

## 1 Introduction

The modules in the `taspresent` directory aim to provide an easy-to-use, consistent interface for writing simple presentations in ConTeXt. I had the idea to write this module when I was preparing my own presentations with ConTeXt. I wanted to be able to achieve visually different results without changing my source files, so I wrote different styles that followed the same logic and provided the same macros. When I created this module, I had the following requirements in mind:

- Most of the styles that are provided are rather sober in appearance. I use them for my university lectures in the humanities. They provide a nice but not too distracting background and a lot of space for presentations with large amounts of text.
- The module is meant for presentations which will be shown with the help of a digital projector. Hence, they have no interactive elements (such as buttons) and no tools for navigation (such as a table of contents).
- The module allows for user configurability. It comes with several predefined styles and some predefined font options. The modular structure makes it easy to add further styles.
- Picture placement and changing backgrounds is made easy by predefined macros.

The module provides a simple, basic structure; I think it will be best suited for beginners or intermediate users of ConTeXt. It is definitely not meant to compete with Hans's fuller and fancier presentation modules, and it offers much less than the L<sup>A</sup>T<sub>E</sub>X `beamer` package. On the other hand, it is much easier to use; you should be able to write your first presentation after spending five minutes with this manual.

## 2 Installation

Installation is easy: just put the files `t-<something>` into a directory where T<sub>E</sub>X can see them. For ConTeXt third-party modules, the canonical place would be in one of your TEXMF trees, under `tex/context/third`. If you want to keep things tidy, place them in a subdirectory `taspresent`. If you just unzip the archive `taspresent.zip` in a TEXMF directory, things will be moved to the right place automatically. On many T<sub>E</sub>X-systems, you will have to run `texhash` after installing new files. To doublecheck whether the system finds your files, run `kpsewhich t-taspresent.tex` from the command line; if all goes well, this should return the position of the file you have just installed.

### 3 Use

I have been using a few of the module styles extensively during the last months and haven't had any significant trouble. The module has been used both with "traditional" `mkii` and with the new `mkiv` version of `ConTeXt`; both have worked. I have not tested the module with `XYTEX`, but there's no obvious reason why it shouldn't work.

### 4 Setting up the Module

To use the module, you put this line into your source file:

```
\usemodule[taspresent][style=,font=,size=,stylecolor=,stylebottom=]
```

The values for the different keys will be explained in the following sections.

### 5 The `style` Key

There are eighteen options for the `style` key:

## 5.1 blackblue

This style was inspired by the “Copenhagen” theme of the L<sup>A</sup>T<sub>E</sub>X `beamer` package. The narrow blue and black stripes at the top and the bottom of the slides display the date and slidenummer (top) and the title and author of the presentation.

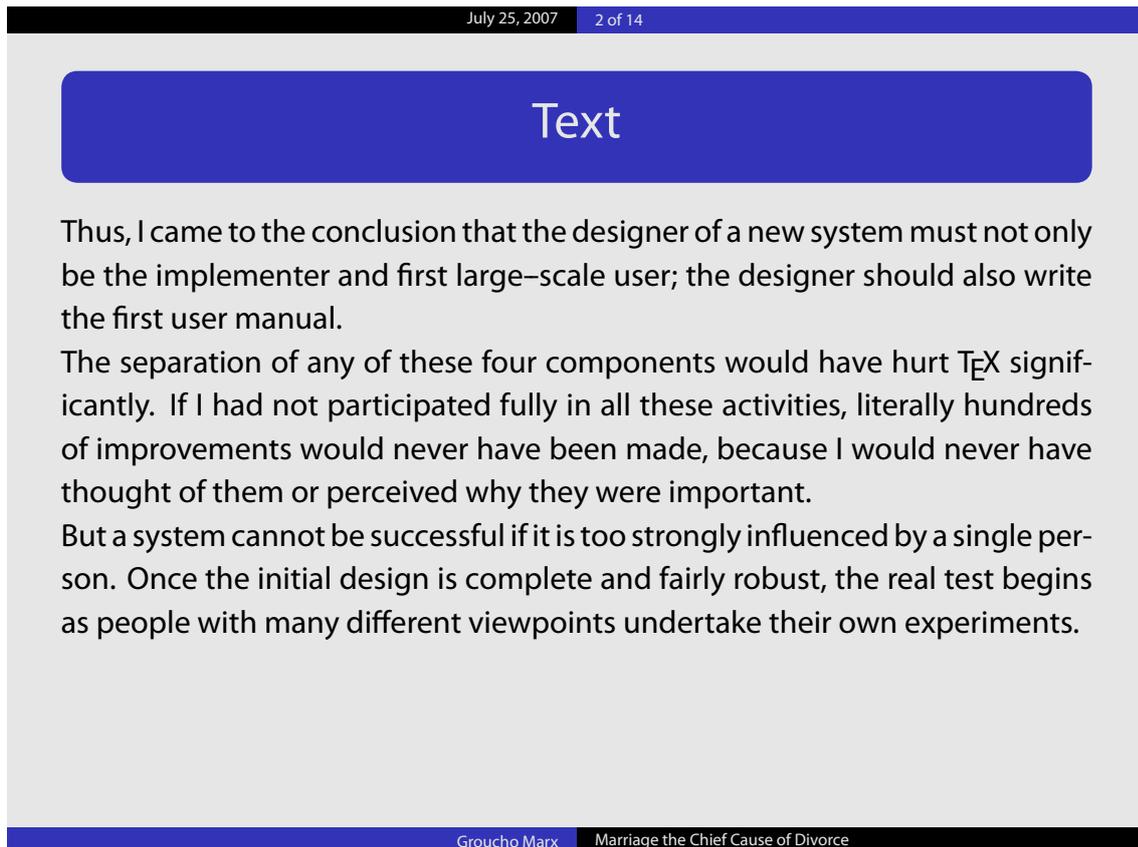


Figure 1 The **blackblue** style

## 5.2 bluegray

The colors of this style are very subdued and quiet; the interesting thing is the pagenumber on the border of the margin and text area; this detail was inspired by Hans’s “split” style (pre-14).

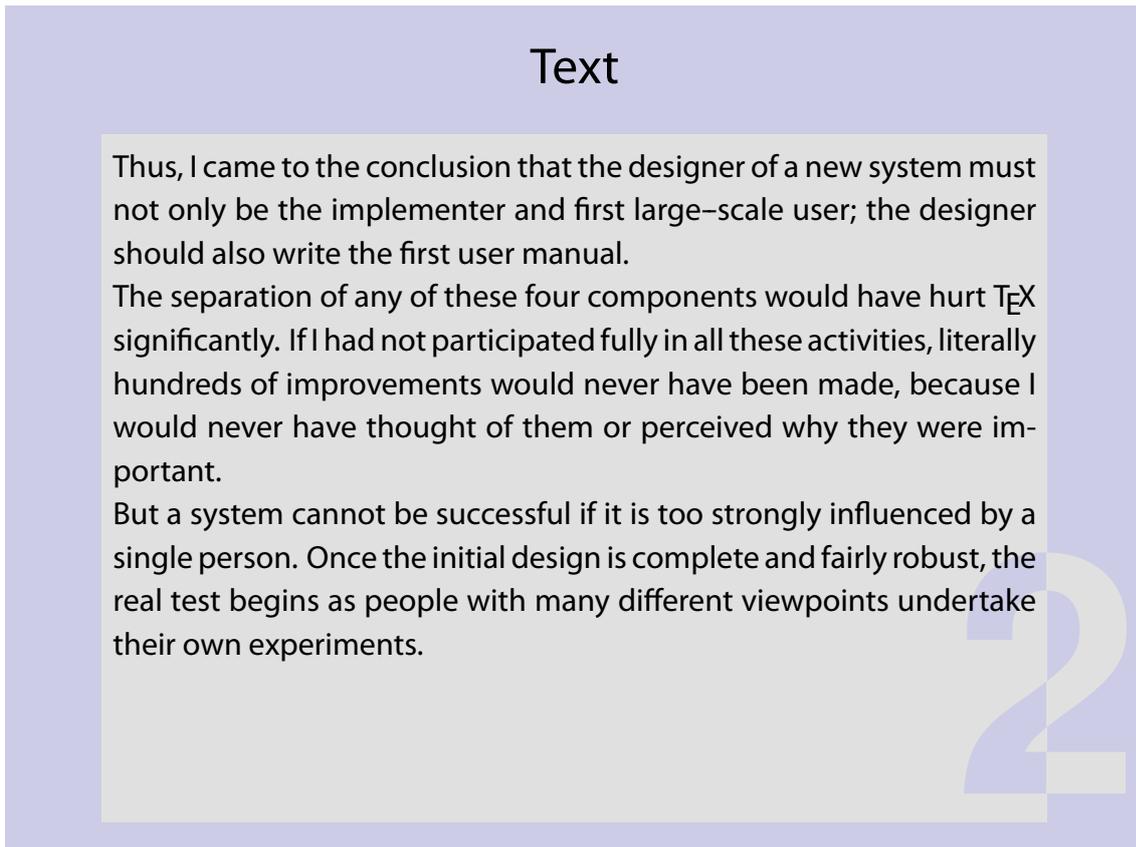


Figure 2 The bluegray style

### 5.3 [bluestripe](#)

This theme is inspired by the “Berkeley” style of the L<sup>A</sup>T<sub>E</sub>X `beamer` package. The only ornament is the little counter in the lower left corner which looks like a stopwatch.

