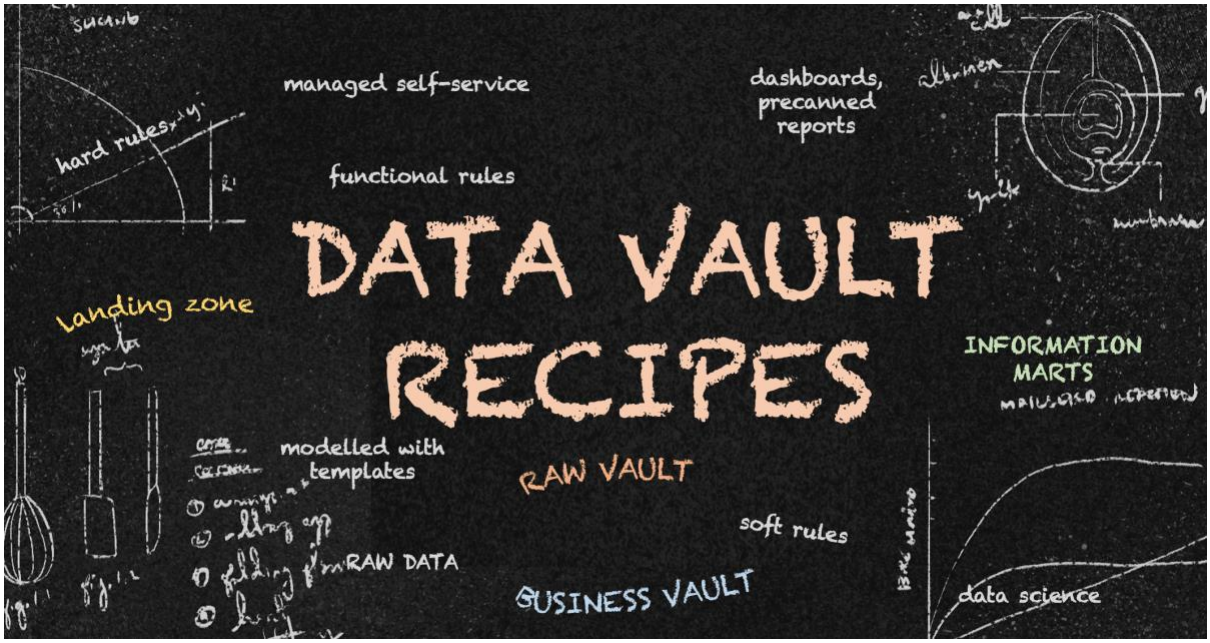


To me if I think about Data Vault 2.0, I think about a proven, repeatable methodology that with the right ***ingredients*** and knowing ***how to mix*** them together, will deliver the outcomes *you put the effort* into, like *baking bread*!
A Data Vault (like baking) is not developed in isolation; it is reliant on people to be invested in the process and responsible to its outcomes, let's list who they are.

- **Enterprise (data) architect** (recommended-input) - to provide an overall picture of the enterprise business and data landscape. This role will also be aware of the current and future business capabilities, data governance practices. Providing clear definitions of important business roles, functions, policies, and interactions and where they see the data warehouse as a part of the enterprise ***capabilities***.
- **Data architect** / **data steward** (required-input/responsible) - in liaison with the EA the DA is aware of the current principles, policies, and practices within the data architecture space as well as data governance, data lineage, metadata management,
- **Solution architect** (recommended-requirements/responsible) - a role familiar with current technology landscape and with the business hat on, the SA can also assist in identifying technology stack requirements (ex. ELT/ETL), Cloud infrastructure, data flows
- **Technical business analyst** (required-requirements) - the role closest to the business requirements and an understanding of how the current data landscape is defined and what the pain points are. This role will know the parts of the business processes that are not automated and perhaps should be.
- **Business users** (optional-requirements) - the business user typically has a day job but can be included to help clarify business requirements.
- **Source-system subject matter expert (SME)** (required-input) - source systems capture the outcome of business processes against business keys (object) and units-of-work. The SS-SME can explain how the business processes are modelled in the source system / application, grain of data available, technical gaps that can be solved by the source, limitations of the source system and advise on future changes of the platform as well as how to integrate with the source-system. Without a SS-SME we are guessing what the

source system does, if an SS-SME is not available then this must be highlighted as a risk to the project, and we will likely rely on the technical BA for information in this space.

