THE 3 Cs OF DATA QUALITY

Data quality is often described in terms of its effects on your data:

It "makes your data trusted." It "turns your data into an asset."

Or it's defined in terms of the activities you perform:

It's the result of profiling and transformation and standardization and filters and cleansing and...

But you can do all of those activities and not be able to describe, exactly, how your data has achieved that sought-after "trusted" or "asset" status.

In this ebook, we're introducing a way to define the goal of data quality.

But we're not going to use vague aspirations or general goals. Instead, we're going to describe tangible properties of the data that, considered together, indicate whether your data is high-quality.

MEET THE 3Cs

Every dimension of high-quality data can be expressed as one of the Cs.

WHY THE 3Cs?

Even though there are a lot of factors that go into data quality, we created a definition with just 3 legs.

Why?

Because data quality *is* complicated, in the understanding and in the execution. And it's also important. We truly believe that, and if you found your way to this ebook, you probably do too.

But there are lots of people out there who don't believe that data quality is important, or have never given the idea of data quality a single thought.

WHY THE 3Cs?

Some of those people work at your company. And before you can get data quality done, you have to explain what data quality is (and why it's worth their time).

This is where the 3 Cs come in.

At the highest level, they're easy to remember and catchy to explain—key for grabbing the attention of busy executives.

But when you need a deeper dive to satisfy a skeptical gatekeeper, you can count on the 3 Cs then, too. You can look at all nuances of data quality through the lens of the 3 Cs.

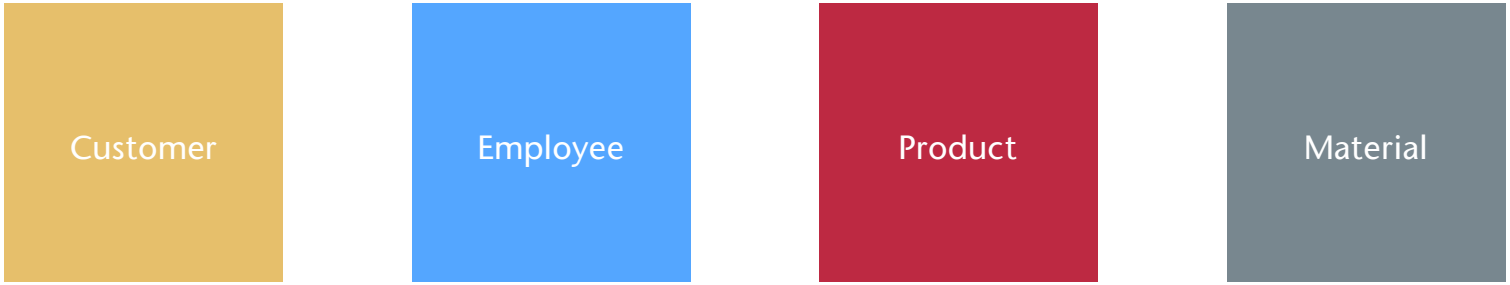
THE FUNDAMENTALS

Here are the fundamentals of the 3 Cs. Everyone should start with these fundamentals, and many people won't need more of a definition than this.

1 COMPLETENESS

KEY IDEA

Your data is describing *something*—people or places or things or some combination of those.



Complete data describes those objects with the level of detail that you need in order to achieve your intended purpose, and is also available to you.

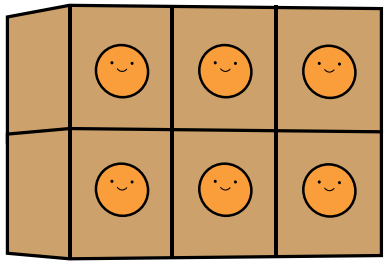
2 CORRECTNESS

KEY IDEA

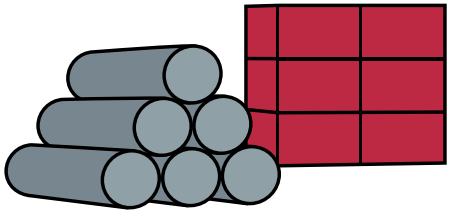
The *something* your data is describing is a real-life something.