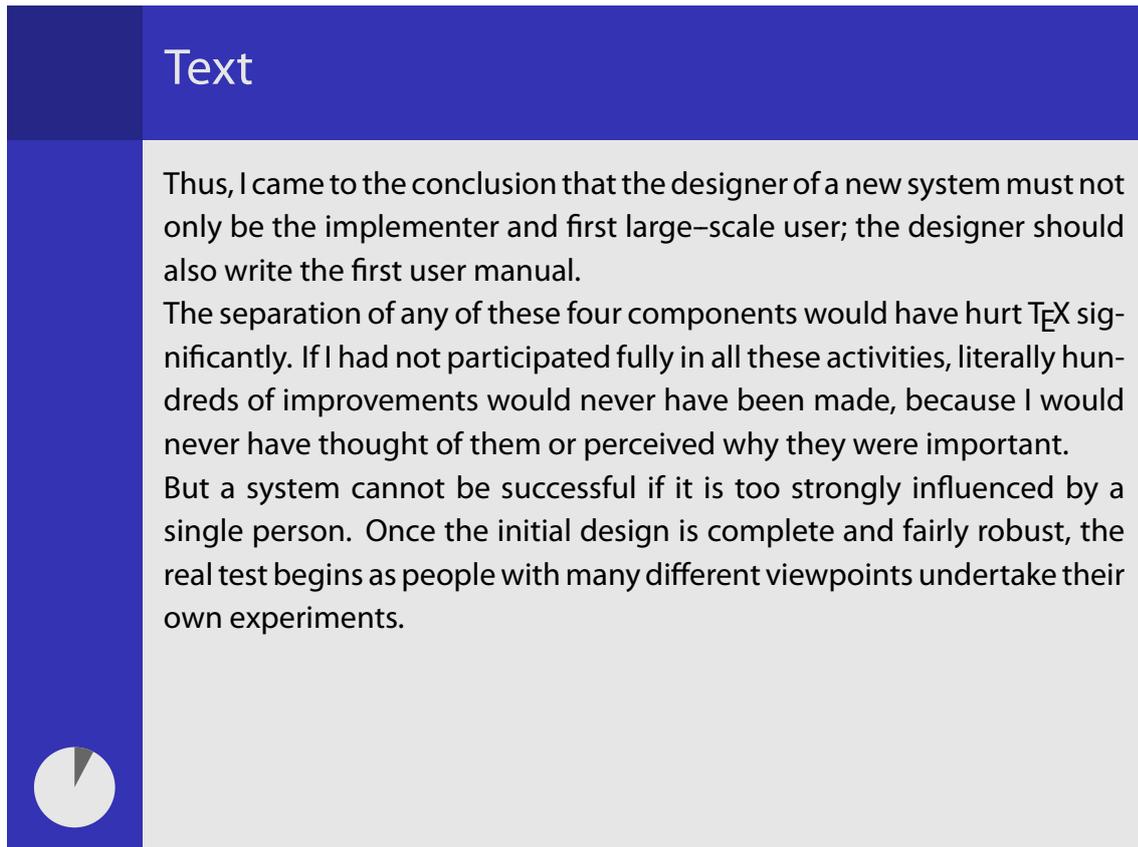


Figure 3 The [bluestripe](#) style

## 5.4 `darkshade`

The only ornament to this style is the dark shaded background; it has two style colors, `stylecolor=blue` and `stylecolor=green`; if you feel really adventurous, try `stylecolor=bluered`! It uses ConTeXt's `interactionbar` mechanism to show the progress of the presentation. It provides much space for text.

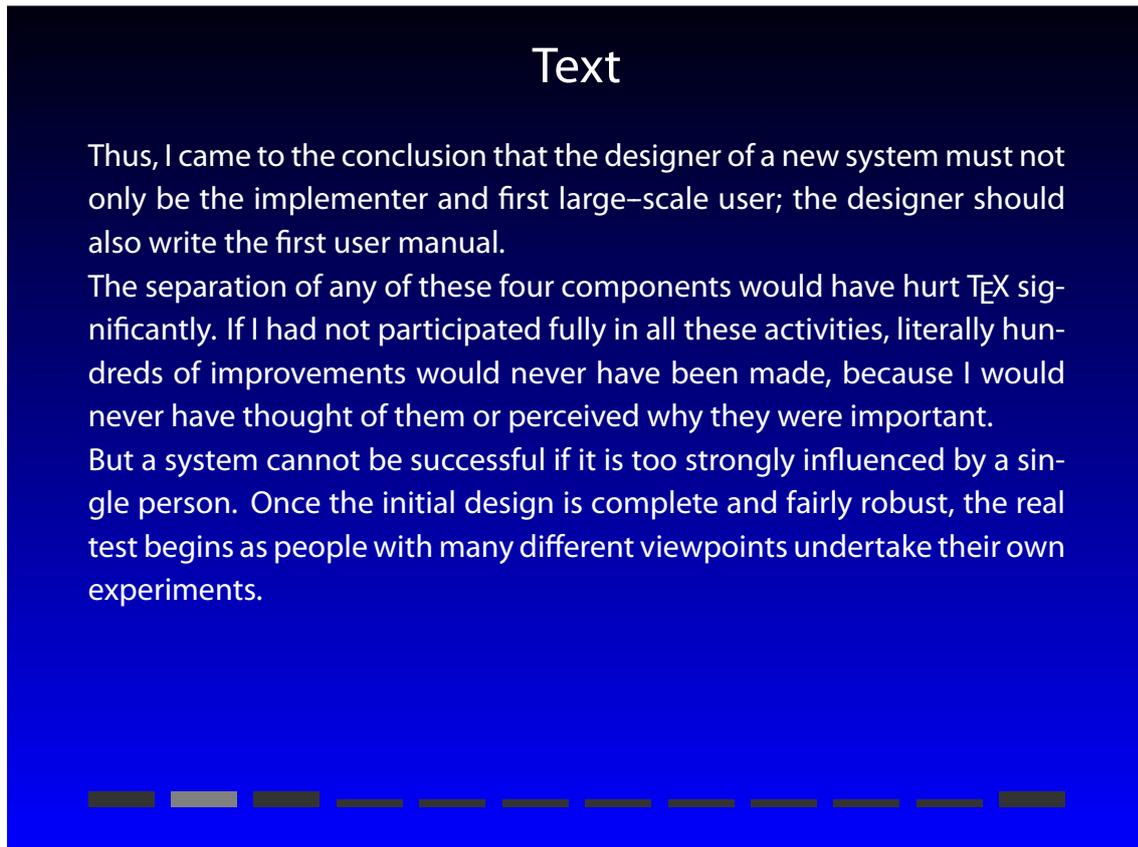
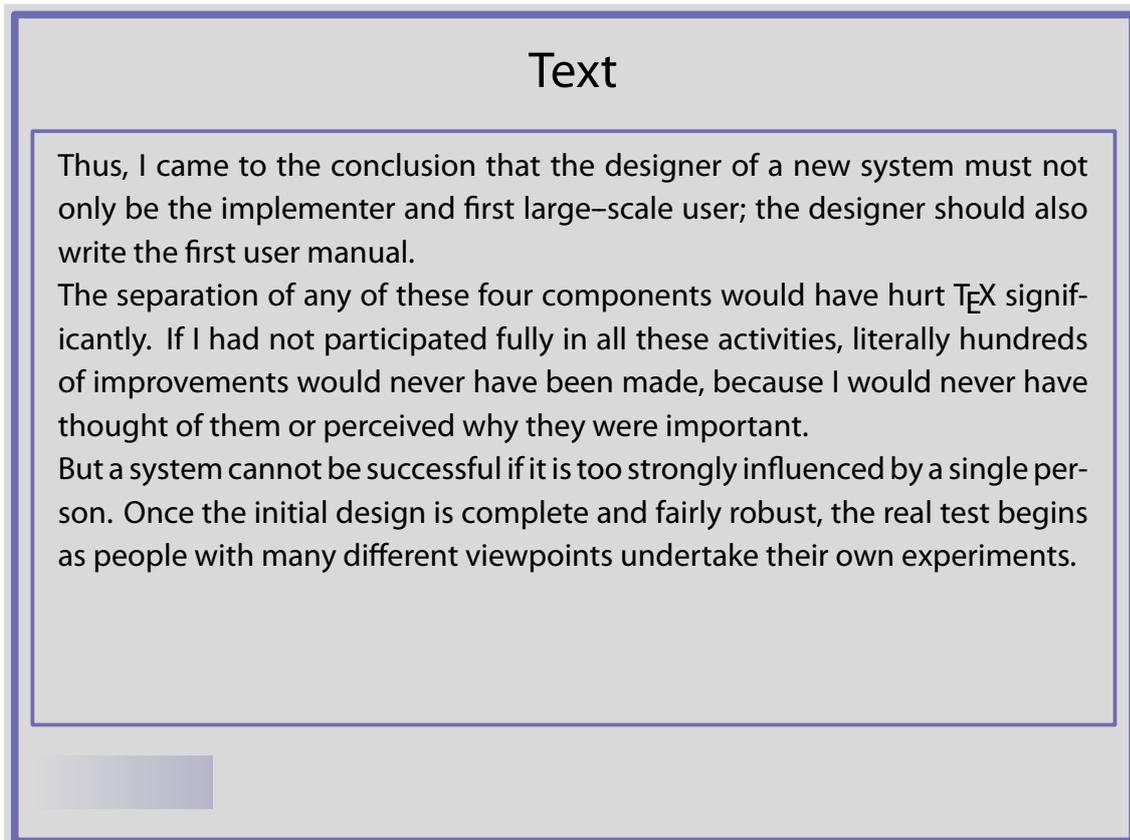


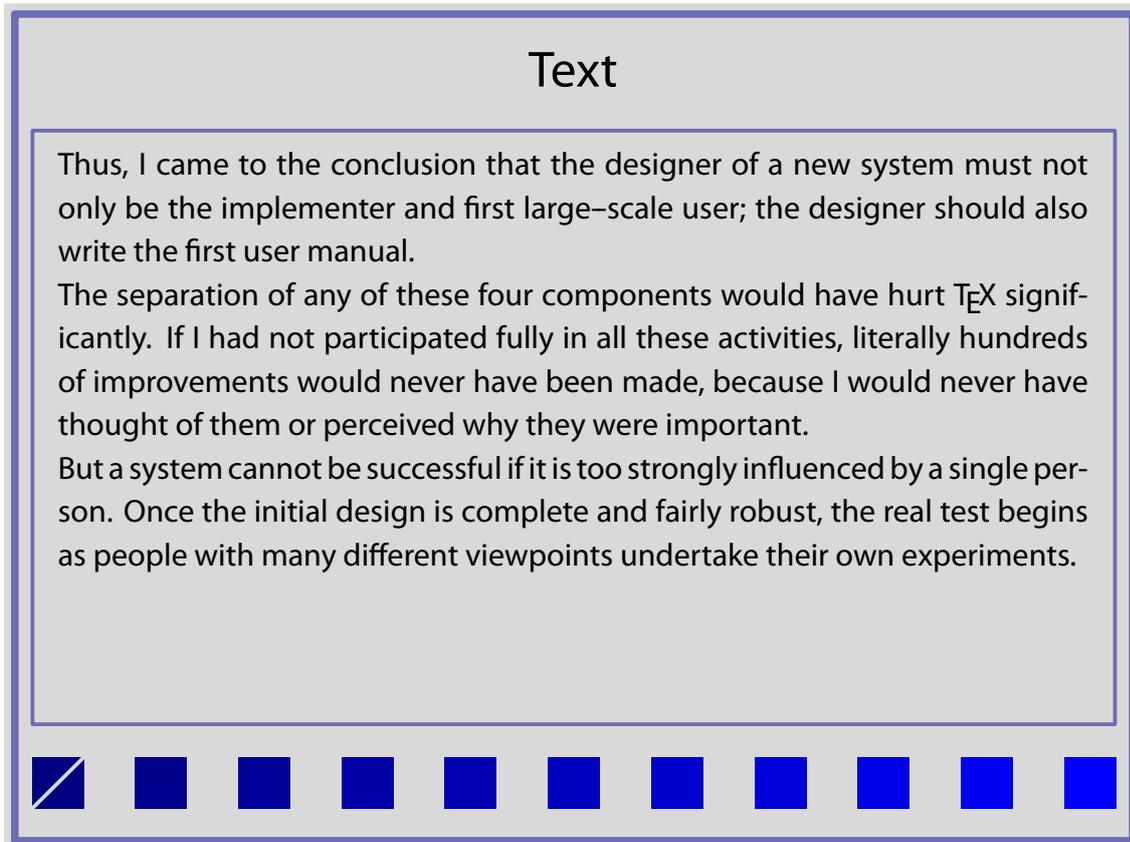
Figure 4 The `darkshade` style with `stylecolor=blue`

## 5.5 `doubleframe`

This style was inspired by Hans’s “green” style (`s-pre-02`). It has a thick blue frame around the entire slide area and a thinner frame around the text area. The style has two options for the bottom area: `stylebottom=stripe` will display a shaded blue area which will grow with each slide; `stylebottom=square` displays a row of blue squares at the bottom which also measure the presentation’s progress.



