CDP Evaluation Guide: Data Ingestion Issues

Introduction

This paper is one of series intended to help buyers evaluate Customer Data Platform (CDP) systems. It provides a list of features related to a specific CDP function, with explanations of what each feature does, how implementations differ between systems, and how different approaches relate to user requirements. The paper is not intended to champion any particular approach to each feature, but rather to help buyers determine which features matter in their own situation and which approaches are best suited to their needs.

Descriptions in the paper are presented in non-technical and generic terms. Readers should recognize that vendors may use different terms to describe similar concepts and that accurately explaining a vendor's approach may require diving more deeply into technical details. Buyers should have someone with the necessary technical knowledge available to help understand these issues.

Buyers should also recognize that there can be different technical approaches to meeting a particular requirement. This means they should try to understand whether the approach taken by a particular vendor can work for them, rather than considering only vendors who follow an approach that the buyer feels is best.

How to Use This Paper

This paper provides a list of items buyers may wish to consider in evaluating a CDP. It is up to buyers to determine which items are important in their own situation and what their requirements are for each item. These requirements are based on how the CDP will be used in their business. Therefore, before using paper or other methods to evaluate individual vendors, buyers should develop an overview of how the CDP and other marketing systems will work with each other, which itself should be derived from the ultimate business goals those systems will support.

But buyers should also remember that business needs and marketing technology can both change rapidly. This means that features which currently seem unimportant may later turn out to matter. So buyers should explore those features with potential vendors, even though they do not weigh heavily in the purchase decision. It also means that flexibility is itself a requirement to consider along with ability to perform specific tasks. Balancing flexibility against other objectives is an important part of making a sound system selection.

In sum, this paper provides a framework for discussions with potential vendors along with ideas of what to listen for when covering each item. Buyers may not need every item in the list and will certainly give different weights to different items. But covering them all will help to ensure they fully understand the system they purchase before they sign a contract.

CDP Background

The CDP Institute defines a Customer Data Platform as a marketer-controlled system that maintains a unified, persistent customer database which is accessible to external systems. Key elements of this definition are:

- marketer-controlled system: the CDP can be purchased and operated by the marketing department with minimal assistance from the corporate IT department or an external vendor. Some technical skills will still be needed, especially during system set-up and when making changes in data sources. For marketing departments without the necessary internal resources, most vendors provide those services directly or through partners.

- unified, persistent customer database: the CDP collects data from multiple sources, stores at least some data permanently, and can associate data with customer identifiers to present a unified customer view. The system does not necessarily incorporate all sources, store all data internally, or combine all identifiers relating to the same individual. But it assembles enough data to provide a reasonably complete view of each customer and support reasonably consistent treatments across most channels.

- accessible to external systems: contents of the CDP are easily available for external systems to use in their own processes. There may be some limits on external access but the CDP is designed with access in mind. This distinguishes the CDP from "integrated suites" that are designed primarily to share data only with other suite components.

CDP Functions

Core functions of the CDP include:

- data ingestion: this is connecting to source systems and loading their data into the CDP. Major topics include the source systems, data types, and connection methods; how the data is stored within the CDP, performance of the CDP in terms of speed and scale, and system functions that support ingestion processes.

- internal processing: this is preparing the loaded data for use. Major topics include transformations such as data cleaning, standardization, and enhancement; linking of data that relates to the same entity to build a unified customer profile; and reformatting the data for access by CDP applications and external systems.

- data outputs: this is using the CDP data. Major topics include how external systems connect with the CDP and applications provided by CDP vendors.

Each of these functions is the subject of a separate paper in this series. Another paper covers commercial issues that are important in CDP selection.

Data Ingestion Checklist

Data Load

These questions relate to the sources of data loaded into the CDP.