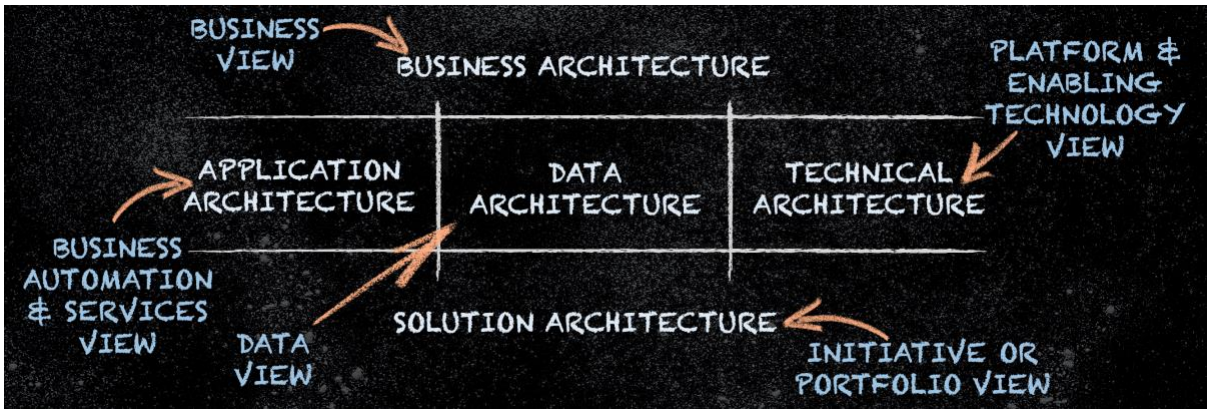
- **Data modeler** (required-responsible) - a role with the understanding of data modelling concepts, the data modeler knows the difference between third-normal form (3NF), dimensional and data vault modelling and strengths and weaknesses of each. 3NF is often needed to understand the thought process and availability of data from source systems, Kimball / Dimensional modelers understand how the data should be presented to the business to deliver business value. A data vault modeler understands the decoupled components needed to automate business process output within raw and business vault.
- **Product owner/Business sponsor** (required-responsible) - the role behind the engagement, ultimately the role you need to keep happy, paying the bills. This role must have enough clout within the organization to drive these types of initiatives and ultimately the role making the decisions. It is non-negotiable that this role exists, the role will be interested in the when and metrics showing the ROI & TCO.
- **Agile coach / Project manager** (required-responsible) - epics, stories, tasks, the role will want to see a breakdown of the overall areas being delivered (technical, business, data, automation, infrastructure …) and does not necessarily need to know how. This role keeps the progress ticking and raises any concerns issues to the relevant stakeholders.