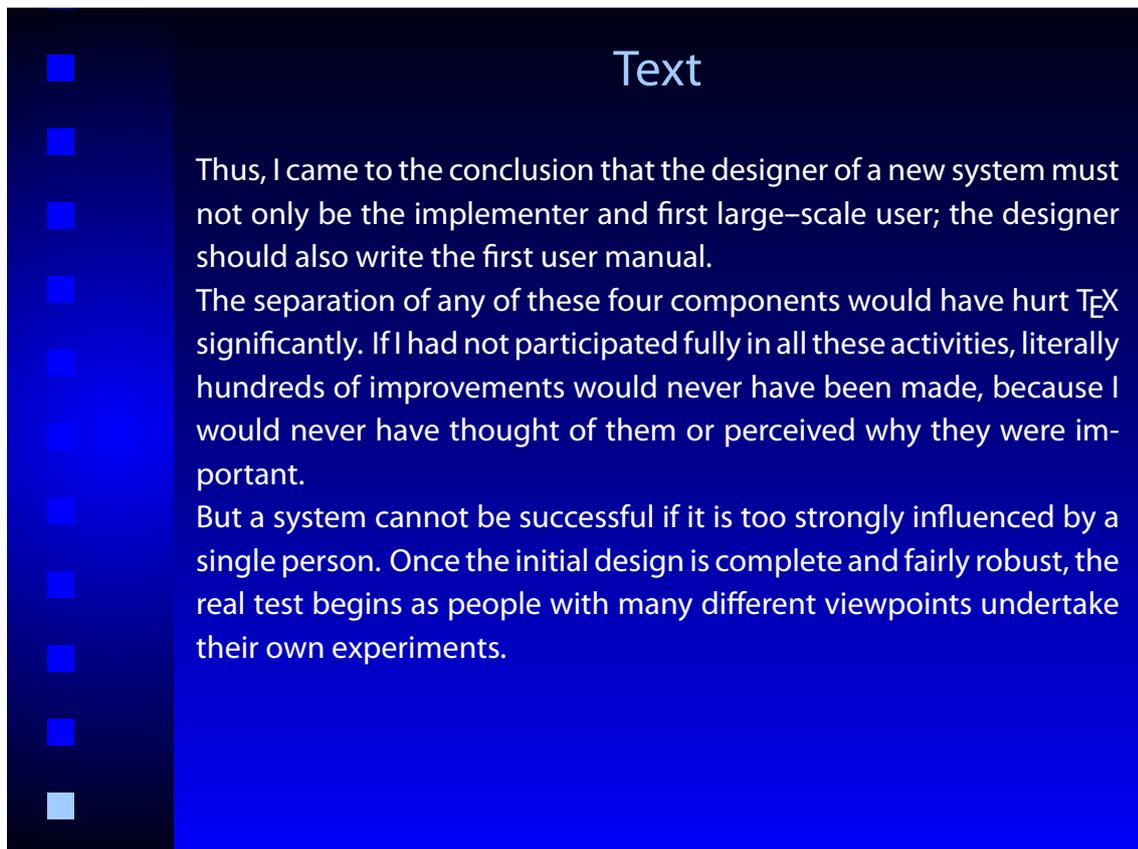
**Figure 5** The `doubleframe` style with the `stylebottom=stripe` option



**Figure 6** The `doubleframe` style with the `stylebottom=square` option

## 5.6 `doubleshade`

Similar to the `blueshade` style, but there is a differently shaded area on the left with a progress meter.



**Figure 7** The `doubleshade` style

## 5.7 embossed

Spread the word, don't be shy! Show your pride in using ConT<sub>E</sub>Xt. The color theme will probably look familiar; I copied it from the `enattab` manual.

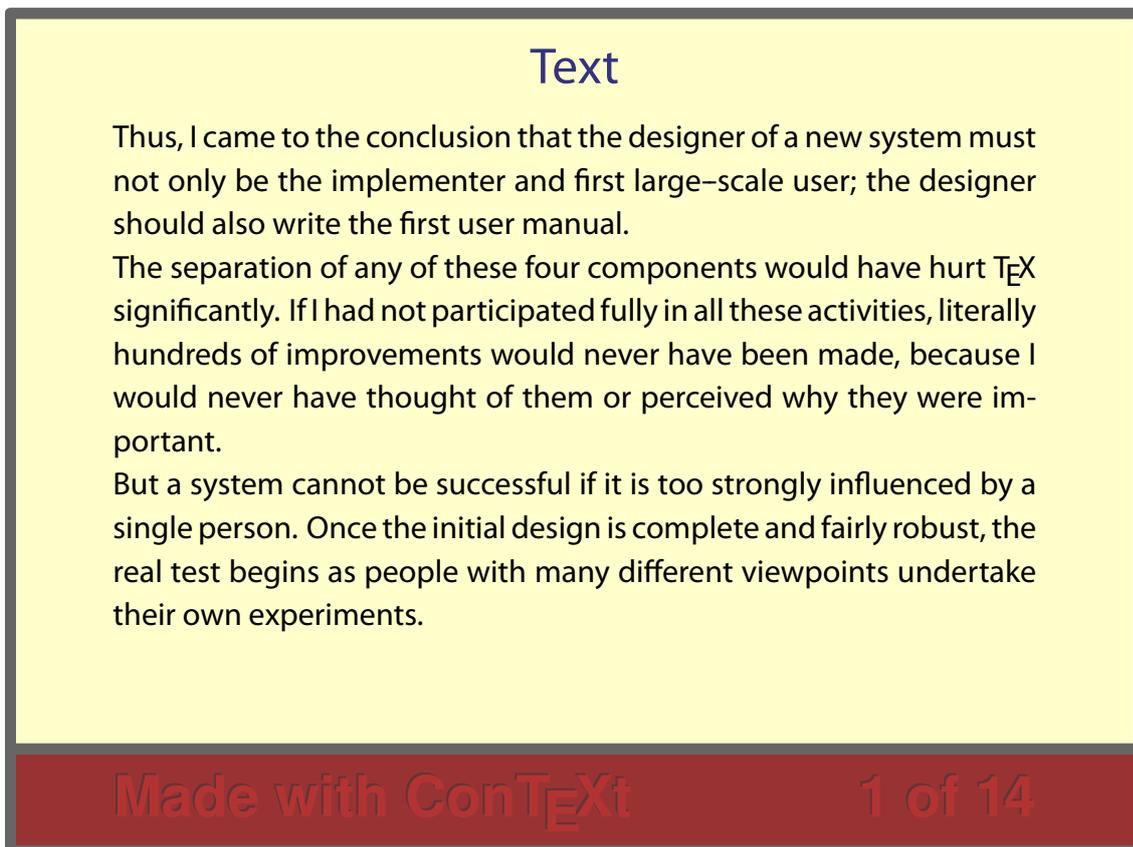
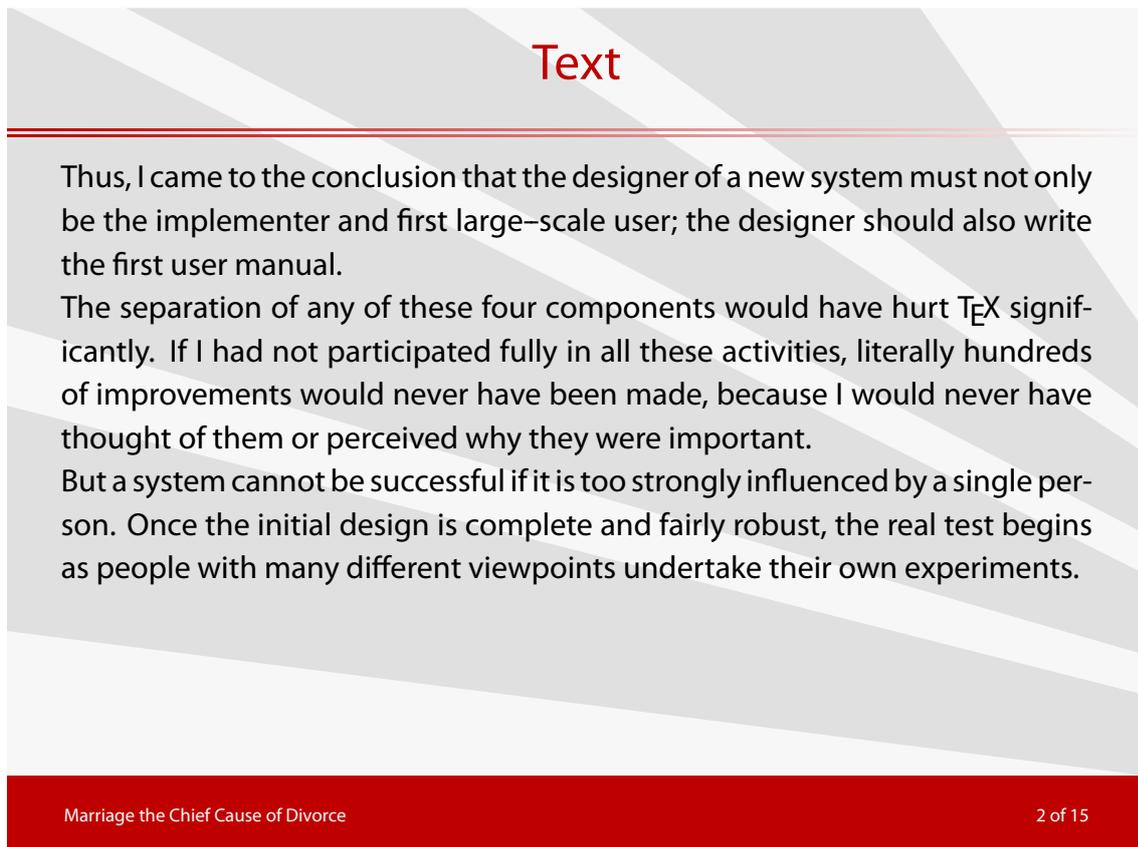


Figure 8 The `embossed` style

## 5.8 [graybeams](#)

This design is inspired by the `husky` theme for the  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  package `powerdot`, created by Jack Stalnakar.