- Internal systems. A company's own systems are the foundation of the CDP because they provide direct information about interactions with customers and prospects. Typical source systems include the company Web site, e-commerce, mobile apps, retail point of sale, sales automation, customer support, order processing, billing, and loyalty programs. There may be other types peculiar to your business, such as deposit accounts in banking or ticketing in airlines. Data from a company's systems is often referred to as "first party" data. Key questions include whether the CDP can incorporate the types of data provided by your company's source systems; whether there are existing connectors for your company's existing systems; whether there is a public Application Program Interface (API) to build new connectors; and, the effort involved in building such connectors.

- Tracking tags. Some CDPs gather data directly through Web site tags and related techniques. Most vendors with this feature originally started with Javascript tags that sent information about Web site visitors back to a central server. They later expanded to become "tag management systems" that incorporated other tags, reducing page load time, simplifying maintenance of multiple tags, and giving owners greater control over data flowing from their sites. The vendors also expanded their scope to capture data from other sources such as mobile apps, which requires a different technology from the original Javascript tags (see next section). Since tag management remains an important function in itself, key questions to ask include how CDP users add, configure, and remove other tags; the impact on page load times; and, how they control which data is shared with which tags. More general questions relate to the types of data the system can capture and how it is stored and accessed.

- Software development kits (SDKs). Some CDPs provide SDKs that can be embedded in mobile apps or similar systems (interactive TV systems, fitness devices, smart home devices, other Internet of Things products). SDKs gather data about user and app behaviors. Many go beyond this to deliver messages and take other actions within the app. Like tag management systems, an SDK from a CDP might control the flow of data from the app to other systems, saving app developers from embedding multiple SDKs in their product. Questions to ask include: what technical skills are needed to deploy and manage the SDK; what kinds of data can the SDK capture; what prebuilt connectors exist to feed data to other applications (replacing those applications' SDKs) and to other sources such as databases or reporting systems; what functions are available to create custom connectors; and, what capabilities does the system have to deliver messages and take other in-app actions.

- Web spiders. Some CDP vendors gather data from external Web sites, including pubic social media activity. This is most often used to develop company profiles for business marketing. The resulting data is used to enhance customer and prospect profiles with information a company cannot gather through its own systems. This sort of processing requires advanced techniques to interpret unstructured sources such as Web site contents and natural language comments. Key questions include what data sources are included; how those sources are selected; how the source data is

processed to extract information; what categories the data is assigned to; whether the user can create custom categories; and, what checks are performed to ensure accuracy. Again, there are other more general questions such as how information is associated with the right companies or individuals.

- External data. CDPs can incorporate data from other owners. These are usually companies whose business is acquiring and selling data, such as compilers of large personal or business databases or advertising networks that associate information about Web behavior with anonymous cookies. This sort of information is called third party data. Sometimes the information may come from another company that is interested in sharing information about its own customers or prospects with another company, perhaps for a joint promotion. This is called second party data. Processing for second party data may be handling by a trusted agent so that neither company needs to share its entire customer file with the other. Questions related to external data include whether there are existing integrations with specific commercial sources; the effort involved in setting up new connections; how often the sources are updated and the data is loaded; whether sources can be accessed in real time without importing the data; and, how any usage restrictions are managed. Second party data also often comes with specific usage restrictions that must be managed in the CDP.

- Load methods. Data can be loaded into CDPs in several ways. The preferred technique is usually an API connection, which requires almost no effort once it's set up. Questions about API connections include whether the API is published and documented; what functions are available (beyond loading data, you might to do things like define new data types or relations); whether multiple records can be loaded at once; what kind of security is in place; whether the CDP can request data from another system's API; and, whether there are any limits to the amount of data loaded or frequency of API calls. When a source system does not support API connections, loads may take place using batch files. The first questions is whether batch loads are supported; other questions include which file formats are accepted (Comma Separated Value, XML, database tables, etc.), whether loads can be automated and how this works; procedures to handle errors during the load process; and any limits on size or frequency. A third load option is for the CDP to actively query an external system. Questions would include the types of queries supported; how connections are configured; automated scheduling; error handling; and volume or frequency limitations.

