

The four core functions showing you need a PLM

Part data management (PDM)

A large variety of business objects

1. Versioning of business objects

2. Change process management

3. Collaboration

4. Applicability management

PDM and detailed-design centric PLM

Extending the scope of PLMs to several business domains

Spreadsheet replacement risks

The difficulty of replacing spreadsheet-based processes

Notes

See also

# The four core functions showing you need a PLM



Photo by [Nick Fewings](#) on [Unsplash](#)

This article is the third one of a series about PLM, after [PLM and Graph Data](#) and [Configuration management of industrial products in PDM/PLM](#).

This article begins with *spreadsheets*, numerous, ubiquitous spreadsheets that seem to rule many manufacturing processes, all along the product lifecycle, from engineering to manufacturing and support.

Contrary to the tertiary sector where many areas that were using spreadsheets got out of them to use software applications (specific development or off-the-shelf software), it seems difficult, in manufacturing, to get out of spreadsheets. In this article, we will expose some of the reasons why spreadsheets are so hard to get rid of.

## Part data management (PDM)

In our last two articles on PLM, we explained why the manufacturing area had a special vision of data, special comparing to the one of tertiary sector. In manufacturing, most data need to be managed as *referential data* even before to attach business rules to them.

That is why, some decades ago, the industry started to focus on "part data management" (PDM) systems, which are managing the part data lifecycles. Once this is taken in charge, it is possible to think about the product and its lifecycle, and enter the Product Lifecycle Management world (PLM).

The specific industry needs around data management make it complicated to replace many spreadsheet-based processes because the underlying requirements are more complex than they appear to be in the first place.

## A large variety of business objects

From now, we will suppose that we are taking about *business objects* and about their lifecycle.

The most common industrial business object is the *part* but it can be many other objects, depending on the business domain, for instance:

- **System engineering and early maintenance engineering:** Requirements, operational conditions, probability of failure, system, function, connector, trade-off, logical block, etc.
- **Detailed design:** Part, assembly, drawing, 3D, document, standard part, tolerances, change, applicability, compatibility, build, etc.
- **Simulation:** Test scenario, test campaign, business object composing the input data or the output data, configurations, etc.
- **Assembly preparation:** Routing, instruction, part, elementary action, sequence, balancing information, resource, tool, duration, etc.
- **Assembly:** Work order, part, resource, supplier, stock, quantity, tool, station, skills, clocking, warehouse, etc.
- **Maintenance:** Maintenance plan, documentation, data module, spare part list, work order, task, frequency, threshold, cycle, tool, skills, duration, etc.
- Etc.

All those objects generally need, as a basis, four core functions to be able to be manipulated correctly and safely within a manufacturing company.

## 1. Versioning of business objects

The versioning function is often done manually by renaming spreadsheets with a version number. But the real need that we can see is a versioning of the business object instances that are contained in the spreadsheet.

Versioning of data is the first fundamental function that many spreadsheet users need. Because the business objects that they manipulate evolve in versions, they have to keep track of those versions, even if doing that in a spreadsheet is often a very manual process.

The version management looks like the one that is done in the software industry especially because it is often relevant that several versions of the same business object are valid and used, *at the same time*, throughout the company or even outside.

This is totally linked to the fundamentals of the manufacturing process:

- The engineering division of the company is designing a product in version  $|N|$ ,
- The manufacturing part of the company is generally producing the version  $|N-1|$  of the product,
- The support organization is supporting all sold versions starting from  $|N-1|$ ,  $|N-2|$ ,  $|N-3|$ , etc.
- The customer of the product will have its product data in a certain version supported by the support organization.

So, the general fundamental need on data is to manage several "living versions" of data.

## 2. Change process management

When a business object version changes, it is very recommended (and sometimes mandatory due to regulation, such as in aerospace) to track several elements:

- Why was the change done?
- By whom?
- Who validated the change?

The first question is becoming more and more important in the industry. Because, if the "why" is not captured, future modification to the product may violate some hypothesis done during the design phase. When a product evolves, it can be very useful to track back the reasons of certain modifications, especially the cases that were explored and that were not chosen (see [note 1](#)).

The change process can be done at the business object level but also at a lower level. Indeed, potentially each field of a business object has at least one owner which has a role in the organization. This owner is able to modify the value of the particular field of the particular business object. In a good data governance environment, the various roles are clearly defined. It is important to track who changed what, especially when the business object is complex.

The third part of the change process is the process of validation of the change: If there is a change, one or several roles in the organizations are responsible to validate that the change, as it is, can be done. This must also be tracked because some responsibilities are involved, especially to validate that, in a complex environment, the change impacts have correctly been estimated by the various functional disciplines that are concerned.

In most evolving spreadsheets, a kind of manual change process is put in place to secure the main reasons of changes and possible the decisions taken by the "approvers". This part is often badly done in a spreadsheet environment because it is hard to track if the data manipulated are big and/or complex and/or under the responsibility of several actors.

