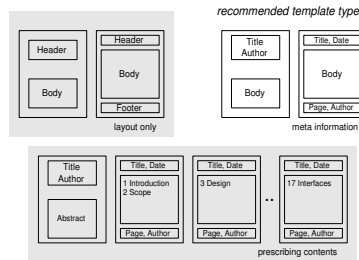


# Template How To

-



Gerrit Muller

Embedded Systems Institute

Den Dolech 2 (Laplace Building 0.10) P.O. Box 513, 5600 MB Eindhoven The Netherlands

`gerrit.muller@embeddedsystems.nl`

## Abstract

The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process.

Unfortunately the benefits of templates are often cancelled by unforeseen side-effects, such as unintended application, inflexibility, and so on. This intermezzo gives hints to avoid the **Template Trap**, so that templates can be used more effectively to support introduction of new processes.

### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudí documents are available at:  
<http://www.gaudisite.nl/>

version: 1.6

status: draft

February 10, 2011

# 1 Introduction

The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation serves mainly the purpose of a smooth introduction of the new process.

Unfortunately the benefits of templates are often canceled by unforeseen side-effects, such as unintended application, inflexibility and so on. This intermezzo gives hints to avoid the **Template Trap**, so that templates can be used more effectively to support introduction of new processes.

Templates are used for all information based entities, such as documents, mechanical CAD designs and SW code. The information in this document applies to all these categories, although the text focuses on document templates.

## 2 Why Templates?

The rationale behind the use of a template is:

- Low threshold to apply a (new) process (1)
- Low effort to apply a (new) process (2)
- No need to know low level implementation details (3)
- Means to consolidate and reuse experiences (4)

Some common false arguments are:

- Obtain a uniform look (5)
- Force the application of a (new) process (6)
- Control the way a new process is applied (7)

Argument 5 is a bogus argument, uniformity<sup>1</sup> is not something to strive for, see section 9. Arguments 6 and 7 are the poor man's solution for lack of leadership and signals a dangerous disrespect for the target group.

## 3 New Process Introduction

Process Improvement drives result in enforcing existing processes or introduction of new processes. Any change introduces reactionary behavior ( $action = -reaction$ ), urging the process improvement people to introduce the change in such a way that this reactionary behavior gives a minimal damage, see figure 1.

---

<sup>1</sup>This will be elaborated in a future Intermezzo, *The Uniformity Trap*.

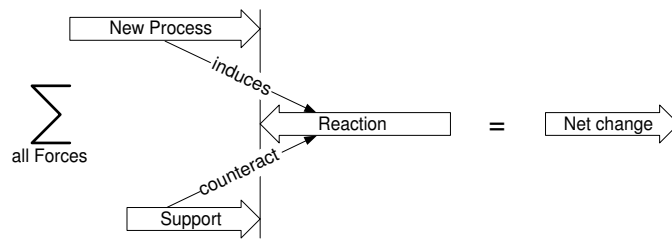


Figure 1: The reactionary force induced by the proposed new process is countered by giving support

The most frequent way to introduce a new process is to supply the means for the implementation of the process, in other words the emphasis is on the **how**, not on the **why** nor on the **what**. Figure 2 shows the relation between a process and a template. The process itself focuses on **why** and **what** (and who and when), while the procedure, the tools and the template are the **how**.

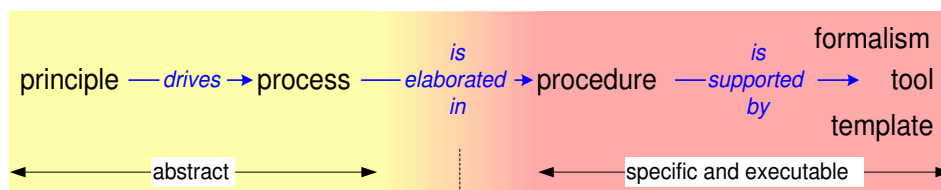


Figure 2: The relation between a template and a process

## 4 What does a Template contain

A template can support from *layout only* up to *complete contents standard*. Figure 3 shows a number of examples, Table 1 summarizes the characteristics. A layout only template does not have any notion of the information which will be in the document, nor does it presume anything about the process in which it is applied.

A template with meta information supports the process in which it will be applied. The template has knowledge of the meta-information of the document and supports both the layout of the document as well as the presentation of the meta-information in this layout.

A template which prescribes the structure of the contents of the document has knowledge of the domain as well.