**Figure 9** The [graybeams](#) style

## 5.9 graysquare

This minimalistic design is inspired by a presentation Taco gave at EuroTeX 2006.

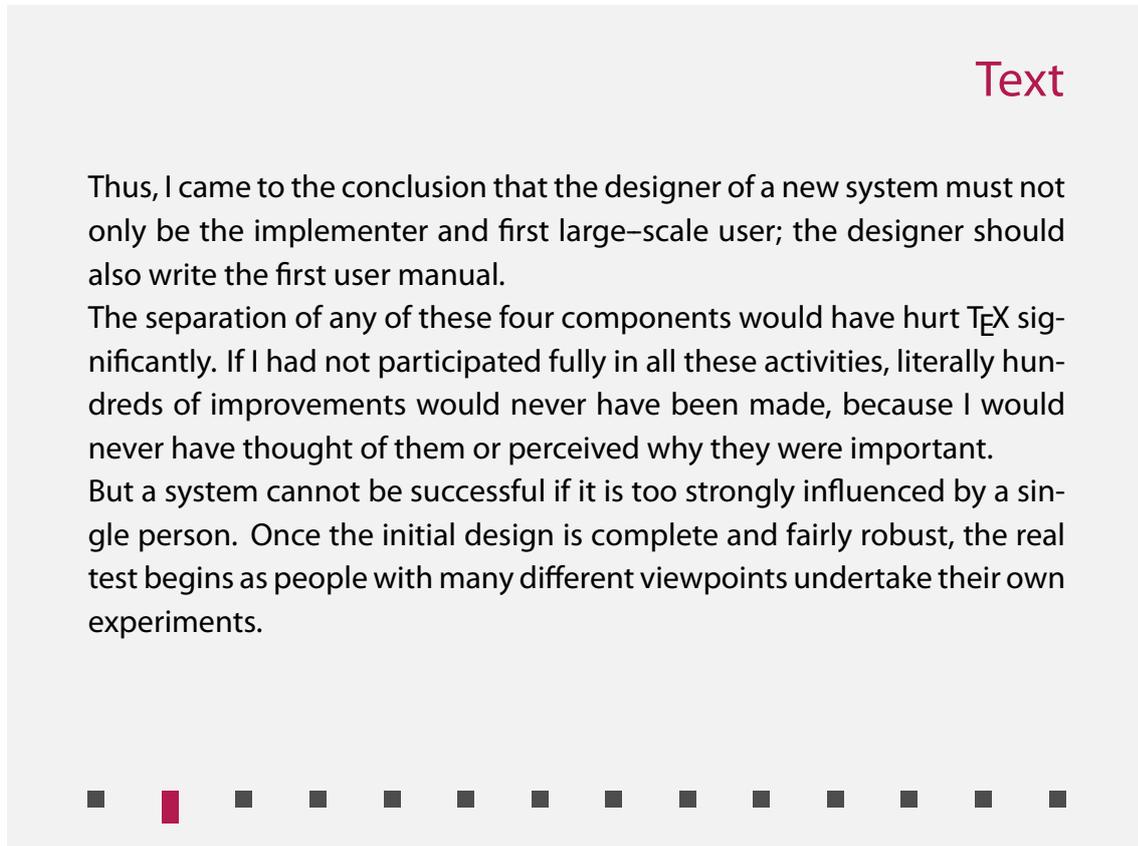
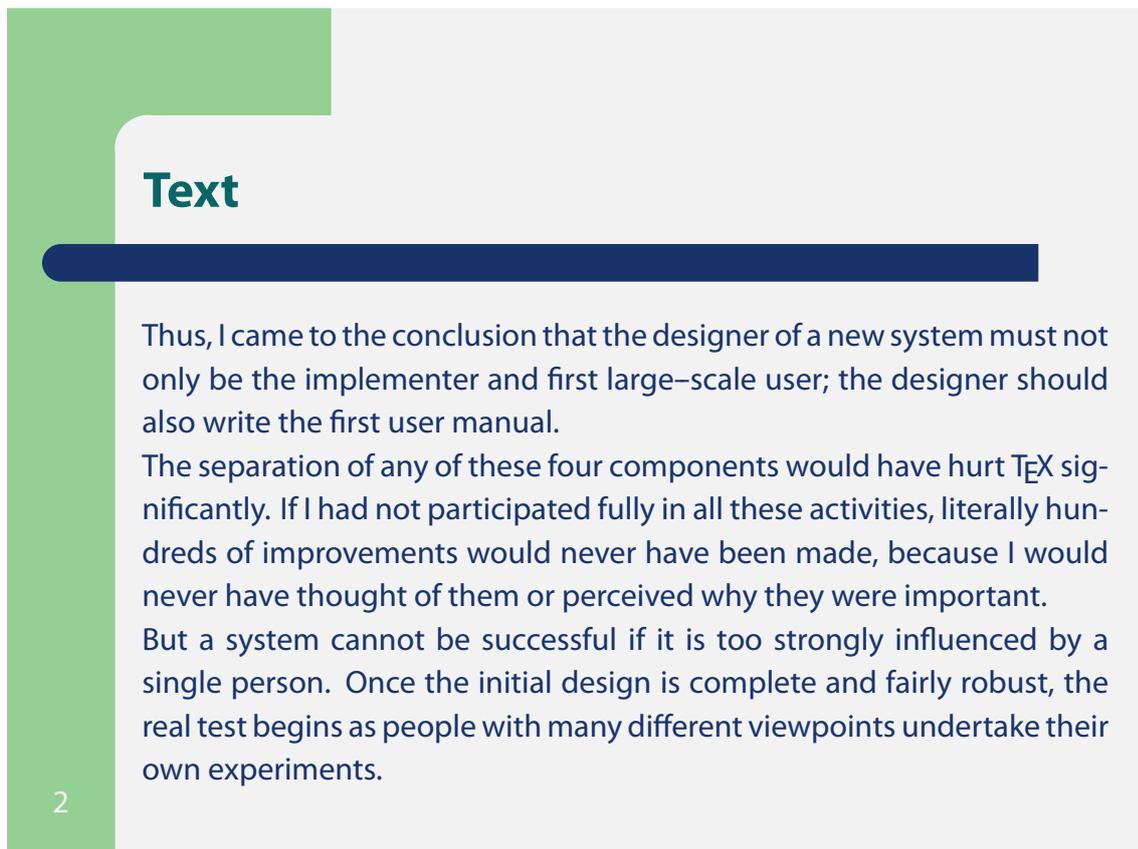


Figure 10 The [graysquare](#) style

## 5.10 greenblue

This style has cool colors and lots of white space; it is probably best suited for presentations with relatively little text.



**Figure 11** The **greenblue** style

## 5.11 `horizontalblue`

A sober style with an emphasis on horizontal lines, inspired by `beamer`'s “Szeged” theme.

Text

Thus, I came to the conclusion that the designer of a new system must not only be the implementer and first large-scale user; the designer should also write the first user manual.

The separation of any of these four components would have hurt  $\text{\TeX}$  significantly. If I had not participated fully in all these activities, literally hundreds of improvements would never have been made, because I would never have thought of them or perceived why they were important.

But a system cannot be successful if it is too strongly influenced by a single person. Once the initial design is complete and fairly robust, the real test begins as people with many different viewpoints undertake their own experiments.

Marriage the Chief Cause of Divorce 2 of 14

**Figure 12** The `horizontalblue` style

## 5.12 `lightblue`

This style emphasizes the title with its lively colors; the text area is delimited by a light blue, shaded margin.

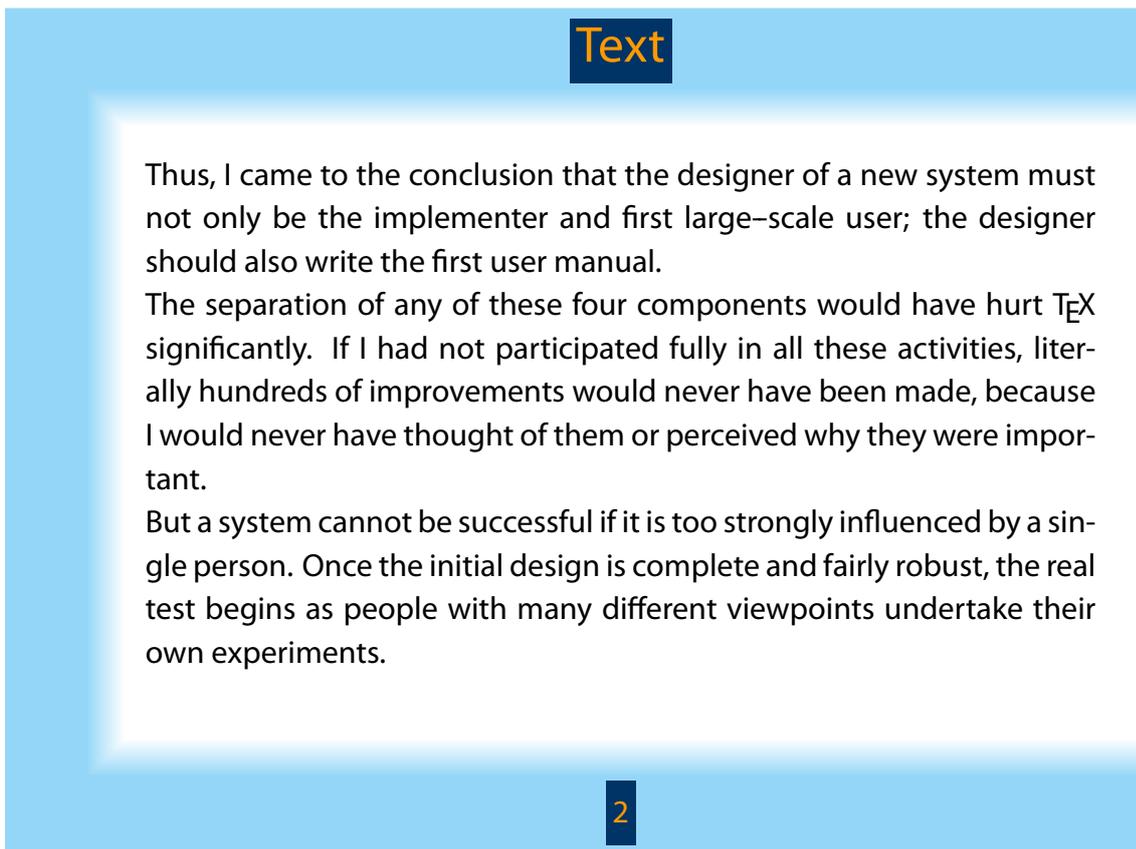
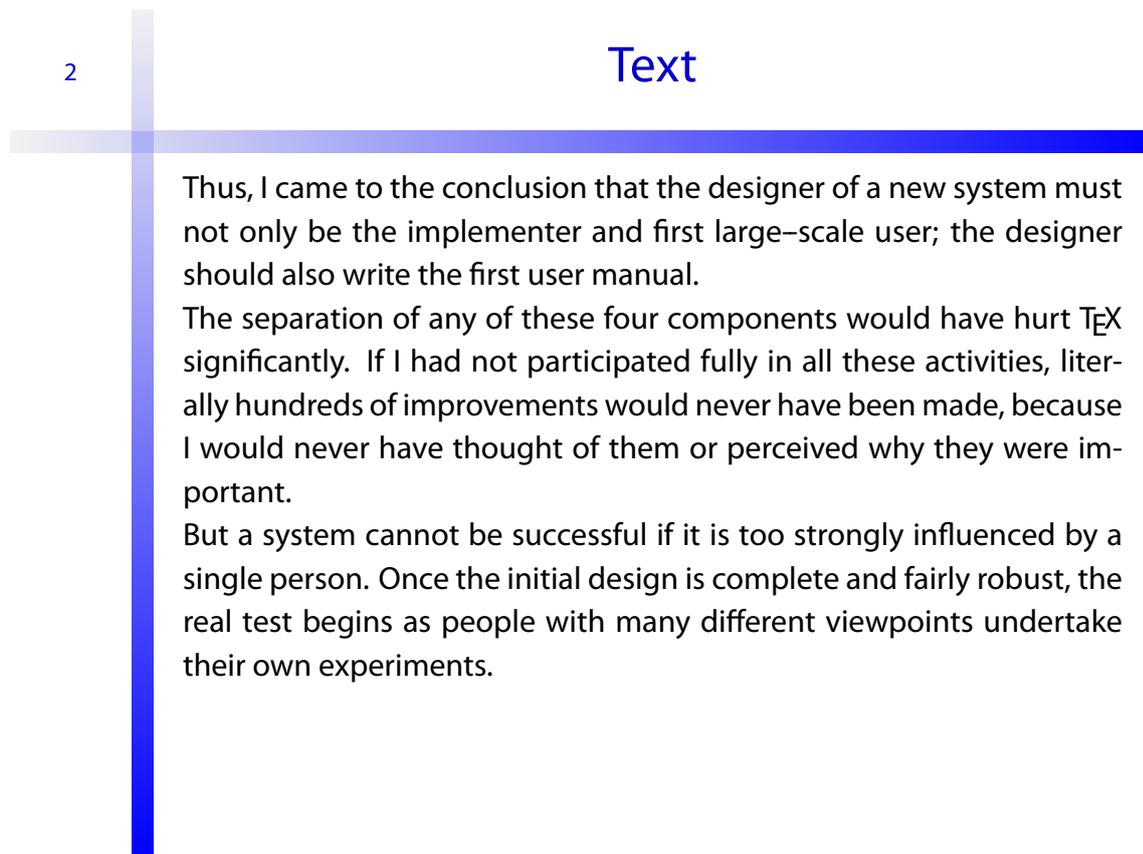


