Managing Variants

Version: 1.0
Status: ACTIVE
Date: 19-June-2017

Executive Summary

Many organisations use multiple variants of the same business process - whether catering for different product lines, different jurisdictions or simply different methods of working in different locations.

Each variant carries additional costs, in areas like training (and employee mobility), technology (including support and licensing), manpower (including specialist skills) and metrics (comparing like with like).

In practice, very few organisations manage to avoid the use of variants altogether. This document describes recommended techniques for avoiding, mitigating, and modelling variants.
Document Control

<table>
<thead>
<tr>
<th>Title</th>
<th>Managing Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Document</td>
</tr>
<tr>
<td>Author</td>
<td>Business Optix</td>
</tr>
<tr>
<td>Style</td>
<td>Standard PDF v6.1.xsl</td>
</tr>
<tr>
<td>Owner</td>
<td>Business Optix</td>
</tr>
</tbody>
</table>

Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Author</th>
<th>Details of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>13-March-2016</td>
<td>ACTIVE</td>
<td>Business Optix</td>
<td>Managing variants</td>
</tr>
</tbody>
</table>
## Contents

1 Overview

2 Costing and Justification

3 Process Levels

3.1 Linkage between levels

3.2 Operational level variants

3.3 End-to-end process variants

3.4 Detailed process variants

3.5 Procedural level variants

4 Implementing Variants

4.1 Variant models

4.2 Variant folders

4.3 Minimising duplication

5 Conclusion
1 Overview

This document provides:

• Recommendations on how to avoid unnecessary variants, using Costing and Justification
• Recommendations on how to minimise the impact of necessary variants, using Process Levels
• Recommendations on Implementing Variants
2 Costing and Justification

Whenever a variant is identified, the operational costs of supporting that variant should be determined. These may include:

- **Training**: do users require different training to use or manage different variants? Is the mobility of employees within the organisation affected?
- **Technology**: does supporting multiple systems for the same function incur additional support costs or license fees?
- **Manpower**: does the variant require the use of expensive specialists?
- **Metrics and auditing**: is there a need for manual collection or translation of data from the variant in order to provide like-for-like information that can be rolled up to organisational level?

Process owners (and finance) may be unaware of these hidden costs: bringing them to light may swing them towards adopting the standard process.

If not, then a detailed justification should be provided for each variant, and made a key subject for review and approval processes.
### 3 Process Levels

Aside from the organisational costs associated with variant processes, variants may lead to increased maintenance costs within the process modelling environment. For example, a regulatory change might require a change to every product-based variant of a process, or a product change might require changes to every regional variant.

The concept of process levels can (amongst other uses) help to minimise these maintenance costs.

Process models may be defined at different levels of abstraction, with users able to drill down from one level to another. The precise number and labels for these levels may vary from one organisation and methodology to another (Six Sigma, for instance, uses levels 0 thru 5) but a common convention is the four level model where:

- **Level 1** maps the *operational* areas and capabilities of the entire organisation, using an Operating Model or Business Model Canvas diagram
- **Level 2** describes each *end-to-end process* across operational areas, using high-level process maps
- **Level 3** identifies the *detailed process* activities, roles, deliverables and interfaces needed to perform the part of a level 2 process falling within a single operational area, using detailed process maps
- **Level 4** provides the *procedures*, work instructions and system documentation needed to perform each step of a level 3 process, using any combination of text, tables and (typically ad-hoc) diagrams. Much of this material may originate outside the process modelling environment, and either be brought into the modelling environment or accessed in situ by hyperlink.

The remainder of this section makes use of this terminology - but the arguments and recommendations hold equally well for others, all of which map more or less closely to the Operational, End-to-End, Detailed and Procedures levels described above.

### 3.1 Linkage between levels

Whilst the number, and names, of these levels may differ from one organisation to another, the linkage between levels should always use a standard drill-down, from a sub-process activity defined at one level to a process defined at the next level down.

Avoid defining a lower-level process as a copy of (all or part of) a higher-level process, with more detail added: if you do this, then whenever a change is needed at the higher level, it will have to be made again at the lower level (with the risk of mistranslation as well as the additional effort).

Instead, in the higher level use a single sub-process activity to represent the lower level process.

As well as providing drill-down from one level to another automatically, this approach will tend to push variants down to lower levels, where they are both easier for consumers to understand and easier for authors to maintain.