- See "Mob Modelling", bit.ly/3zgP7OP

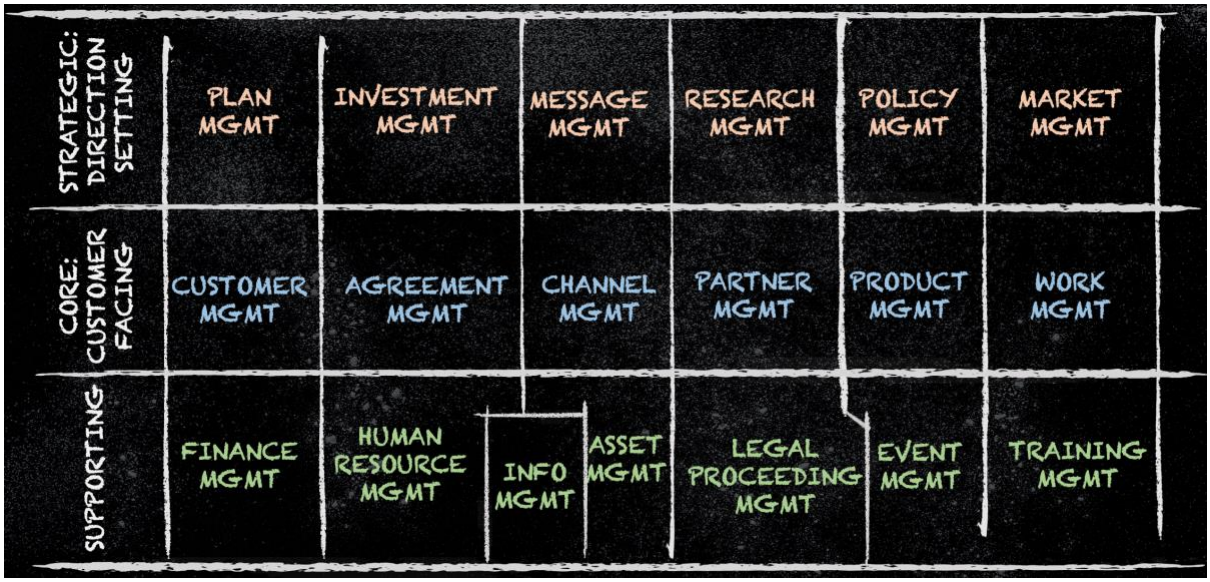
Everything in its Right Place

Data Vault 2.0 is not hard, with the minimum steps you can get the right mix to build a successful data analytics platform based on **patterns**. A *simple* recipe is easily repeatable, in contrast a recipe with pages and pages of ingredients and steps is not only difficult to memorize, but also too delicate to scale, *like a house of cards*. A single missed step puts the rest of the outcomes at risk.

Understand that for a business to thrive it must have an **enterprise vision** (recipe outcome), so too must the data represent what the business is invested in, then it stands to reason that the data should be modelled within the organization's **capabilities**, value streams, **business processes** and **business objects**. The first block of enterprise architecture is **business architecture** and capabilities are based on business objects.



Business Architecture: Putting Business into Enterprise Architecture, Ulrich, W. & Soley, R., Feb 2016, CIO Review



Sample capability map, *or Business DNA*

As a business becomes more successful the more it needs software to automate its business processes. But a business does not purchase software that does not fit its **business model**, it will pick software that best suites its own business processes, managing business objects through its **value streams** and processing it through **automated business rules**. Every step of the way there are metrics and dimensions to be measured, after all you, "if you cannot measure it (business process) you cannot improve it" - Lord Kelvin. Improvements in the business means that the enterprise data model must be designed to be adaptive and simple to change. Keeping it simple and *representative* (to the business model) must be the goal, otherwise maintaining the platform becomes an accumulating cost (a technical debt tax) to the business itself! A symbiotic relationship between business and the data platform.

Without further ado, let's introduce the components of the data vault methodology.

Raw Vault

Designed to **passively integrate** by business keys, RAW Vault is made up of the following three simple ingredients:

1. **Raw Hubs** – business objects, place, person or a thing like an agreement, contract, account, product, order. These all have **identifying business keys** that your business can relate to, and may be that these immutable values are used in discussions with your customers… they are the **business objects**.
2. **Raw Links** – the **unit of work**, the relationship between two or more business objects as so mapped by your business processes, operating models, and value streams. Of course, they can exist within or across business units and partners, but these are the things that interact with each other, that together exist are a part of a business process and/or value stream that can be isolated within a **business unit** or exist across business units.
3. **Raw Satellites** – business processes (automated by applications) and objects shed details and attributes about their state, as their states change, we capture that state within these satellite tables. It is not necessary that it might be a single state (multi-active), descriptive information can be in multiple states at the same time for a relationship or business object. A satellite describing a business object is called a hub-satellite and a satellite describing a relationship is called a link-satellite.

For all three we track the record source, load date of the record, applied date of the record and there can be variable application of the above two table types, (no not the hub table) that tracks more intelligence of the source applications. These are:

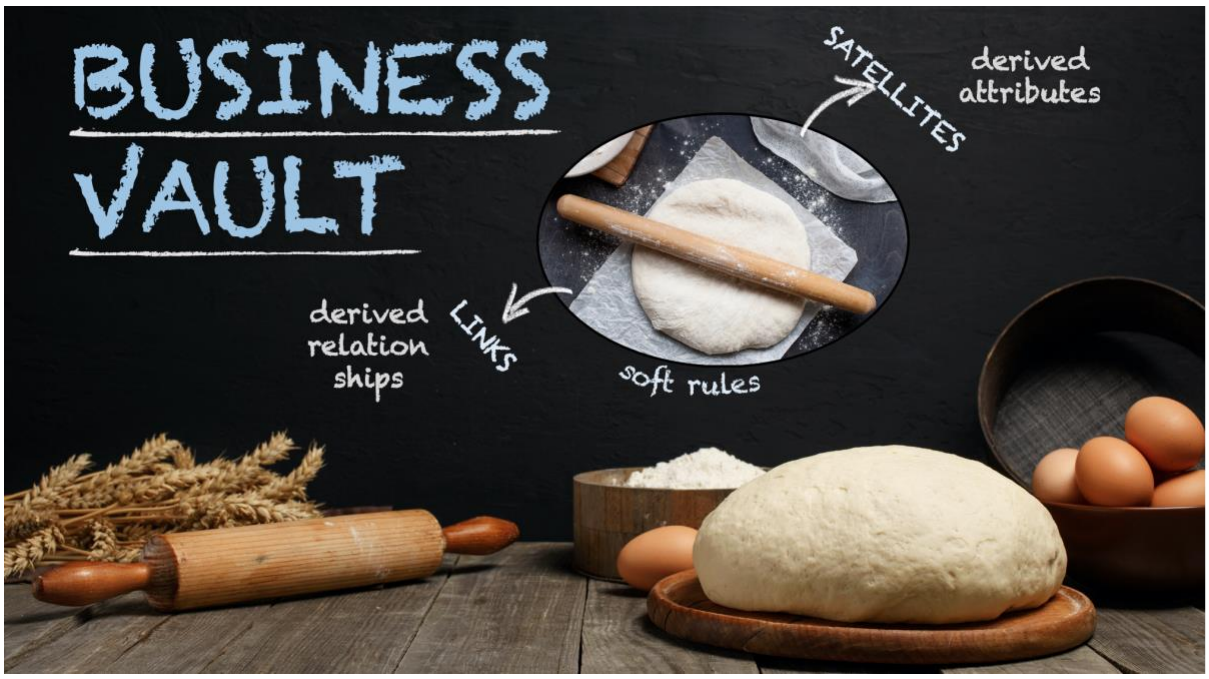
- **Links / Satellites with dependent child keys** for tracking categorization of the parent key
- **Multi-active Satellites** to deal with changes in a SET of records
- **Record Tracking Satellites** to detail the *last seen* occurrence of a business object or relationship.
- **Status Tracking Satellites** to track if the business object or relationship is new, updated or deleted; and

- **Effectivity Satellites** to track the movement of a principal **driver** of a relationship when the source application does not do the same.

To explore more of Raw Vault, peruse these links
- See how to model a Raw Vault, bit.ly/3wZbGql
- Passive integration, bit.ly/3xlFK0s
- How Data Vault promises no refactoring, bit.ly/3tPI66B
- How to build a self-healing data vault, bit.ly/3y4mUdV

What if the source application supplying the raw data does not match the business' view of the business process?

Business Vault



A business vault is ***sparsely*** modelled… why? Too often a business vault is used to solve **technical debt** (see: bit.ly/315q83U) when in fact tech debt *should* be solved at the source! Analytics teams have taken the brunt of poor source applications and its many shortcomings. Instead, a business vault is used to complete the business process when a source *cannot* be modified and where derived content needs the same **auditability** as raw vault guarantees.

A business vault will often contain:

1. **Derived Links** – the **unit of work**, supplied in raw vault might not meet the requirements of the business and it might not be feasible to have the source modify their more generic industry business process to change. The business vault link provides the *business view* of the relationship and may be deployed as *exploration* links. If the enterprise depends on that link and an audit trail is needed, then the link ceases to be exploratory.
2. **Derived Satellites** – providing the auditable derived attributes about business objects, and relationships. Its parent table could be either a raw vault hub, a raw vault link or even a business vault link. Any analytics derived from raw or other business vault objects and needs auditability must be persisted as business vault

links and/or satellites. Business rules deployed in this area must be **idempotent**, lack of this quality makes the business rule itself non-repeatable and just be a point of technical debt.