Figure 13 The `lightblue` style

### 5.13 narrowstripe

A variation on the **bluestripe** style, with shaded narrow stripes. This style comes with two color options, which you set with the `stylecolor` key; `stylecolor=red`, `stylecolor=green`, or `stylecolor=blue`.



**Figure 14** The **narrowblue** style with `stylecolor=blue`

## 5.14 quadblue

This style is inspired by the colors and corporate look of my university. It is very sober and offers much space for text and images. There is a rough progress meter built into the blue quadrangles.

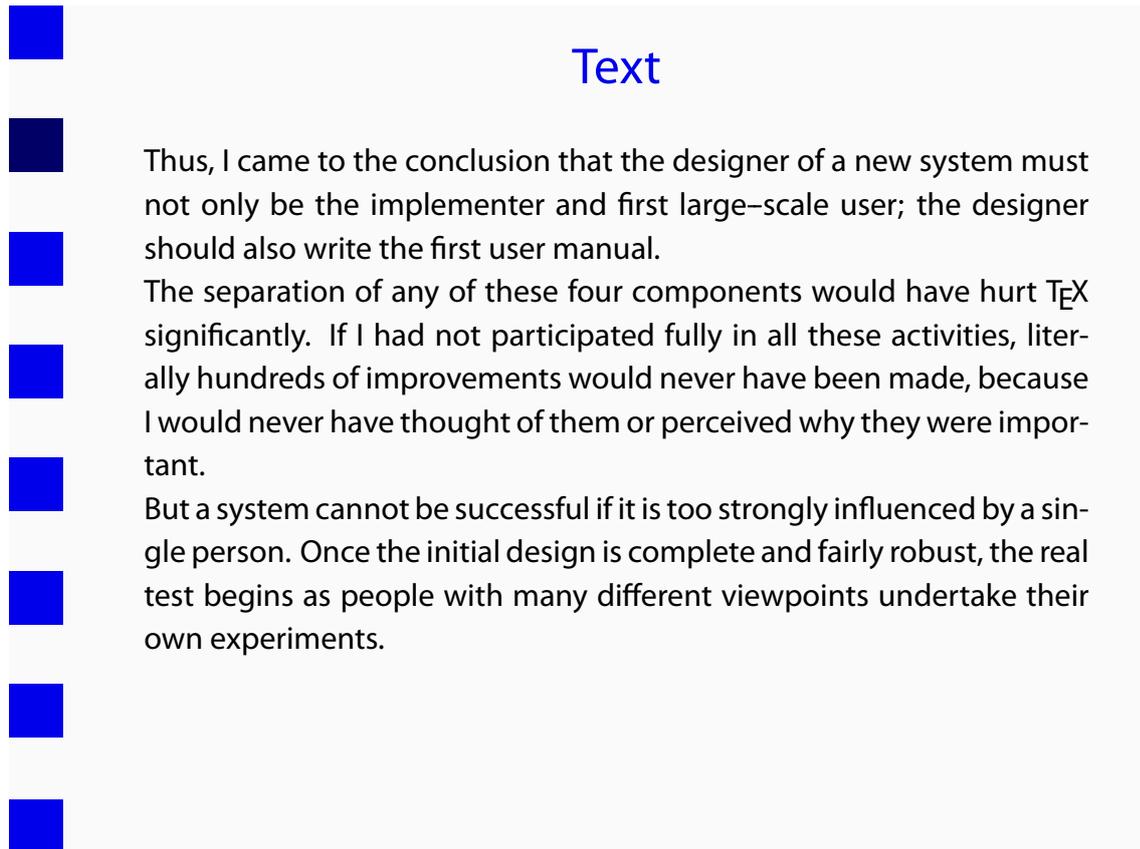


Figure 15 The **quadblue** style

## 5.15 rainbowstripe

A colorful style for daring presenters. The black line which marks the progress is reminiscent of absorption lines in star spectra, so this style may be apt for astrophysical presentations?

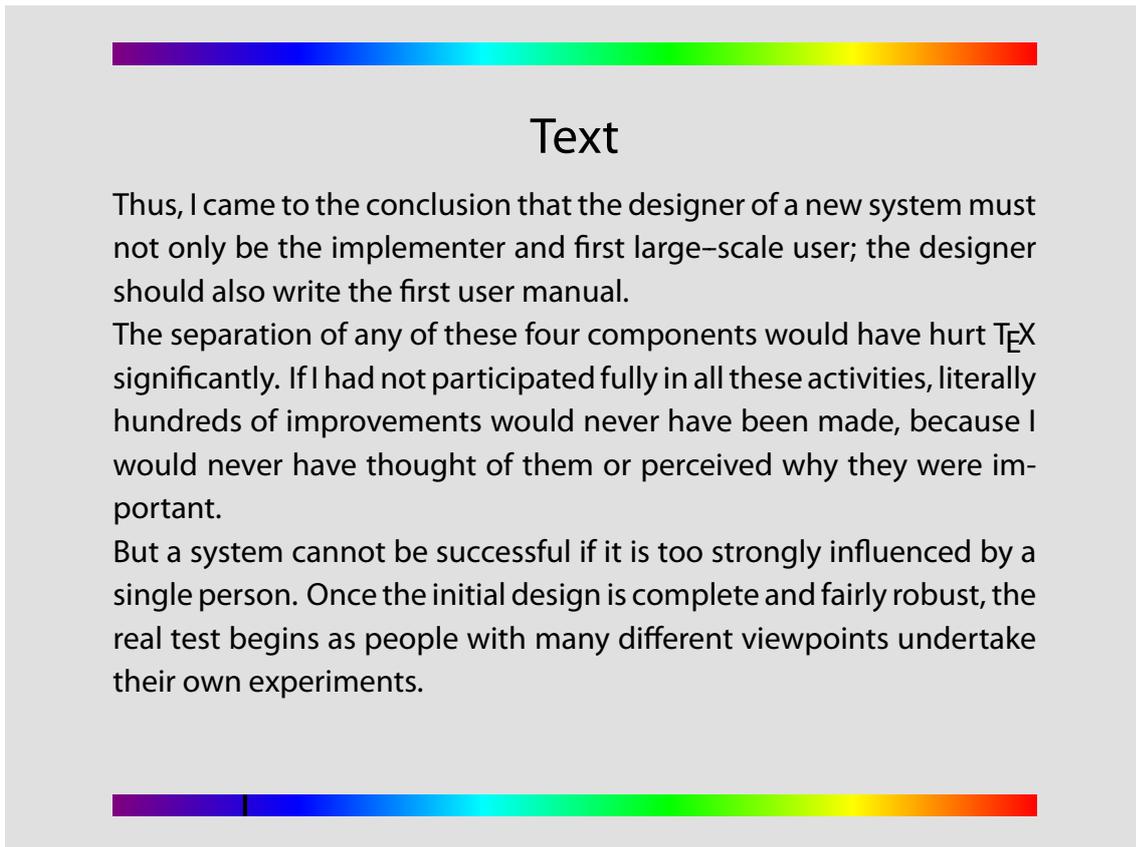


Figure 16 The [rainbowstripe](#) style

## 5.16 [redframe](#)

This style is inspired by the screen version of the Metafun manual. Watch the small gray circles at the bottom!