Not all sub-process drill-downs need be to a process defined at a lower-level: some (especially at the detailed process level) may be to more detailed sub-processes described at the same level. Users will simply drill-down as many times as they need to reach whatever level they are interested in, and can move back to a higher level using the breadcrumb trail displayed above the model.

Whereas sub-processes at the same level as their calling activity are often managed in the same model, different levels should always be managed in different models.
Drill-down from non-process models (such as Operating Model or Business Model Canvas diagrams) to non-process models (like procedures and work instructions) should use exactly the same mechanism. In the latter case, the target of a drill-down will often be a hyperlink to a document held elsewhere in the corporate environment (or the internet at large).

3.2 Operational level variants

Variants seldom arise at the operational level: when they do, it is often a sign of too much detail being put in for the level: see if any of that detail can be moved to the end-to-end process level.

Variants can occur at this level in some large or highly heterogeneous organisations - or immediately after a merger. In such cases, if the intention is not to bring the different parts closer together, consider modelling the overall organisation as a group of smaller organisations, where some may provide shared services to others.

3.3 End-to-end process variants

Variants are less unusual at the end-to-end process level - but again, are often a sign of too much detail being put in for the level: you should always try to move variant details down to the detailed process level, because:

- Variants defined at this level tend to become 'set in stone': any future attempt to replace such a variant is likely to impact a multitude of downstream detailed processes.
- The existence of end-to-end variants encourages the further use of variants downstream. An organisation unwilling to standardise processes at the corporate level is unlikely to do so at lower levels.

The advice given under 'Linkage between levels' above will often ensure that variants are defined at the detailed, rather than end-to-end, level.

3.4 Detailed process variants

Whilst it is usually possible to push variations down from an end-to-end process into underlying detailed process models, pushing them down from detailed process to procedural level may not be so easy.

The most compelling reason for variants at this level is different legislation in different countries, which often mandates actual process differences, rather than simply different procedures within individual activities. At the other end of the spectrum, variants may be wanted to avoid the cost of re-training staff to use an unfamiliar standard procedure.

Thus variants are fairly common at the detailed process level - though even here, wherever possible variations should be moved down to the procedural level.

3.5 Procedural level variants

At this level, even in the most homogeneous companies, variants can arise to at least some extent. Technology differences may be removed over time; regulatory differences are outside organisational control. Between these extremes, things like variations between different service or product offerings are often a matter of corporate taste and direction.
An advantage here is that variants at this level cannot affect lower levels, since there are no lower levels.
4 Implementing Variants

Whenever a variant is required, three areas are of concern to the process modeller:

- Ensuring users can readily access the relevant (to them) variant
- Providing a way of discovering all variants of a standard process
- Minimising the need to duplicate changes across multiple variants when the standard process itself is changed

4.1 Variant models

Use a separate model for each variant, and use the BusinessOptix variant mechanism. This allows the author to identify, at the model level, the standard model that a variant specialises - along with the cost and justification for the variant.

This mechanism allows users to list and select from all the variants of a standard process.

4.2 Variant folders

Store variant models in folders, named for the area of the business (e.g. by geography or product) for which the variant is specific - and give all variants of a given standard model the same name, based on that of the standard model.

For larger organisations with multiple departments, each with their own process team, this approach can be combined with the use of BusinessOptix groups to give a named group of users privileged access to its own set of variants. This reduces information overload, by providing users with only the information that is relevant to them.

4.3 Minimising duplication

Variant models and folders do not, on their own, address the need to minimise duplication. Where a variant is very different from the standard process, this may not be a real problem since most changes to the standard will result in quite different changes to the variant.

Where a variant differs only in detail, the recommended approach is to isolate the variant part in a separate sub-process, called from a sub-process activity in the standard model.

However large or small the difference, the variant sub-process should still be defined in a separate variant model and folder, to enable the standard variant reporting and model selection mechanisms.
5 Conclusion

Process variants should be treated as a necessary evil. Wherever a variant is identified, its cost should be determined and a justification presented, to allow the business to make an informed decision on the continued need for the variant.

Where the need is justified, variants should be modelled at the lowest possible level in order to minimise the cost of maintaining multiple variant models when the standard process must be changed for any reason.

At whatever level, each variant should be defined in a separate variant model, linked to the standard model, and stored in a folder along with any variants to other processes required for the same reason.