Customers

Employees



Products



Materials

Correct data reflects the actual properties of the object with the level of accuracy that you need in order to achieve your intended purpose.

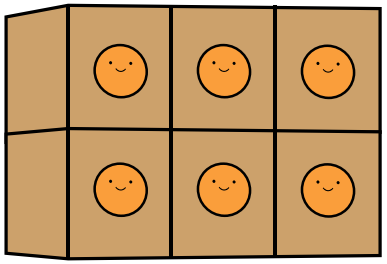
3 CLARITY

KEY IDEA

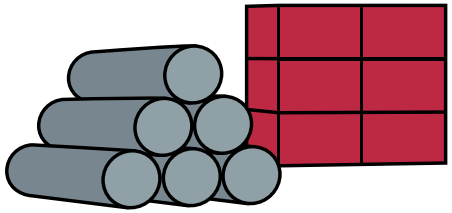
The *something* your data is describing has more than one aspect, and it has connections to other objects.

Customers

Employees



Products



Materials

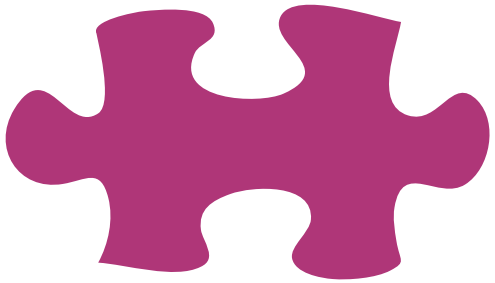
Clear data makes it possible for you to understand the different aspects of the object and how they relate to each other, as well as how the object relates to other objects.

MIOsoft

Imagine that your data is pieces of a puzzle.

In the data quality world, this is a pretty common image.

And imagine that to get value out of your data, you need to assemble the puzzle.



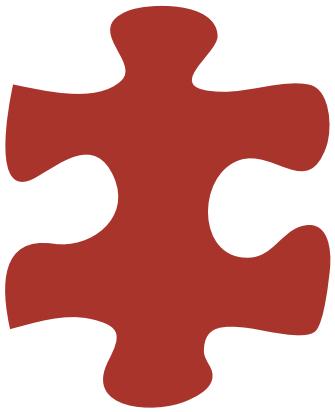
Completeness is the equivalent of having all of the puzzle's pieces.

Correctness is the equivalent of having all of your pieces come from the same puzzle.

Clarity is the equivalent of having the puzzle assembled.

In many cases, this definition—the fundamentals of what data quality are and why they're relevant to your business—will be all you need.

But if you or your stakeholder want a more specific breakdown, read on. In the following chapter, we'll take a deep dive into the details of what makes up each of the 3 Cs.



THE DEEP DIVE

If you like detail, this is the chapter for you.

A definition of data quality that shows everyone the broad strokes is helpful in many ways.

But when it's time to dive down into the details of what solution you want and how you're going to implement it, everyone needs to be on the same page about what each of those strokes encompasses, exactly.

This is also the chapter you can refer to if a project stakeholder starts asking questions like: *"How do you plan to demonstrate results?"*

Measuring these factors is one way of doing that.

COMPLETENESS

Completeness is about the **presence** of the data.

EXISTENCE

When data exists, it's somewhere in your systems. Pretty obvious.

And the data that doesn't exist? Maybe it was never collected, or maybe it was deleted with no backup. For data quality purposes, it doesn't really matter.

Part of data quality will involve identifying existence gaps in your data. Sometimes, you may discover that those gaps need to be filled in order to achieve the desired results for your project.

Whether you collect that data yourself, get it from a third party, or both, don't forget to conduct data quality on this gap-filling data. Recency of acquisition is, by itself, not a reliable indicator for high quality.

ACCESSIBILITY

When data isn't accessible, you know that it exists—but you can't get to it.

