PLATFORM MODERNISATION BRIEF

MODERNISING A DATA PLATFORM WHILE THE BUSINESS KEEPS WORKING

Published on 10 November 2026 · Version 1.0.0 · Updated 10 November 2026
Online article:
https://datanalysis.re/ressources/data-engineering/modernising-data-platform-while-business-runs

A new platform can be technically ready while the applications and teams that depend on its data are not ready to change. The delivery date of the new environment and the cutover date of the uses are two different decisions.

To prepare the second one, you need to look at what produces the data, what transforms it and what uses it. The method below proposes five decisions to document on a first scope. It prescribes neither a complete migration nor a particular tool.

1. Start from the uses that must continue

A dataset can feed an application, an invoicing process, a partner export and reporting. These consumers do not necessarily have the same constraints on timing, format or availability.

The map must therefore connect applications, flows, rules and owners. It must also flag uses that are still unknown. An export reworked by hand or a poorly documented legacy process may not appear in the first technical views.

AWS guidance on preparing a cutover stresses dependencies, tests, responsibilities and the transition runbook. [1] For our scoping, the expected result is a usable list: which use must be preserved, who signs it off and what evidence will show that it works after the change?

2. Choose a coherent scope

The first step is not necessarily the smallest table or the least visible application. It must form a set that the team can migrate and verify without leaving essential dependencies aside.

A progressive approach can limit the scope of each cutover, but it adds a period of coexistence. You then have to maintain the interfaces, understand which system is authoritative and manage any differences between the two paths. A grouped cutover may be necessary when components are tightly coupled. The choice depends on the architecture and the constraints of the business. [2]

The useful decision is therefore not “migrate everything or do nothing”, but “which uses can move together, with which checks and which stop conditions?”

3. Compare on the basis of identified rules and data

Comparing two totals is not enough if the periods, rules or starting data differ. Before acceptance testing, the team must set the time reference, filters, units and special cases that define the expected result.

A gap can come from the new process, from an update to the source or from a business rule that was changed during the project. Without an explicit reference, the team risks treating all these changes as a single anomaly.

Checks must cover the results, but also availability times, access rights and the behaviour of consuming applications. Data that is correct in the platform can still be read in the wrong unit or arrive too late for a use.

4. Assign the cutover decision

The go-ahead should not boil down to “the jobs are green”. You need to know which technical and business sign-offs are expected, which have been obtained and which reservations remain open.

Cutover decisions and what to document

Decision	What to document
Scope	Applications, flows, uses and exclusions.
Reference	Datasets, periods and rules used for the comparison.
Validation	Expected result and owner of each check.
Go-ahead	Person authorised to decide, criteria and acceptable reservations.
Monitoring	Immediate post-cutover checks, observation period and alert recipients.
Stop or rollback	Triggers, owner and handling of data already produced.

The runbook must specify the order of operations and the points where the change can be interrupted. A rehearsal in a suitable environment lets you test the plan against real dependencies; it does not turn the remaining assumptions into guarantees.

5. Prepare rollback together with the new writes

Going back to the old code is not always enough. Operations may have been recorded after the cutover. You have to decide how they will be kept, reconciled or replayed, without loss or duplicates.

The strategy must also cover outputs already distributed to applications and partners. A correction in the platform does not automatically withdraw a file that was sent or a decision taken on the basis of an incorrect result.

Cutover guides stress this need to plan rollback and the handling of modified data. [1][2] The project must identify the point beyond which a simple rollback is no longer appropriate and a recovery or correction procedure becomes necessary.

Turn the map into a first work step

The accompanying migration map is used to prepare a meeting between application owners, the data team, operations and the business. It brings together the scope, the expected evidence, the sign-offs and the rollback conditions.

It may reveal that a problem presented as a migration is first of all about a definition, a flow or a poorly controlled dependency. Dealing with that point can become the first step of the project, without presuming the complete trajectory.

Let's review a modernisation scope with DATANALYSIS. Bring the applications concerned, the uses to preserve and the change you are considering. We can scope the dependencies, the sign-offs and a coherent first step.

→ Review my modernisation scope: https://datanalysis.re/contact?lang=en

→ Fillable map (PDF): https://datanalysis.re/downloads/data-modernisation-carte.en.pdf

→ HTML map: https://datanalysis.re/ressources/data-engineering/modernising-data-platform-while-business-runs/migration-map

In the same series

When data changes meaning: preparing changes across applications:
https://datanalysis.re/ressources/data-engineering/data-changes-meaning-across-applications

Resuming a data flow without multiplying its effects:
https://datanalysis.re/ressources/data-engineering/resuming-data-flow-without-duplicate-effects

Related expertise

Data Engineering (available in French): https://datanalysis.re/expertises/data-engineering?lang=fr

Data Governance (available in French): https://datanalysis.re/expertises/data-governance?lang=fr

Sources

References consulted on 10 November 2026. The map and wording are a teaching adaptation of preparation and control principles. They do not constitute an executable migration plan without a review of your environment.

[1] AWS Prescriptive Guidance | Migration cutover best practices: pre-cutover stage (cutover preparation: dependencies, tests, responsibilities, transition runbook and rollback)
https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/pre-cutover-stage.html

[2] AWS Prescriptive Guidance | Migration cutover best practices: cutover stage (cutover execution, rollback and handling of modified data)
https://docs.aws.amazon.com/prescriptive-guidance/latest/best-practices-migration-cutover/cutover-stage.html

Fictional teaching example; no client results.