Notice that there are *two* tables in business vault and not three, a business vault *hub* implies that the business key was derived in the analytics platform and not in the source… that is an incorrect implementation of hubs! Business keys are manufactured by source systems and applications (by the defined business process itself) and not within the data vault! *Everything in the right place…*

Also note that the business vault *is not* a separate entity or area even though the name vault implies a separation of data sinks; no, a business vault extends the raw vault model with derived content, and they must always be separate only by table name. Do not model derived content in the same artefacts as raw vault artefacts. *Separation of concern* is key here to keep the overall data vault model adaptable to change. What does this mean? Within the same database and schema, you will find **both** raw and business vault.

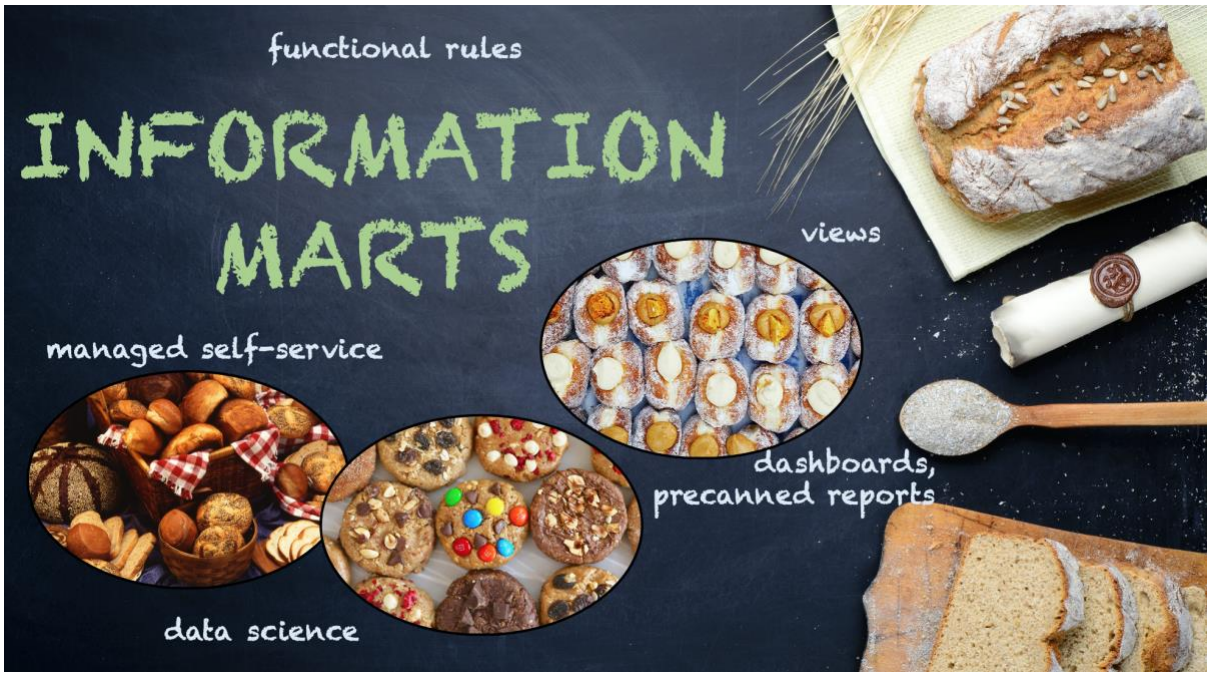
- Business Vault, bit.ly/3EtPUQk

An often-quoted adage about data scientists is that they spend most of their time cleaning and organizing data; if the data is being persisted for and remain auditable then why not persist the data into business vault? This process is termed **data preparation** which is a set of repeatable **feature engineering** techniques used to prepare data for machine learning algorithms, some of the techniques below overlap with **data quality** and are used in combinations of

- **Imputation** – missing values, discussed under data quality section below
- **Outliers** – detection and the *means* to deal with them
- **Binning** – aka banding or drilling up data to a higher category
- **Log transform** – deals with outliers and *normalizes* the data
- **One-hot encoding** – unpivot data to flags of boolean values
- **Grouping operations** – such as averaging, min, max
- **Feature splits** – grouping usable attributes together
- **Scaling** – normalizing and standardizing attributes for an algorithm
- **Extracting the date** – into groupings like year, month day columns