It doesn't matter why. The barrier could be regulations, policies, technology, resources, or anything else. But if you can't access relevant data, your data isn't complete.

Some accessibility barriers are more surmountable than others, of course.

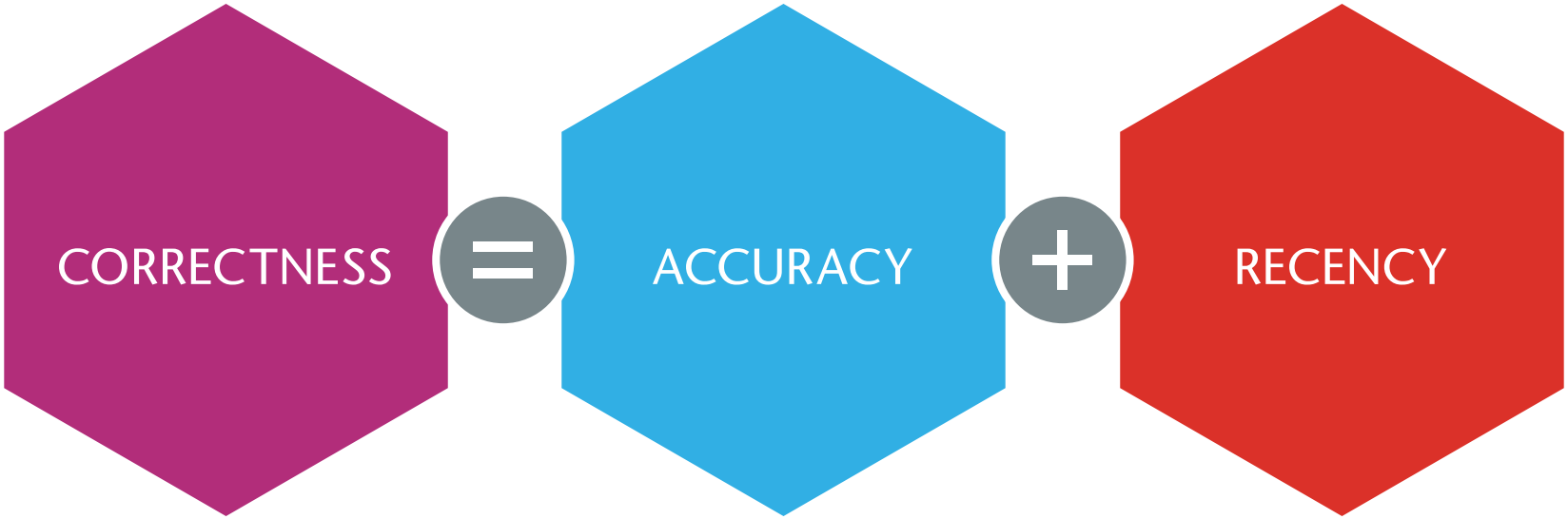
In some cases, choosing the right data quality might, itself, be able to help: if you're having resource or technology barriers, or if you need to meet a technical regulatory or political standard like data masking.

But if you're facing a non-technological barrier, like laws around where you can access from, or policies about who can access the data, you might simply have to count that inaccessible data as a negative in your overall data quality assessment.

If you know that inaccessible data is out there, don't just forget about its existence. The output of your data quality project might make its way to people who do have access to that data. Those people should be able to weight your results against their knowledge of that data. (And hopefully, *they're* doing data quality on the sources you couldn't get to.)

CORRECTNESS

Correctness is about the **factual appropriateness** of the data.



MIOsoft

RECENCY

Data can only be correct if it's recent enough that you have confidence in its accuracy.

Recency and accuracy are two sides of the same coin, but it's worth making the distinction between them.

For lots of data, *accumulation* is a type of change. So while a person only has one accurate birthdate, they can easily have multiple accurate shipping addresses: home, work, and school, for instance. For a shipping address, recency matters when you're trying to figure out whether a given address should be considered "good" or not.