## 3. Collaboration

As we saw, a business object can be shared by several roles, at different moments of a complex process.

Even if only one people is able to modify the spreadsheet at a certain moment, this person will be bound to meet with many people to gather all the information, or to exchange many mails to consolidate information coming from many actors.

This Excel-based collaboration process is not generally "hidden", in the sense that collaboration is hidden by the "people+spreadsheet" paradigm. As long as one person is maintaining the spreadsheet and is responsible for "gathering information", the collaboration feature that results from the data gathering process is often not visible.

Indeed, the real need is generally to have the involved roles update directly the data in an appropriate place, while pumping versions and explaining the reasons for change.

Even if the word "collaboration" is quite vague in terms of functionality, it can cover, in the industry, several complex realities:

- The capability of having many actors working on the same data;
- The capability of having many actors working on several versions of the same data at the same time;
- The capability of being notified about the impacts of a data change on linked data;
- Etc.

Collaboration is a feature that must be analyze when an organization wants to get rid of a spreadsheet-based process.

Collaboration around business object lifecycle can indicate the presence of complex collaboration processes between teams. The "concurrent engineering" is one of them, the idea being that the design phase is done taking into account one or several other phases of the product lifecycle, such as the manufacturing of products or their support.

## 4. Applicability management

Generally, in spreadsheets this functionality is basic and represented by one or several columns with applicability indicators, typically:

- Applicable to case  $|1|$ ,
- Applicable to case  $|2|$ ,
- ...
- Applicable to case  $|N|$ ,
- Applicable to all cases.

For sure, those indicators can become quite complex in complex spreadsheets with macros. As we explained in the article about [configuration management in PLMs](#), the manufacturing area needs to manage more or less complex configurations.

Most of time, in spreadsheets managing some aspect of the lifecycle of some business objects, there is a notion of applicability management that opens to the management of some configurations.

## PDM and detailed-design centric PLM

Most manufacturing companies use now PDM systems for their core detailed design process. Because many disciplines are collaborating to the design of a complex product (mechanical engineering, electrical engineering, electronics, engines, etc.), the product detailed design process was put in the center of preoccupations in many companies, which is understandable.

However, outside the detailed design world, many other domains are often still not covered by PLM software. When the manipulated business objects are requiring the four core functions we just discussed, light PLM engines can provide the four basic functions we just mentioned.

We think this phenomenon will generalize in the coming years, as fast as the maturity of the industry to realize that most of the data that are managed in spreadsheets need the four core functions we just detailed.

## Extending the scope of PLMs to several business domains

Modern PLM systems are able to manage many business object lifecycles. In most of them, it is also possible to easily implement your own business objects and to define their custom lifecycle, benefiting from the PLM engine.

Considering those capabilities, the question is: Why not integrate many new business objects coming from various business domains inside a single PLM that would be in charge of the management of all the business object lifecycle, on top of the existing detailed design process? This approach exists in the ERP (Enterprise Resource Planning) world and why wouldn't it be possible in the PLM world?

This question is a complex one and it is linked to the optimal mapping of several interlinked business object lifecycles. We already brought some elements of response in previous articles such as [the real nature of data](#). We intend to bring more elements in a next article of this series, dedicated to the question.

## Spreadsheet replacement risks

If the industrial area can reuse the same approaches used in the tertiary sector for some processes, there are many cases where the basic solutions of the tertiary sector are not applicable, due to the specific core requirements attached to data we just saw.

The most classical cases are to try to replace a spreadsheet by a custom development, typically:

- A specific web application,
- A customized content management system (CMS).

In the first situation, the focus is made on the data management whereas in the second situation, the focus is made on attached documents.

In both cases, if the business objects that are manipulated are requiring the four core functions we have detailed, the development team may progressively discover the extent of the complexity managed manually outside the spreadsheet. This can jeopardize the success of the project.

Many projects that were appearing as simple reveal themselves as being very complex. This situation happens regularly when IT service companies took commitments without understanding the full complexity of the project.

## The difficulty of replacing spreadsheet-based processes

We think the need for the four core functions we detailed indicates that there is a need for a PLM engine, even if it is a simple one, and that need cannot be easily satisfied with other more common IT solutions (unless people want to redevelop those four core functions).

In the manufacturing business, for instance in the aerospace, many people ask why so many processes are based on spreadsheets. A possible explanation is that, considering the very nature of industrial data, it is often surprisingly complex to replace some manual spreadsheet-based processes.

If your business data need the four functions we detailed, you need a PLM.

## Notes

- **Note 1:** We can note that it appears more and more important to also track the very early choices, or "design justification". Quite often, during the early design phase, system engineering methods were applied in order to perform complex trade-offs that lead to a certain product architecture. This can be seen as an "generalization" of the change management process, i.e. the capability of going back in the past to find back the original design choices.

## See also

The series of articles about PLM:

1. Article 1: [PLM and graph data](#)
2. Article 2: [Configuration management of industrial products in PDM/PLM](#)

Other articles:

- [The real nature of data](#)

(July 2021)