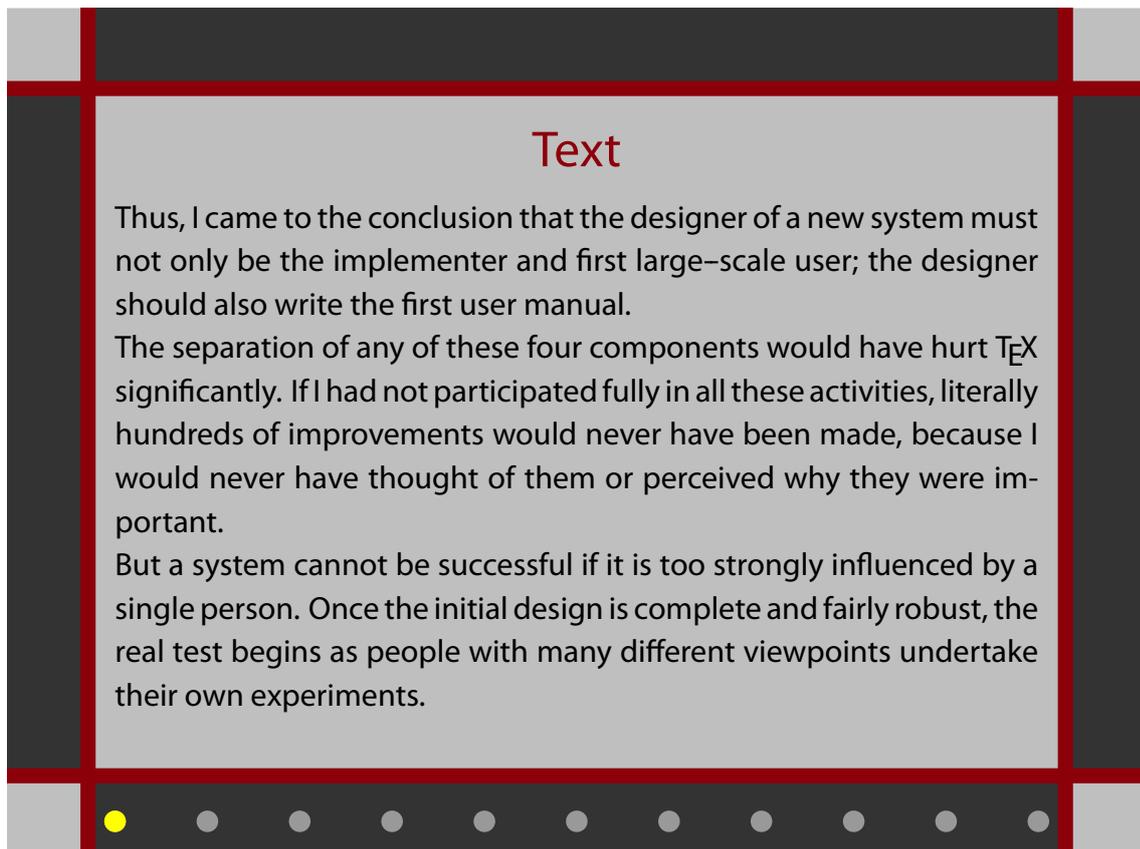
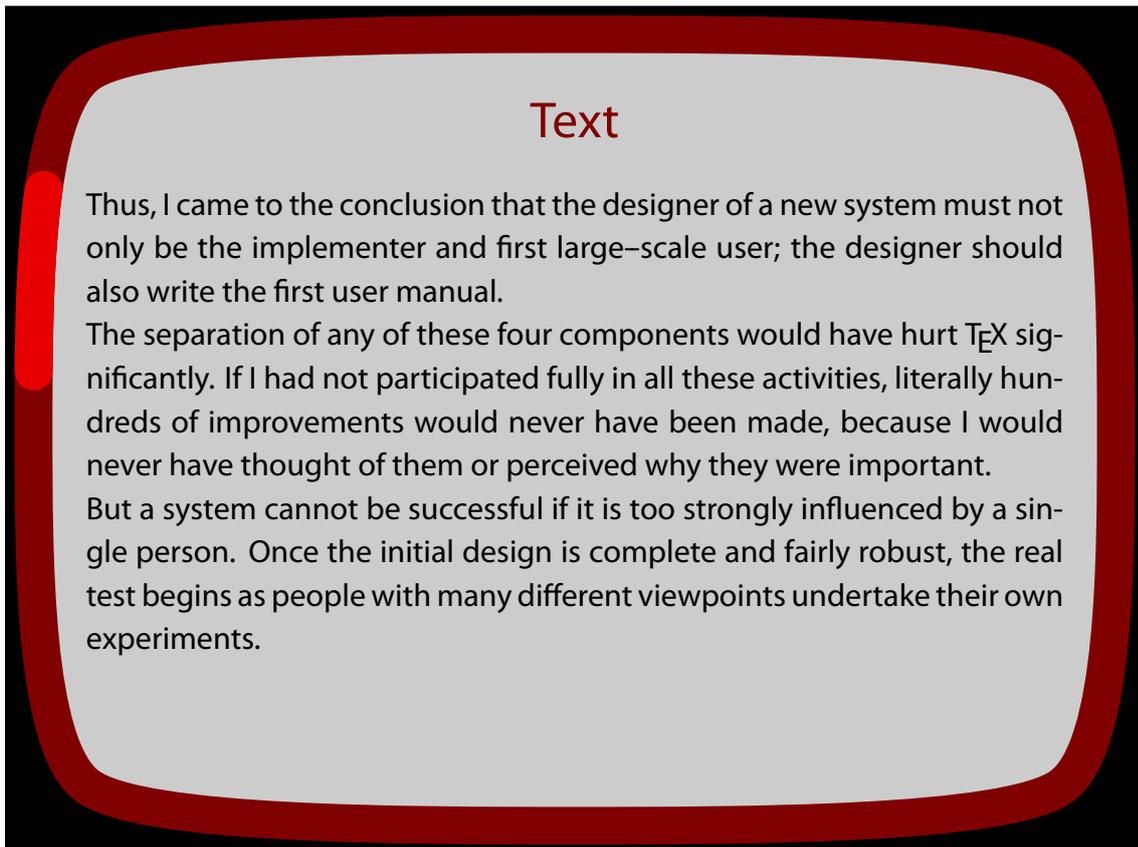


Figure 17 The [redframe](#) style

## 5.17 [superellipse](#)

This style is inspired by Hans’s “funny” style (s-pre-03). The light red stripe marks the progress.



**Figure 18** The [superellipse](#) style

## 5.18 `titleframe`

The interesting feature of this style is the “scratch counter” at the bottom; it is derived from section 7.2 of the Metafun manual.

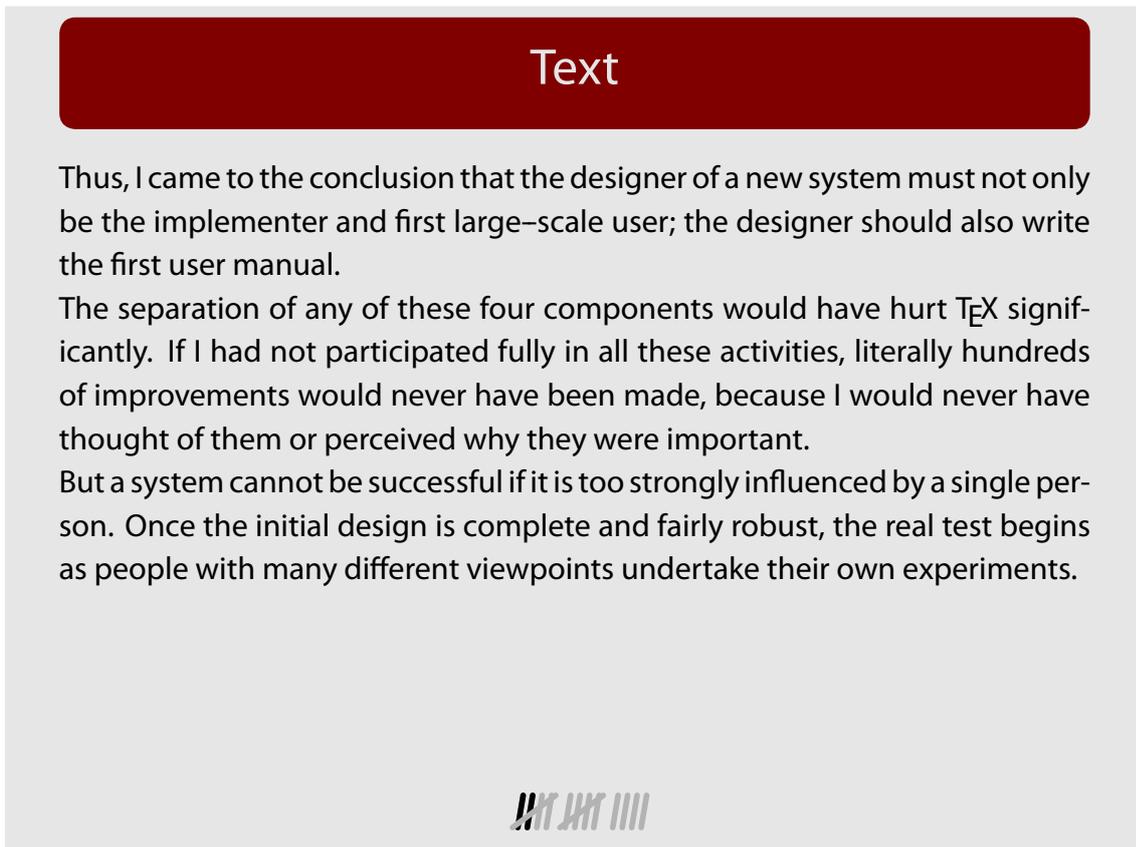


Figure 19 The `titleframe` style

## 5.19 Customization

The style parameter allows easy customization. If you want to develop your own theme, I would suggest copying one of the style files to another name, say `MyStyle.tex`, and modifying it to your heart’s content: you can change the colors or define a different background altogether, think of a different way of displaying titles, different margins, etc. Just be sure to define all the macros that are needed. After producing your own style, again, copy it to a place where T<sub>E</sub>X can find it (such as the same directory where the source of your presentation resides) and instruct the module to use your file:

```
\usemodule[taspresent][style=MyStyle,font=Times,size=17pt]
```

## 6 The font Key

There is a number of predefined fonts which can be selected by setting the `font` key.

<code>LatinModern</code>	typesets in LatinModern Serif
<code>LatinModernSans</code>	typesets in LatinModern Sans
<code>Times</code>	the free clone of TimesNew Roman
<code>Helvetica</code>	the free clone of Helvetica
<code>Pagella</code>	the tex-gyre clone of Palatino; this will only work if you have the tex-gyre fonts installed

In addition, there is a value `User`; this will not set a font but allow you to provide your own settings. If you set your own font, please remember to select the `bodyfont` at `\Normalsize` and to give your setup commands *after* loading the module (or  $\TeX$  will not know what `\Normalsize` means and complain about an “undefined control sequence”). For example, for the samples included here, I have used my own typescript which defines the Adobe MyriadPro font:

```
\usetypscriptfile[type-myriadpro]
\usetypscript[MyriadPro] [texnansi]
\setupbodyfont[MyMyriadPro,ss,\Normalsize]
```

## 7 The size Key

This selects the font size for the main text and defines a corresponding size for titles.