- External access. Some CDPs can access data that is stored externally, either in the company's internal systems or in an external system. This approach is sometimes called "federated access". It avoids loading large volumes of detail, such as Web logs, or data that is only relevant during some situations, such as weather or location when a purchase is made. Questions related to external access include the types of connections available; what connectors exist for specific sources; the time needed to retrieve the data and make it available (very important for real time interactions); how the system finds data for a specific individual (e.g., is a customer ID needed?), how the CDP specifies which data elements are returned; and, how the system responds if the requested data is not found.

Data Structure

These questions relate to how data is stored within the CDP.

- Data types. Broadly speaking, your CDP needs to store whatever types of data your source systems will send it. At a minimum, this will include standard, structured elements such as customer name and transaction dates. These easily fit into conventional data structures which each item is carefully defined and stored in its own location. But most customer data today also includes less structured information, such as Web logs and message text. Instances of this sort of data may vary from each other, containing different sets of elements depending on exactly what is being reported. Sometimes those elements are labeled when they are provided, for example in a "key:value pair" that includes two elements: a key specifying the data type and a value with the actual data. Such a pair might be "FirstName: David". As the example suggests, additional context is often needed to indicate the owner of the pair – in this case, it might be associated with another pair showing customer ID or account number. Such pairs are common in "NoSQL" databases. They make it easy to add new data elements (i.e., keys) without formally defining them in advance. Of course, the keys must be named consistently to make the data usable.

  Other data types have even less structure, such as blocks of text which must be parsed using advanced language processing techniques that can extract specific elements, such as identifying the company name within a press release or the product name within a customer complaint email. More advanced processing can move beyond finding entity names to understanding relationships among entities (for example, that one news article says Company A bought Company B, while another article says Company C is suing Company D) or to understanding emotions (for example, that a customer is unhappy with a product). Typically this sort of processing is used to transform unstructured data into structured elements, which can then be processed using standard techniques.

  A CDP may also store non-textual data, such as images, videos, or audio. Often such information is accompanied by structured data, such as names, topics, and dates, which is used to access and analyze it. Advanced techniques such as image recognition may also be used to append structured data to these inputs.

  Questions related to data type include what types of data the system can ingest; what capabilities it has to extract information from unstructured or semi-structured inputs; and, how the stored data and descriptive attributes are accessed.

- Schemas. Traditional relational databases, such as Oracle or SQL Server, stored data in defined tables with defined data elements (columns) in each table and defined relationships (linked with a common element such as Customer ID) between the tables. This set of definitions provides a fixed scheme that makes it easy to understand exactly what is stored where. Such structures can be processed very efficiently for many purposes and are almost always present to somewhere in a CDP, if only because external systems need them to access the CDP data. But such schemas are inherently rigid, meaning that any change, such as adding a new element or table, must be defined in advance. CDPs that ingest less structured data may also store that data in less structured fashion, such as the "schema-less" key:value pairs mentioned previously. These can accommodate new data elements without advance planning, making them significantly more flexible.

  To make the data accessible, the CDP must still keep track of what elements have been loaded and often must convert these to defined data structures so they can be mapped and indexed for use by external tools. Questions related to CDP schemas include: how data elements are defined; the

process to add new elements; how users and other systems are informed what elements are available; and, whether there are any limits to the relationships among elements. The relationship question is important because CDPs are increasingly expected to store relationships such as connections within a social network and then to let users query against them – for example, to find "friends of friends" or "products owned by friends". These can be difficult queries for a conventional relational database. At the same time, schema-less systems may have problems capturing relationships that are precisely specified in a relational database schema. So marketers need to consider a variety of situations and explore specifically how the CDP would handle each of them.