And the degree of recency that a particular project requires in order to be accurate can vary significantly for the same piece of data.

We divided the concept of "recency" into three rough categories, for the purposes of discussion: stable data, steady data, and volatile data. These categories are about how long data, once recorded accurately, is likely to remain accurate.

When we talk about these categories, we assume that once data is made accurate, it stays that way—inaccuracies aren't (re-)introduced. This isn't always the case, of course, but it affects data of different recency categories in pretty much the same ways, so we're not going to discuss it further here.

- Stable data maintains its accuracy over time, full stop. If the data was accurate, then it's still accurate now. The date a building was built, or an item was manufactured, or a person was born are all examples of data that is stable regarding recency.

- Steady data maintains its accuracy over an extended period of time. Accurate steady data is likely—but not guaranteed—to be accurate now, as long as it's been gathered in the last few months or years. Phone numbers, addresses, and spousal relationships are all examples of data that's steady regarding recency.

- Volatile data maintains its accuracy for only a short period of time. You can't count on volatile data to stay accurate for very long—months at the most, minutes or seconds at the other end of the span. The time of last login, a credit card balance, or a cell phone location ping are all examples of data that's volatile regarding recency.

The degree of recency that different pieces of data require can vary significantly, of course. But so can the degree of recency that different projects require, even if the data is the same.

This is because *recency* tends to become more important to data quality as the time-sensitivity of a project increases and as its scope narrows. That's scope not in terms of the number of entities involved in the project, but in terms of how the project's data is used.

For example, consider data that's always used in aggregation, for analytics, versus data that goes into front-line systems and guides interactions with high-value customers. For the latter, up-to-the-second recency, even for volatile data, is important. For the former, depending on the type of analytics you're performing, you can potentially widen the scope of what is "recent enough"—and therefore "correct enough"—for the project.

ACCURACY

Correct data is factually accurate.

Ensuring accuracy is a huge part of data quality, because all the analytics in the world can't deliver benefit if the data they're given is just flat-out wrong.