Fundamental Techniques of Feature engineering, bit.ly/3EzfzHl

Information Marts



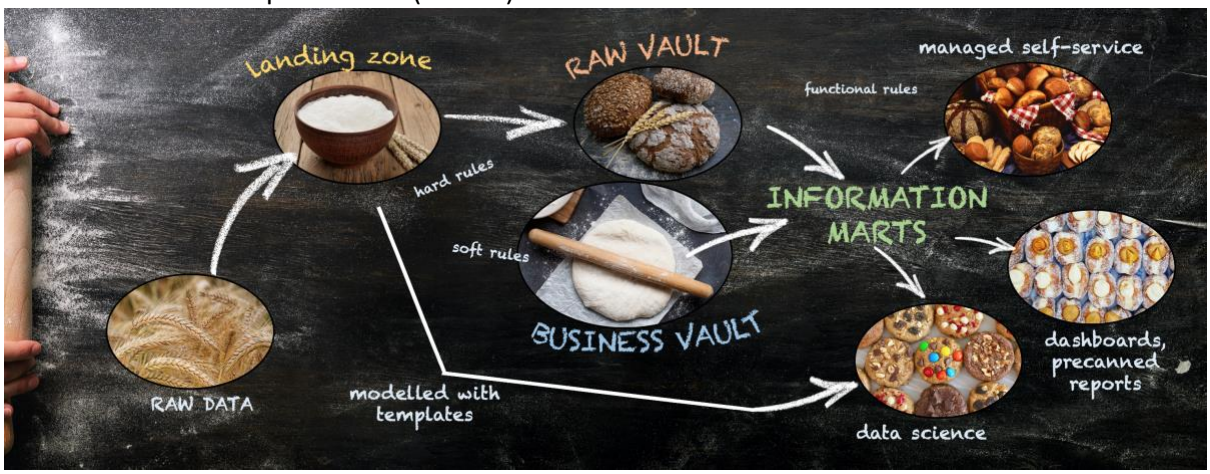
Now, skilled modellers and analysts who know how to query a data vault should be able to do it directly. But a data vault contains *many* tables and therefore many joins. So, a data vault is not the layer of content that should be exposed to those users that do not enjoy writing SQL! At the information mart level, you should also not expect to see system keys, and that includes **surrogate hash keys**! The beauty of building views over a data vault is that when the source is up to date, so are the marts! No lag times!
Use query assistance tables to manage the depth and frequency of the information marts; what are they?

1. **Point-in-Time (PIT) table** – a **disposable** table that is used to enhance the joins between a principal table (either hub or link) and its surrounding satellites by enabling EQUIJOINs. Using a PIT to support information marts means you forgo the need to use between clause in your information mart queries. Build multiple PITs to contain the keys at different required frequencies and depths.
2. **Bridge table** – a **disposable** table used to shorten the distance between a hub at one end of the data vault model and a hub table at other side of the data model. If it weren't there then the analyst would have to write the long SQL between hub, link, hub, link, hub link, hub… Like PITS build multiple bridges to contain the keys at different required frequencies and depths.

Both query assistance tables and information marts may contain **functional** derivations that subset the data, contain derivations suited to the **business intelligence (BI)** tools, reports and dashboards.

- See how to make a PIT, bit.ly/3iEkBJC
- And how PITs work, bit.ly/3vjTXdg

Minimal viable product (MVP)



What are the minimal ingredients that are mixed before the product is bread? You'll hear this in every agile project, "don't boil the ocean." This is true in building a data vault and every platform. You need to "fail fast" (break a few eggs), prove the end-to-end implementation to ***establish repeatable patterns***, what are they? What do we need to build a *steel thread (see: [bit.ly/3tUs77h](bit.ly/3tUs77h))?*

The approach

To begin with, what approach will be used to manage the data analytics platform? And they are applicable to how the enterprise is structured, at least the **business units** that are using it!

