye on the user needs and business benefits at the start, as well as being a great tool for communicating to stakeholders.

I found that while these design thinking tools and product techniques are standard in software development, in the data world it was more unusual and people were less familiar with the terminology or rationale for some of these activities. Data products might not have the same end-to-end experience as a retail product or government service but there are parallels when it comes to meeting user needs. Being able to self serve for example - finding the data (discoverability), getting permissions and gaining access to it, as well as having alerting and monitoring in places, to understand if there is a problem. Core needs around reliability of the service, quality and speed to complete a task are also very relevant.

When to build a data product platform

As you find success with your steel thread data product, and begin to create multiple data products across multiple teams, commonalities will begin to naturally emerge such as:

- Reused infrastructure & IaC modules.
- Orchestration and CI/CD pipelines.
- Standards around storage, interoperability and testing.
- Common security and compliance/ governance needs.

The teams may be looking after these individually, or sharing them as a community. Due to a lack of ownership, you may see the following:

- Out-of-date or consistently broken common tools and infrastructure - impacting delivery.
- Multiple formats for consuming data products, increasing workload for consumers.
- Lack of security knowledge within teams, or only responding when there is an incident.
- No clear understanding on how compliance is being met across teams, or a view across data quality.

Now's the time to consider building a self-serve data product platform.

By taking ownership of the data product development experience, a platform team can enable:

- **Faster delivery and experimentation** by building data product delivery tooling and pipelines for quick spin up (and tear down) of data product projects.
- **Lower product team cognitive load** by having platform concerns owned by one team instead of many.
- **Increased data quality** by having standard data quality testing frameworks, with platform enabled monitoring and alerting out of the box.
- **Out-of-the-box security compliance and data governance** embedded as code within the platform - so all data products inherit base security and compliance requirements on project setup, with common compliance tools available e.g. GDPR data tagging and deletion tools.
- **Enforced standards and best practices** by using code-based guard-rails to check adherence e.g. templates & libraries for building data products with different modal needs.
- **Increased response to change** by providing telemetry around products e.g. schema change notification tooling, data quality telemetry.
- **A global view of adherence to governance** by implementing security and compliance checks, providing assurance wider organisational goals are being met.

For more on building platforms, see our digital products playbook https://playbooks.equalexperts.com/digital-platform .

Explore

Our playbooks are collections of observations that we have made many times in different sectors and clients. However, there are some emerging technologies and approaches that we have only applied in one or two places to date, but which we think are really promising. We think they will become recommended practices in the future - or are at least worth experimenting with. For now we are recommending you explore them at least.

Should you provide self-service data access?

The easier it is to provide access, the quicker data will flow through the business and generate more insights. One way to accelerate access to data is through self-service, which we seek to provide wherever possible. But we have seen examples where it has been the wrong thing to do. Different users have different skills and needs, and different users know the data and understand domains differently. Senior leaders will often want shrink-wrapped reports tied to a specific business requirement and don't need access to the underlying data. Data analysts or scientists, on the other hand, will be very capable of working with raw data and repurposing it for their own needs. You need to explore what is right for your data products.

Letting other teams work on the data product

After a data product has gone live, the development team will move onto other products or projects. But there are likely to be requests for changes to the product. We want to avoid users suffering from long waits for these changes. One way of expediting changes is to let other teams work on your code base if they need something done that the original doesn't have capacity (or priority) to do. This approach has been applied successfully for micro-services. This way, the data product team is more of a custodian than a "sole owner". Of course, any work should be agreed with the data product owner first.

Improve your data product descriptions and show similar products (if you can)

In a large data estate it is difficult for potential users to find the data products that they want. Typically this is addressed using discoverability tools like data catalogues. However, it is easy to create dry descriptions of the products that may be technically accurate, but will not resonate with wider users. You should be user centred and strive to use language that is discoverable and can reach the audience on a level they understand. Experiment with descriptions that include the sorts of problems the data product is trying to solve, as well as what is in the data.

Likewise, users often find being able to search for similar products a good way to find what they are looking for. If your tooling allows this, enable this functionality.

Pitfalls

Experience Report

Neha Datt | Product Director

As a product person, it's been really interesting to apply product thinking to data. The approach in data is often fundamentally different to how we build consumer products and platforms.

I've seen many data product teams focus on the tech before understanding the value proposition. Teams have then ended up with data products with poor adoption (ie few people using them) and unsustainable costs (exorbitant licensing).

*Data product owners would really benefit from using the product "DVF framework". This asks: is this product **D**esirable (because it solves an important user need), **V**iable (because it creates the **right value** for the business) and **F**easible (it can be built/maintained sustainably)?*

*In product, we answer these questions **before** buying or building any tech.*

I find Viability especially interesting. It's generally rare to measure the return on investment (ROI) of internal products in a business as there's often no impact or consequence if the business value isn't realised.

In contrast, product teams typically have service dashboards that measure business value daily: how many people are using our product, how useful is the data, etc...And this drives product funding and improvements.

Plus product teams tend to take a leaner approach which data product teams could benefit from. Instead of capturing all the data and building all the capabilities, focus on primary and secondary users. Build the minimum to get to value faster. Rinse and repeat!

Building products for engineers rather than for users

Just as data products should be driven by the needs of the business, they should also be designed to be used by the business users.

Seeing a dataset as the end result of an ETL, or for feeding into applications, can lead to data as a product being designed for technical optimisation over user friendliness. A balance needs to be struck between the two so that end users can query the dataset in a fast, self discovery manner that is also performant.