**Recommendation:** Use a template for layout and a minimum meta information set.

Avoid using a template to structure the contents. A documentation structure needs to be **designed**, see [1]. To help people in this design process of documen-

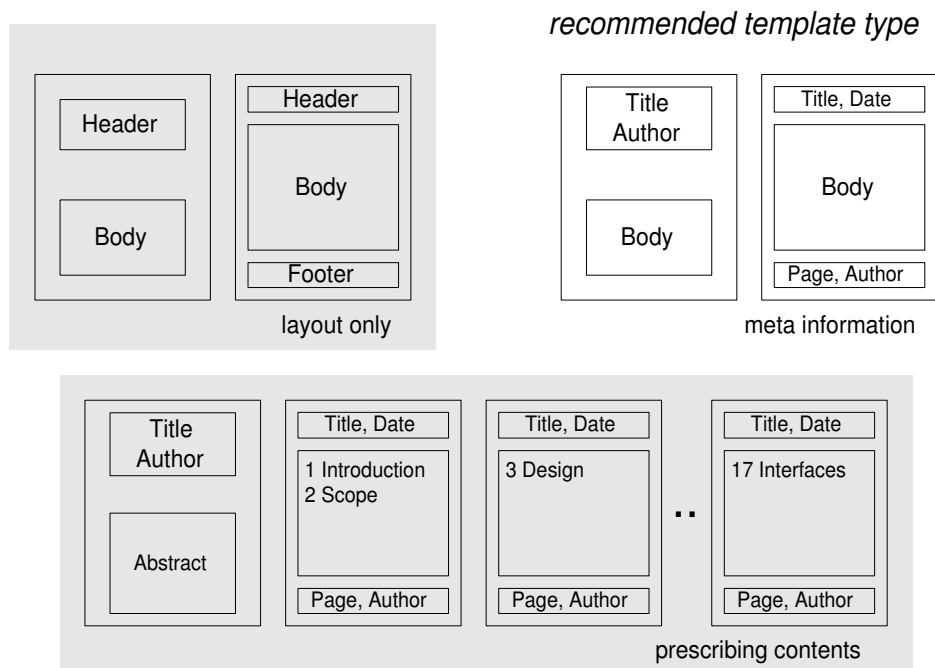


Figure 3: Templates from layout only, up to prescribing structure or contents, templates to support meta information are recommended.

tation guidelines containing checklists are effective. Templates invite people to generate "noisy" chapters (which should not have been present at all), or to write monolithic documents (because the entire checklist is present in one template). Guidelines with checklists at the other hand only mention contents, without suggesting any modularity yet.

**Recommendation:** Use checklists for structure and contents.

template type	context knowhow	value
layout only	no	low
meta information	process	high
prescribing content	process and domain	constraining

Table 1: *Overview of Template characteristics*

## 5 Copy Paste Modify Pattern

The understanding of the *copy paste modify pattern* will help to use templates effectively. The dominant implementation<sup>2</sup> strategy is the *copy paste modify pattern*:

- Look for a similar problem
- Copy its implementation
- Modify the copy to fulfil the new requirements

Majority of the work is to select the parts to be copied (or remove the unneeded parts) and to substitute the problem specific names, variables, functions et cetera.

A template is an optimization of this pattern in case of frequently reused implementations. The selection is performed once and the substitution is prepared to be easy.

## 6 Template Development

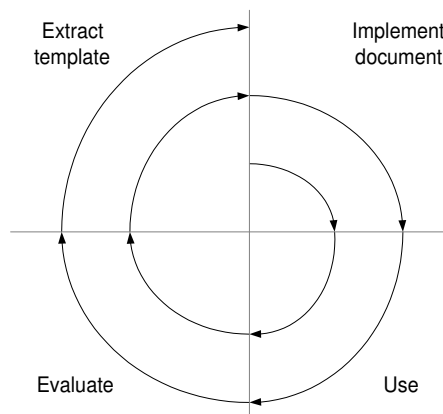


Figure 4: Spiral development model for Templates

The development of templates is basically the consolidation of experience. Once more a spiral development model best fits on the capturing of learning experiences, see figure 4. The motto is:

*Use before Re-use*

So implement a limited amount of documents (ca. 3), use these documents with other people, evaluate explicitly and extract then the template from these first

---

<sup>2</sup>This holds for all information based implementations, from mechanical CAD drawings to management spreadsheets

documents. Implement the next set of documents on the basis of this template and repeat the same use, evaluation and extraction process. Keep repeating this forever!

## 7 Guidelines

A template is applied to support one or more processes. The deployment of the template is enabled by guidelines describing the way it should be used. The guidelines must be classified in mandatory rules and recommended practice.