- Standard objects. By definition, CDP data is organized around the customer. Some CDPs treat all inputs as attributes of the customer. This makes loading data easy but may limit the ability to store relationships among items – such as, linking products to campaigns. Other CDPs provide data structures with standard objects such as products, channels, marketing campaigns, and messages. Standard objects let the CDP include pre-built features that use those objects, such as campaign reports and next-product-purchased predictive models. Standard objects can also simplify mapping of external data into the CDP and access to CDP data by external systems. Questions related to standard objects include: what standard objects are built into the CDP; how are the objects related to each other; what is required to add an object; are there limits on how new objects can be related to standard objects; what happens if standard objects are not used; and, are there specific functions that rely on the standard objects being populated?

- Input mapping. Although some technologies allow ingestion of data without assigning it to predefined data elements, that data must ultimately be classified to be useful. This means users must specify a set of standard elements, such as name or email address, and then define which information from each source will be assigned to those elements. This mapping process is essential for combining information from different sources and for doing many kinds of analysis and processing. Questions related to the mapping process include: how are the standard elements defined; how are inputs from source systems assigned to standard elements; how are missing elements handled (especially key elements such as customer identifiers); how are new or unmapped elements handled; and, how does the system transform or standardize elements before placing them in standard fields.

- Access restrictions. Allowable use of some data in a CDP may be limited by government regulations, company policy, or agreements with the provider. The CDP needs to be able to enforce such limits or at least capture data to make enforcement possible. Questions related to access restrictions include: is there a standard method for defining access rights to specified data elements; can access be limited by time (such as, expiration date of a contract); can the system require specified credentials to access specified elements (such as, a special password); can the system limit access to other specified systems; can the system keep a log of access to specified elements; can the system alert managers to unauthorized access attempts; can the system add attributes to a data element that describe permitted uses; and, can the system anonymize data by removing specified identifiers before extraction.

Performance

These questions relate to the performance of the CDP.

- Latency. This refers to how much time it takes for new data to become available within the CDP. One factor is how quickly data is acquired from source systems, something which may range from instantly (as soon as it is entered into the source system) to periodically (a daily, weekly or even less frequent load of all new data since the last load). Another factor is the time needed to make the new data available: the system may need to extract structured data from unstructured sources, to transform data to standard formats, to check data for accuracy or completeness, to create aggregations or indexes, or to load the data into specialized databases that are optimized for external access. Questions related to latency include: ability to accept real time inputs; any limits on frequency of batch inputs; processes needed to prepare the data, and time for those processes; time to load data into any secondary structures; and what is accessible while update processes are taking place. That final question can be important if update processes are lengthy: if proper provisions are not made, the system may be wholly unavailable, run slowly, or return inconsistent information. Bear in mind that the frequency of inputs is often determined by the source systems, so some latency may be beyond the control of the CDP system.

- Response time. This refers to how quickly the system can return data when it is requested. Response time is most pressing if the CDP will support real-time interactions, such as Web site personalization, bidding on display ads, or ecommerce product recommendations. Such interactions can have very strict response requirements, as low as 30 milliseconds for some applications. Response time can also matter for non-real time processes, such as how long it takes to count records in a user-specified segment or to extract data on a segment. The computing resources assigned to a CDP can often be adjusted to meet specified performance requirements, or other features such as indexes can be introduced. Users need to know their required response times in advance. Questions related to response time include: time to return real time requests; activities that can be completed during a real time request (such as calculating a predictive model score); any limits on the volume or type of data returned in real time; any pre-specification required of which elements are available in real time; time to perform analytical calculations, and what factors impact the time; time to extract data using different methods (API, queries, batch files); and, options available to improve response time if necessary.

- Scalability. This refers to the volume of data the CDP can handle. It has numerous dimensions, including the number of source systems, number of customers, number of data elements per customer, complexity of the data model, and total amount of data stored. Latency and response time are often impacted by data volume, making them part of the scalability equation. For real-time applications, scalability also includes the number of simultanesous connections (to Web sessions, call center agents, mobile apps, etc.) the system can maintain while providing the required response time. Questions related to scalability include: any limits on the various dimensions (source systems, customers, data elements, data model, total volume, connections, etc.); configuration options to overcome scalability limits; and, scale of existing vendor configurations.