Value	Normalsize	Titlesize
16pt	16pt	25pt
17pt	17pt	27pt
18pt	18pt	28pt
19pt	19pt	30pt
20pt	20pt	30pt
21pt	21pt	30pt

**Figure 20** Text and title sizes

## 8 Macros

The module provides some macros to facilitate the preparation of presentations.

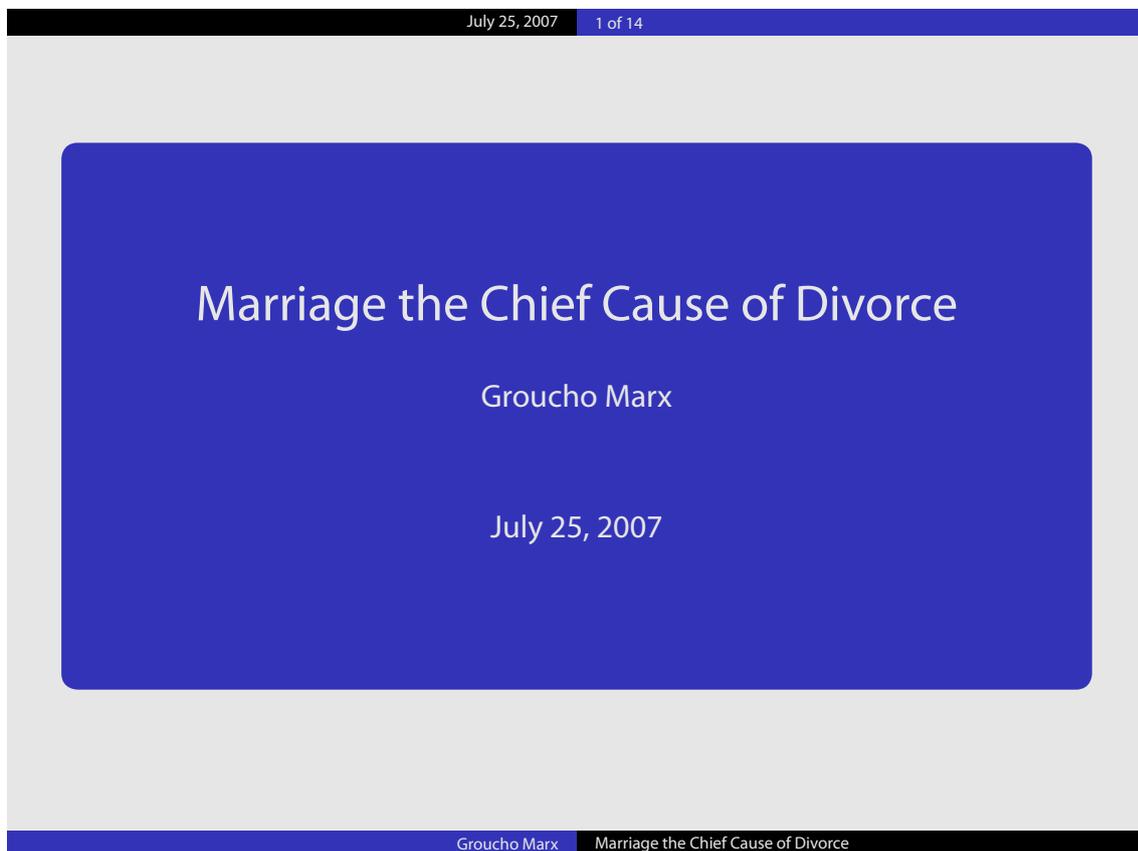
## 8.1 `\setvariables`

Begin your presentation by setting the name of the author(s) and the title with this macro:

```
\setvariables [taspresent]
               [author={Groucho Marx},
               title={Marriage the Chief Cause of Divorce}]
```

## 8.2 `\Maketitle`

This macro will produce a title page with the author and the title of the presentation; the look is of course determined by the style of your presentation.



**Figure 21** A Title Page

## 8.3 `\Slidetitle`

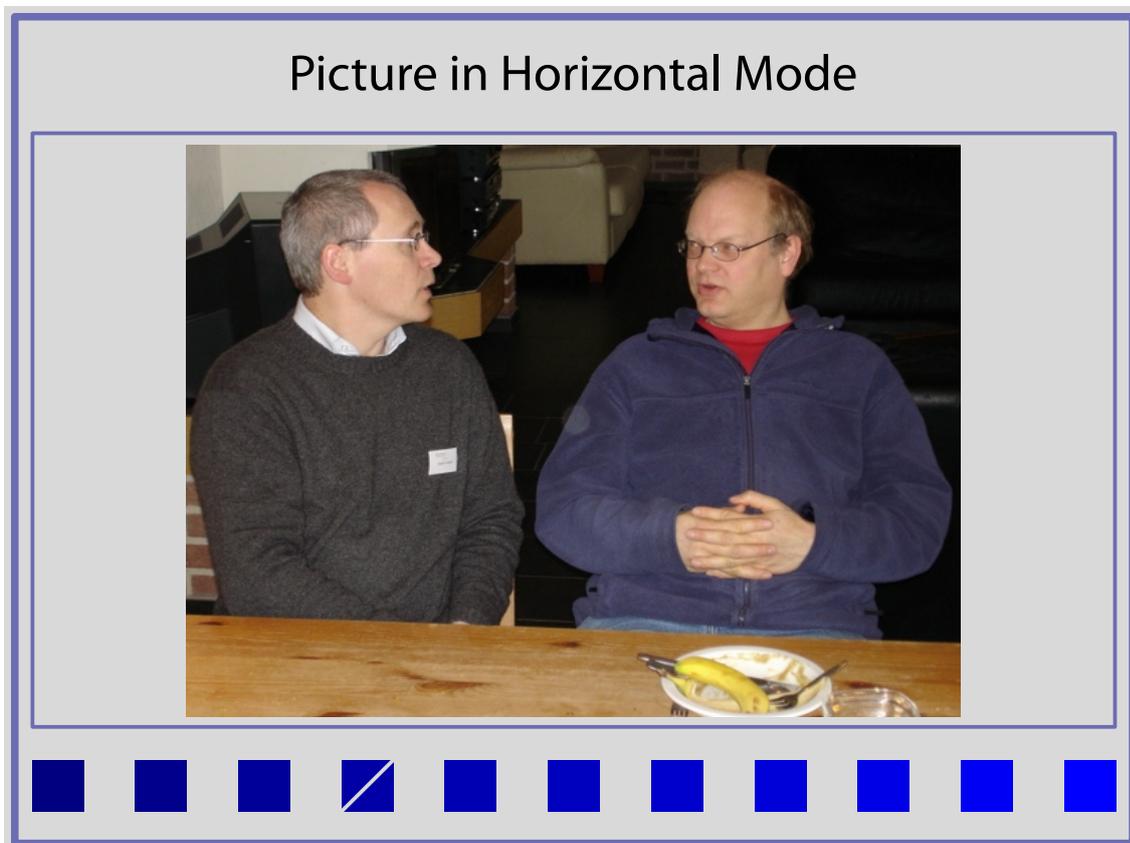
As the name suggests, this macro typesets its argument as the title of the slide. What the title looks like is determined by the selected presentation style.

## 8.4 `\PicHoriz`

This macro facilitates the placement of landscape images. It takes two arguments:

```
\PicHoriz [image] [height=\NormalHeight]
```

The first argument is the name of the image you want to place; the second argument determines the size. If your picture is not too broad, a height of `\NormalHeight` will make it fill up the entire text area. If your picture is too broad, you should set `width=\textwidth`.



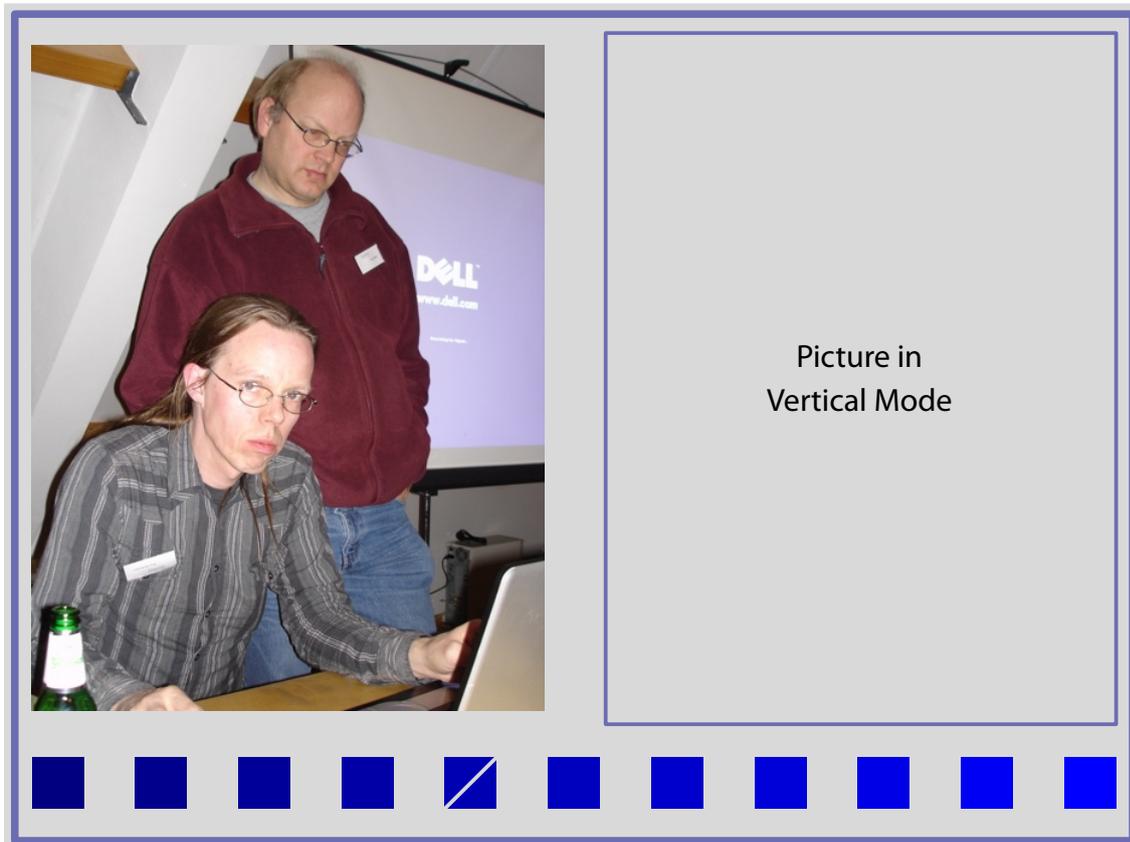
**Figure 22** Placement of a horizontal picture

## 8.5 `\PicVert`

This macro facilitates the placement of portrait images. It takes three arguments:

```
\PicVert [image] [width=\NormalWidth] {Text \ to go \ with the picture}
```

Again, the first argument is the name of the image you want to place; the second argument determines the size. If your picture is not too high, a width of `\NormalWidth` will make it fill up the entire left half of the text area. If your picture is too high, you should set `height=\textheight`. The third argument is the text that will be placed opposite the picture.