An example of guidelines for meta information of a document is:

**Mandatory on every page**

- Author
- Title
- Status
- Version
- Date of last update
- Unique Identification
- Business Unit
- Page number

**Mandatory on every document** on top of the mandatory information per page

- Distribution (Notification) list
- Reviewers and commentators
- Document scope (Product family, Product, Subsystem, Module as far as applicable)
- Change history

## Recommended Practice

- Short statement on frontpage stating what is expected from the addressed recipients, for example:
  - Please send comments before february 29, this document will be reviewed on that date
  - This document is authorized, changes are only applied via a change request
- See Granularity of Documentation [1] for guidelines for modularization and contents

The example defines a minimum mandatory set. No layout guideline is given except the fact that a subset of the meta information is mandatory on every page.

This illustrates a very important aspect of templates:

*The procedure is mandatory, the template is only an enabling means*, which means that anyone can make its own template as long as it fulfills the mandatory rules of the procedure.

## 8 Pitfalls

The most frequent pitfalls in the application of templates are:

- Author follows template instead of considering the purpose of the document.
- Template is too complex.
- There is an unmanageable number of variants.
- Mandatory use of templates results in:
  - no innovation of templates (= no learning)
  - no common sense in deployment
  - strong dependency on templates

A tendency exists to put a lot of information and intelligence in a template. For instance specialized Word templates, which prompt for the required fields. These kinds of templates are very vulnerable with respect to tools and environment. Changes in tools, environment or process play havoc with these nice looking templates. Good templates are, as good designs, simple. Simple templates are easily understood and easily modified, providing flexibility, room for innovation and room for common sense by customization to the problem.

In due time the amount of specialized templates grows. As in a normal design

re-factoring is required to keep the overall set simple and consistent and hence maintainable.

The most common pitfall is to make the template mandatory instead of making the procedure mandatory. In other words the **how** is enforced instead of the **what**. This is the main cause of all the following pitfalls, such as no innovation, no common sense and a strong dependency on templates. The mandatory use of templates inhibits the innovation and common sense by individual users.

**Recommendation:** Enforce the procedure (*what*), provide the template (*how*) as supporting means.

## 9 Why I hate templates

Personally I hate templates. My way of working is based on immediate visual recognition of objects such as documents. For instance searching for a document on a large chaotic desktop is based on the visual image in my memory which is compared to the visual look of the documents on the desktop. When receiving a document the visual look immediately classifies the document with respect to author, project and status.

The uniformity caused by templates dramatically degrades my recognition performance and worse false matches turn up frequently.

This problem can be countered by allowing "personification" of documents, for instance by adding personal icons, images fonts et cetera. So by making variation on purpose!

Several people have pointed out to me that I violate my own needs for visual recognition with all Gaudí articles. Obviously here is room for improvement!

## 10 Summary

- Templates support (new) processes
- Use templates for layout and meta information support
- Do not use templates for documents structure or contents
- Stimulate evolution of templates, keep them alive
- Keep templates simple
- Standardize on **what** (process or procedure), not on **how** (tool and template)
- Provide (mandatory) guidelines and recommended practices
- Provide templates as a supportive choice, don't force people to use templates

## 11 Acknowledgements

Jürgen Müller identified the weak spots as usual, enabling an improved and more clear intermezzo. Discussion with Saar Muller helped to improve the terminology, such as guidelines and rules. The sharp eyes of Jaap v.d. Heijden helped to improve the figures.

## References

- [1] Gerrit Muller. Granularity of documentation. <http://www.gaudisite.nl/DocumentationGranularityPaper.pdf>, 1999.
- [2] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.

## History

**Version: 1.6, date: August 6, 2002 changed by: Gerrit Muller**

- minor change

Version: 1.4, date: October 5, 2001 changed by: Gerrit Muller

- layout update

Version: 1.3, date: March 7, 2000 changed by: Gerrit Muller

- sharpened the terminology with respect to guidelines, checklists and rules
- improved readability of figure 1

- extended the section *Why I hate templates*
- Version: 1.2, date: February 24, 2000 changed by: Gerrit Muller**
- Figure and Template Types name inconsistency solved
- Version: 1.1, date: February 23, 2000 changed by: Gerrit Muller**
- Figure Template Types updated
  - Template Type Table low changed in no context knowhow
- Version: 1.0, date: February 23, 2000 changed by: Gerrit Muller**
- table with characteristics of template types added
  - recommendations highlighted
  - added missing information to Presentation
- Version: 0, date: February 22, 2000 changed by: Gerrit Muller**
- Created, no changelog yet