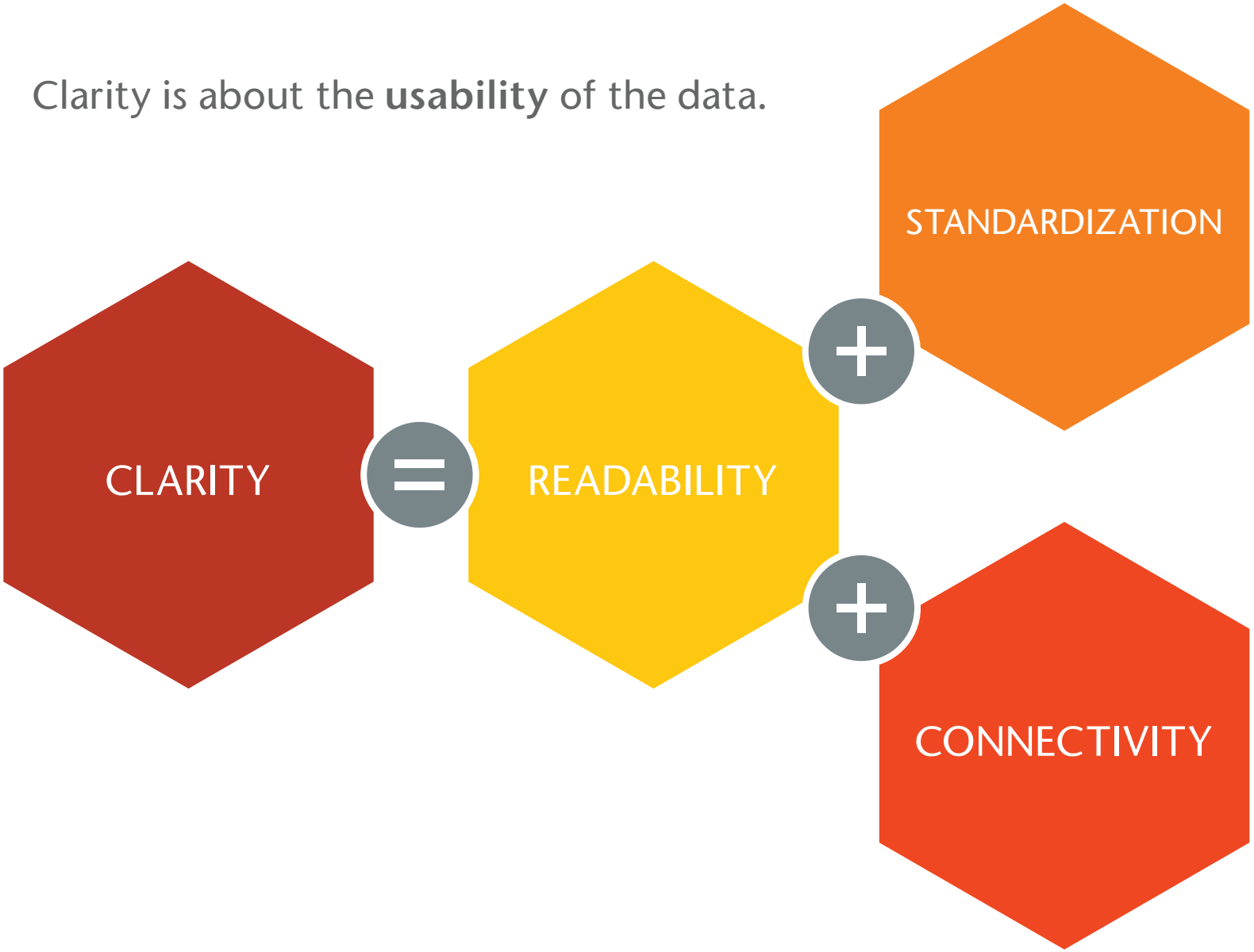
Part of ensuring accuracy is simple validation, which is essentially checking whether the data's format and value even allow the possibility of the data being factually accurate.

The other, trickier part of accuracy is determining whether the data is a true reflection of the real world. If you're dealing with data that ever changes (and you probably are), then achieving 100% accuracy is unlikely.

Accuracy often remains important even when recency is less so.

CLARITY

Clarity is about the **usability** of the data.



READABILITY

Readable data is comprehensible, specifically comprehensible to the intended readers. And a data reader could be a human, a machine, or both. Most likely, it's *multiples* of both.

Almost no data is only relevant to exactly one thing. Most data will be put to use for different purposes, and it'll often have to be consumed differently for those purposes.

That means high-quality data has to be ready to be presented in multiple forms, which might not have very much in common with each other—or with the way the data is being stored.

To build this variety of readable structures, high-quality data has to be very clearly defined and prepared.

STANDARDIZATION

The basic concept of standardization is straightforward: when data is standardized, data of the same type is directly comparable, without having to do a lot of cleansing or transformations on whatever form the data is being stored in.

At its most fundamental level, you have to conceptualize type as, of course, data type: integer, string, boolean, date, etc.

To be standardized at this level, all dates must be comparable with other dates, strings with other strings, and so on.

But you can't just stop there. You have to consider type at a higher level of abstraction, in terms of types of entities.

At this level, you need standardization like customers that are comparable with other customers, and products that are comparable with other products.

This doesn't mean that every entity of a type needs to have an absolutely identical structure, of course; a clothing product vs a food product, or an individual customer vs a business customer might (and probably will) require different models.

But it does mean that comparable entities should be structured in the same way to the extent possible. Personal names, for instance, should be fundamentally treated the same way for any entity that has a personal name. Individual customers should be comparable to individual employees, because both are people.

To achieve this, there needs to be a certain level of unity in your data quality initiative overall, so that different projects and different domains approach comparable entities in a standardized way.