Functions

These questions relate to specific capabilities needed to support data ingestion.

- Deployment effort. This refers to the staff time and skills needed to deploy the system. The particular focus is demands on marketing staff, both technical and non-technical. A CDP that

requires too much work will simply never be deployed. Questions related to deployment effort include: the tasks performed by marketing staff, including the work hours, skills, and specific business knowledge needed; tasks performed by vendor or other external staff; tasks performed by corporate IT staff; training required; scope of the initial deployment; project timeline; assumptions built into the timeline; and, project management process.

- Maintenance effort. This refers to the staff time and skills needed to maintain the system after deployment. It includes both the effort to operate the system on a day-to-day basis and to make changes such as adding new data, adjusting data preparation processes, exposing new outputs, and connecting new systems to use the CDP. Although these cannot be known precisely in advance, marketers need a realistic estimate of the resources they will need to commit to the CDP. Questions related to maintenance include: the tasks required to maintain the system; work hours, calendar time, skill, and training needed for specific tasks such as adding a new data source; tools available to help with these tasks; and, options to use the vendor or outside resources for those tasks.

- Check inputs. This refers to basic checks on input data such as whether it includes expected data structures, elements, and values. Finding bad inputs when they are loaded is important because a small set of bad data is easily overlooked once it merges into the much larger pool of the entire CDP. Input checking is most critical for highly structured data, since the system may have no way to process records in unexpected formats. But even systems that can accept unstructured inputs may need to flag new elements or labels so users can decide how to classify them. Some systems can also check that the distribution of data values in particular field is reasonable: for example, a data feed that had all customer birthdays set to the same day or all transaction amounts set to zero would be highly suspect. Questions related to input checks include: what kinds of checks are possible; what reports are provided; how does the system identify questionable inputs; and, how does the system alert users to potential problems.

- Roll back bad input. This refers to an ability to remove bad data after it slips into the system. It requires that the data be tagged with its original source and that the system either physically delete the bad information or flag it to be ignored. Systems that overwrite existing data instead of appending new records can be much harder to correct because the previous value may be lost. Removing bad data may also require recalculating derived data such as aggregates, extracts, and indexes. This can be very time-consuming when large volumes are involved. Questions relating to roll backs include: whether roll back capabilities exist; which types of data can be rolled back; configuration choices needed to make roll back possible; steps and technical skills needed to execute a roll back; time required to execute roll back; and, system availability during a roll back.

- Discovery and exploration. This refers to features that let users explore new data before it is added to the system. (Exploration of data after it is loaded is covered in a different checklist.) It extends beyond automated input checks for input quality to let users find new data elements, values, and relationships. Core capabilities include viewing samples of input records; frequency analysis of data element labels and values; correlation reports between labels and values; and comparisons of new inputs against previous inputs. The specific objective is to help users determine whether and how to use new data sources. Questions to ask include: what tools are available to examine input data; what kinds of standard reports are available; how much of the load process must be completed before the data can be explored; what skills are needed to use the tools; can data be loaded into a staging area before it is merged with the primary data store; what kind of data store is used for the

staging area; what third party tools can be used to examine data in the staging area; and, can data in the staging area be compared with data already loaded into the primary data store.

About the CDP Institute

The Customer Data Platform Institute educates marketers and marketing technologists about customer data management. The mission of the Institute is to provide vendor-neutral information about issues, methods, and technologies for creating unified, persistent customer databases. Activities include publishing of educational materials, news about industry developments, creation of best practice guides and benchmarks, a directory of industry vendors, and consulting on related issues.

The Institute is focused on Customer Data Platforms, defined as "a marketer-controlled system that maintains a unified, persistent customer database which is accessible to external systems."

The Institute is managed by Raab Associates Inc., a consultancy specializing in marketing technology and analysis. Raab Associates defined Customer Data Platforms as a category by Raab Associates in 2013. Funding is provided by a consortium of CDP vendors.

For more information, visit www.cdpinstitute.org.