- **Platform as a Service** – centrally provision environments and allow business teams (skilled in data acquisition, data modelling and integration) to build and manage local BI solutions. *You have an empty kitchen with the baking tools.*
- **Data as a Service** – central analytics and engineering team acquires and provisions raw data. Business teams (data modelling and integration) performs integration, transformation, and analytics work. *You have been provided with flour, eggs, and yeast.*
- **Analytics as a Service** – central team manages all aspects of data acquisition, transformation, data modelling and analytic solutions. Local business entity consumes insights. *You have been provided with the baked goods based on your order!*

The what and the how the platform is managed can be a combination of the above, some business units will prefer pre-canned reports, others will want the ingredients provided to build their own insights (in a **data lab**) and business rules to supply new **curated** data.
Secondly, identify the data required for the use case and determine how the data will be **ingested** while at the same engage with the business architects (or nearest role) to identify the naming standards (especially for hub tables). Follow up with the mob modelling session and depending on the engagement model determine how the data will be provisioned for consumption.
Repeat for future use cases
The initial use case will undoubtedly take the longest, but with the steel thread established future iterations are using now established templates for delivery.

Quality ingredients make quality outcomes

Source data from disparate sources, what is there to guarantee that all sources will deliver an acceptable qualify of data? How will we ensure that they effectively integrate? Let's look at a few additional ingredients for the data platform,

- Built in **testing**, **documentation**, and data **lineage** – provided by tooling and recorded for auditability, does the raw data on the analytics platform reconcile with the source that provided it? Testing and documentation occurs at all levels, identifying critical data elements (CDEs) and data that should be treated differently due to its nature with regards to **privacy** and to meet **regulatory standards**. Topics here include **business glossaries**, **data catalogues** and **data dictionaries**, automated and built-in **data classification** that can rapidly accelerate future initiatives.
- Data quality metrics (aka *technical DQ*)
  - **Accuracy** and veracity that can be achieved by automated testing patterns
  - **Completeness** refers to missing attributes and rows, a count of columns and records received versus what was sent. A missing attribute might make the data you receive appear to contain duplicates.
  - **Conformity** is a hard rule implementation, data sent not in a consistent format as that what was agreed within the service level agreement (SLA)
  - **Consistency** refers to the content being sent that is not quality controlled, also within the SLA this is a measure of the *variance* of the data sent to you
  - **Timeliness** and **freshness** not only refer to data being sent *on time* (and could affect the measurement of the business process) but is the data sent in the *right sequence*?
  - **Uniqueness**, especially as it relates to the state of a business object and/or its relationship with other business objects that there is only one state of that parent entity!
- Business process quality metrics (aka business DQ),
  - **Coverage**, are there faults detected in the business process? If a customer gets a home loan but no address is supplied this could be a measure of the success of the source platform and/or applications that these are measured and repaired.

Data quality rules can be (derived) established with the outcomes stored in (you guessed it) business vault.

- DataOps, bit.ly/3vhEDxJ
- Automated testing, bit.ly/3dUHPIS
- Advanced automation, bit.ly/3y4mUdV

Indulge on repeatable patterns customised to your business

Adhering to **standards** (ingredients and steps) – a data vault that is a part of the larger enterprise vision will reflect its business architecture. Every business process has business objects in it, represented by *immutable* **business keys**, that object's relations and transformations are tracked through automation tools and applications while *shedding* (some critical) data elements to be historised and mapped to the data analytics platform. A key component of business architecture is **Information Mapping**, the principals of information mapping are

1. Information is a **strategic business asset**
2. Information improves **decision making** and innovation
3. Information is **owned by the business** and its **suppliers**, **partners** and **clients**
4. Information **integrity is essential** to business success
5. Information is a **foundation** for other business views
6. A common, **shared business vocabulary** streamlines collaboration, communication and automation
7. Business rules are **intrinsically associated** with business information
8. Information access is **restricted** by security, confidentiality and privacy polices
9. Information is based on **business objects**
10. Information is categorised into **types**
11. Information has **states**
12. Information has **relationships** to other information
13. Capabilities **modify** information
14. Capabilities use information to **deliver outcomes**

(BIZBOK Guide, bit.ly/3zofiDl)

Data Vault is the *embodiment* of *information mapping*

No Knead Bread - https://nyti.ms/3lB2p49

The views expressed in this article are that of my own, you should test implementation performance before committing to this implementation. The author provides no guarantees in this regard.