This is where data governance—which often goes hand-in-hand with data quality—can play an important role. A consistent data dictionary can play a major role in standardization; so can associated meta-dictionary information like how standardization tools should be configured for different types of data.

Using a central business model to drive a data quality project, instead of attempting to work off the sources as-is, also contributes significantly to standardization. Different projects can use different combinations of sources, but if they're all referencing the same business view of the world, then it becomes much easier to standardize cross-project data.

CONNECTIVITY

Data connectivity is the understanding and representation of the meaningful relationships in your data.

Relationships between what? Between people, places, and things—the real-world entities that your data represents.

That includes both the relationships *within* entities, and the relationships *between* entities.

"Understanding the relationships within entities" is basically the idea of entity resolution: if you have multiple records about an entity, you should be able to recognize that fact, and understand the relationships between the data in the records well enough to synthesize those records into a single representation of the entity (or at least to generate a single-record representation of the multiple records).

Understanding relationships between entities is equally simple conceptually, but harder to execute in practice.

Data quality should not only establish explicit relationships (where, for example, the name on a policy indicates a relationship between that policy and the person by that name), but implicit relationships.

Implicit relationships are relationships that the data suggests, but does not outright state: for example, if two people with the same name live at the same address at the same time, they're likely to be family members, and they are definitely members of the same household.

These types of relationships can be detected even when each person's record doesn't mention the other, simply through the meaning of the data in the records. Data quality needs to discover these relationships in order to achieve maximum connectivity—and therefore clarity.

THE TARGET

To get value out of your data for a particular project, the 3 Cs need to apply to all of the data that's relevant and available.

We can't give you an exact outline of what "relevant" and "available" mean. That's going to vary from company to company and even project to project.

Availability is going to be affected by factors like:

- Company security policies concerning data access
- Regulatory rules concerning data access
- Operational needs of the systems
- Budget/effort/time available to create access

While **relevance** is often more affected by factors like:

- The subject of the project
- The scope of the project
- The confidence level required for the results

As you expand data quality within your company, your definitions of "relevant" and "available" are likely to change, as you discover data and connections (or a lack thereof) that you didn't know about before.

MIOsoft

THE CHECKLIST

Ready to start working toward the 3Cs? Here's a checklist with some key questions to answer as you begin your data quality project.

- [ ] What data do you need for this project?

- [ ] Which systems contain the data you need?

- [ ] Can you get data out of all of the above systems?

This includes direct access to the system, but you could also read from a backup, from an Enterprise Service Bus, etc. to get the system's data without accessing the system itself.

- [ ] If not, is there any way to start getting data from those systems?

This includes direct access to the system, but you could also read from a backup, from an Enterprise Service Bus, etc., to get the system's data without accessing the system itself.

- [ ] Does the data that you need actually exist in the expected systems?
- [ ] How are you going to check for factual accuracy of the data?
- [ ] How much of the data you are working with is stable vs steady vs volatile?
- [ ] How recent does each of these types of data need to be for your purposes?
- [ ] How will you determine a data's recency?

- [ ] What is the type of each piece of data that you have in your project?
- [ ] Which pieces of data need to be directly comparable?
- [ ] Who (or what) needs to read the data that results from this project?
- [ ] What requirements for the data do the readers have?
- [ ] How are entities related to each other?

RELATED READING

Proactive Data Quality: A Definition and Review of Application Points

Data Quality Vendor Evaluation Checklist

The Trailblazer Approach to Data Quality: An Introduction & Definition

MIOsoft®

READY TO HELP

If you're ready to start seeing the 3 Cs in your data, be sure to check out MIOsoft. Our data quality software is designed to help you bring completeness, correctness, and clarity to your data by combining powerful data preparation and cutting edge entity resolution with a proactive, success-focused approach.

Are you working with a lot of data? No worries: MIOvantage is designed to scale, so no matter how much data you need to improve, MIOvantage has the capabilities you need to stay current.

STANDING BY

Interested in working with MIOsoft for your data quality?

Get in touch!

sales@miosoft.com