Some signs your data product has been built for engineers instead of end users include:

- **Nested data and arrays** - these are hard for users to parse with SQL, but easy in code.
- **Columns with confusing or abbreviated names** - shorthand used by developers but not self-explanatory.
- **Types that require casting for use in dashboards** - more extra hoops for users to jump through to create queries.
- **Lots of columns** - easy to work with programmatically, but difficult to navigate in query editors and data catalogues. Look to break the product down into smaller products joinable on domain entities.

Treating a technology as a data product

We often see organisations treating key infrastructure components, SaaS based services or technical approaches as data products. For example, people refer to the database solution or the data catalogue solution or a particular database as a product that needs a product owner, or a particular team is '*The Graph Database Team*'. We do not consider these to be data products. They may well be part of the technical solution for the data product, but do not in themselves meet the business needs of understanding their operations or customers better. Additionally, they can be switched out for alternatives if needed or desired. Treating them as a product leads to a technology-focused view of the data estate rather than a business-focused view.

Of course they need to be configured and managed, so will need service managers dedicated to operating and supporting them, but these are not product owners.

Believing data discoverability can be achieved out of the box

You need appropriate tooling to enable users to understand what data you have in your estate and find the data products they need. This will typically be a data catalogue. But please don't buy a catalogue, sit back and expect it to solve all your discoverability needs. It will only be as useful as the metadata it has captured, and then only if users know that the tool exists and can help them. It is common to see data catalogues with only rudimentary data, which is unlikely to meet the needs of people looking for data. Do invest in capturing the metadata and governing and promoting the metadata as useful. Treat that metadata as a first-class citizen! Where possible, automate the provision of metadata so it is up to date.

Likewise, it is common to have one person whose role is to collect the metadata for all the data products. We think this is an anti-pattern. The people who know most about the data are the people who have created it. Data discoverability is an important part of making the product and should be owned by the product owner. In mature data environments you may even publish data quality metrics to the data catalogue, so that users are able to self-serve from a single place.

(See https://www.equalexperts.com/blog/tech-focus/lessons-data-governance-product-thinking/ for an example of taking the data catalogue seriously as a key service.)

Insufficient focus on data quality

If data quality is not a focus, and the data becomes untrusted, then it will not be used for decision making and has no business value. Or worse, if similar/overlapping data products show discrepant results this will lead to lots of effort working out which is the true answer, creating negative business value. Many organisations have found to their cost that once trust in a data product is lost it is hard to regain.

Data products are owned by a centralised enterprise function

It is common to see organisations in which the creation and delivery of data products is provided wholly by a central data function. We do not recommend this approach because it distances the understanding and knowledge of the data from the people who own and maintain it. It also leads to backlog buildup, frustrated data users, and frustrated data engineers, as the complexities of meeting everyone's needs and ensuring appropriate information governance, as well as a lack of self-service, make it hard to ingest new data sources.

With these organisational structures we have also seen that:

- Data users do not have substantial contributions to the definition of the data product.
- Data catalogue entries are left to one person - who has to create all the entries.
- The focus is on the technology solution rather than the business need.

We want to make clear that there is still an important role for a data function. Cross-cutting data platform services are often delivered by a single platform team. But the focus must always be on developing self-service tooling and APIs that allow data product teams to leverage these platform capabilities autonomously. As demand/backlog increases, more specialised teams should be spun out that focus on particular aspects of the platform, e.g. telemetry or build & deploy.

Similarly, data governance policy and standards should be encoded in the platform as governance as code API/tools/services. These provide the governance guard-rails and abstractions to product teams.

There is a distinction between defining enterprise data governance standards, which is likely owned by an enterprise governance team, and their delivery/automation.

Don't process and expose data just because you have it

It can be tempting to learn about data sources and decide to simply expose the data. The data looks interesting - surely someone will make use of it? We don't recommend exposing any data without understanding the users and then managing it properly. There will always be some effort in exposing the data, and more to keep the data available. Without talking to end users you will not know what transforms or quality measures you should apply to the data to make it useful, and you probably won't know how best to describe it so that end users can understand what it is and what it can do for them.

If this is done with too many datasets, then you end up with many poor quality data products with no users, costing money and effort to maintain, with no value to the business. In the worst case it just confuses data users, devaluing data products that have been properly developed and curated.

Contributors

Andrew Taylor

Austin Poulton

Barry Hostead

Brian Mason

Chandan Pawar

Gill Gibson-Piggott

Liz Leakey

Michele Dallachiesa

Michael Czerwinsk

Munish Malik

Nathan Roche

Neha Datt

Paul Brabban

Scott Cutts

Simon Case

Our playbooks

We have produced a compilation of playbooks filled with good practices based on the experience of many Equal Experts people who worked over the years with hundreds of customers. They are a great example of the power of the Equal Experts network.

One of our shared principles is that we value a passion for learning over knowing all the answers. We don't pretend to know all the answers – but we are confident in our ability to find them. We do not hire very experienced people because they know everything, but rather because they are better at knowing how and where to look for the answers.

We trade on our ability to learn and share knowledge, rather than protecting or 'guarding' it. At its core, Equal Experts is a haven where this sharing happens freely and happily, between like-minded practitioners and our customers. This is why we felt compelled to open-source these playbooks, so that this knowledge can be shared as widely as possible.

equalexperts.com/playbooks/

We hope you find these playbooks as useful as we do.

About Equal Experts

At Equal Experts we don't just build things; we work to upskill teams in a process of enablement, leaving behind a learning culture. Engaging 3,000 consultants across 5 continents, we help create leading digital products and services. We decide with data, design for users, deliver at pace and scale sustainably.

If you'd like to learn more about Equal Experts get in touch.

EQUAL
EXPERTS