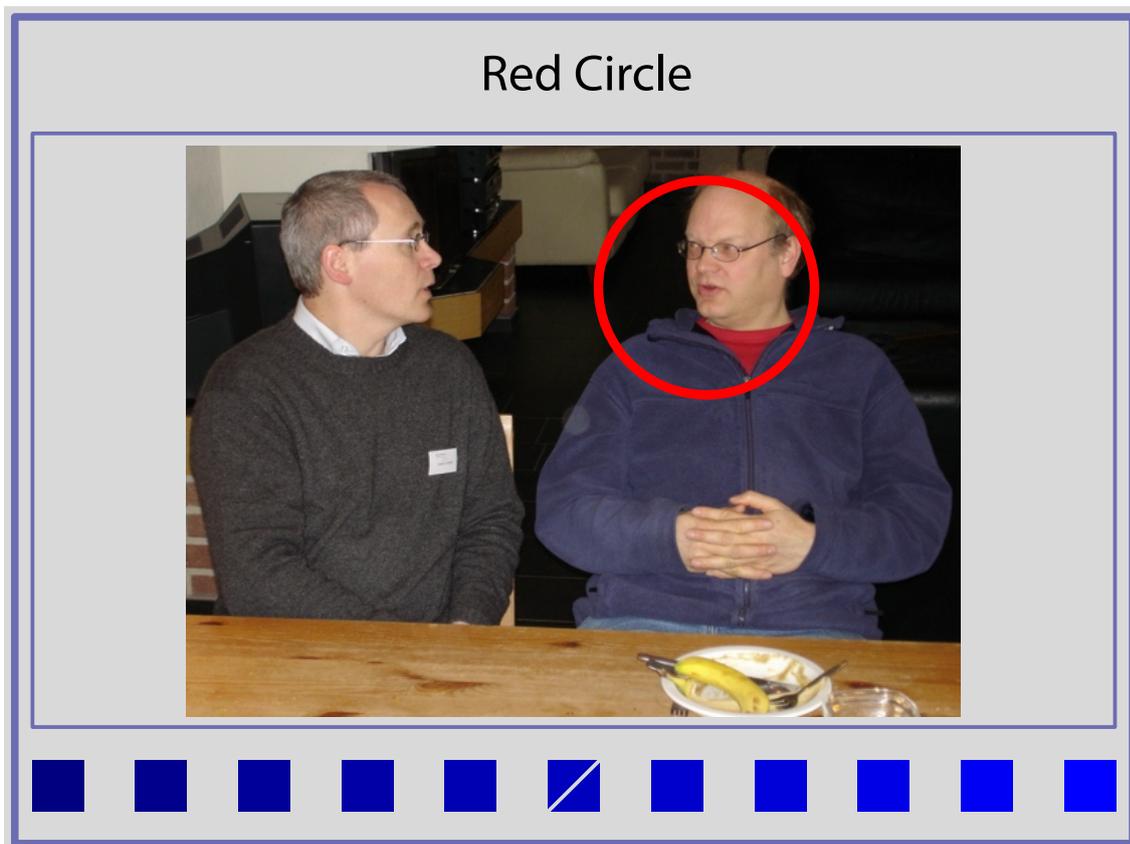
**Figure 23** Placement of a vertical picture

## 8.6 `\CircHoriz`

This command works exactly like `\PicHoriz`, but takes an additional (third) argument. It places a red circle on top of the picture; the placement and size of this circle is determined by this third argument:

```
\CircHoriz[scale=40,x=120,y=80][image][height=\NormalHeight]
```

The `scale` key sets the diameter of the circle (in mm), `x` and `y` set horizontal and vertical position. You will probably have to fiddle with these keys to get the circle exactly where you want it.



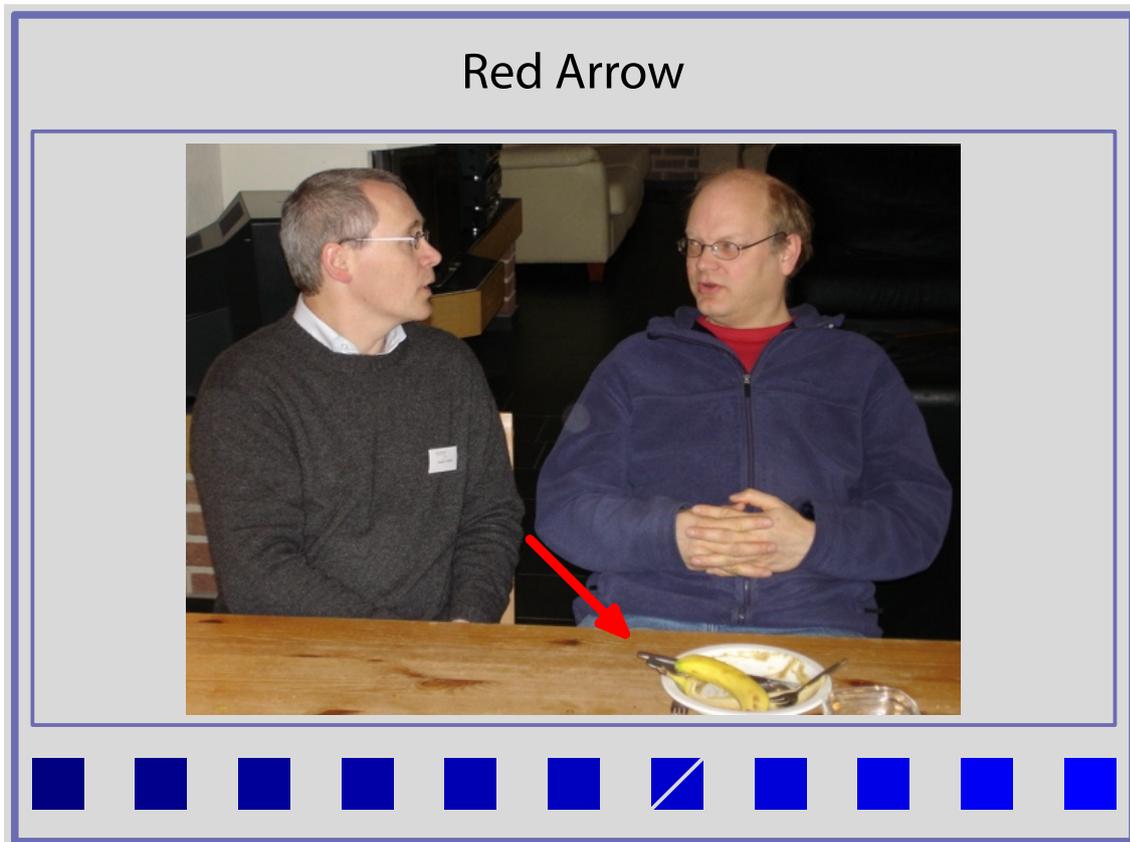
**Figure 24** A picture with a red circle

## 8.7 `\ArrowHoriz`

This command works exactly like `\PicHoriz`, but takes an additional (third) argument. It places a red arrow on top of the picture; the direction and size of this arrow is determined by this third argument:

```
\CircHoriz [direction=135,x=120,y=80] [image] [height=\NormalHeight]
```

The `direction` key sets the direction into which the arrowhead points, `x` and `y` set its horizontal and vertical position. You will probably have to fiddle with these keys to get the arrow exactly where you want it.



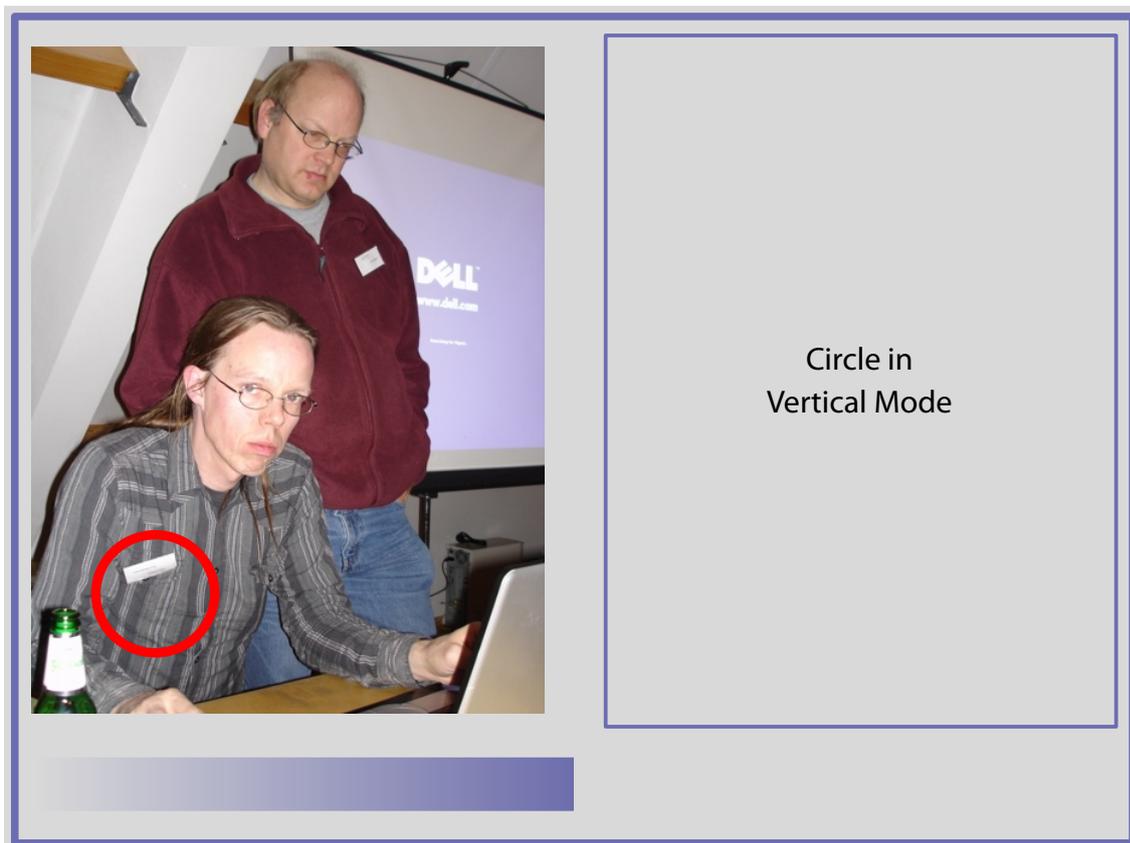
**Figure 25** A picture with a red arrow

## 8.8 `\CircVert` and `\ArrowVert`

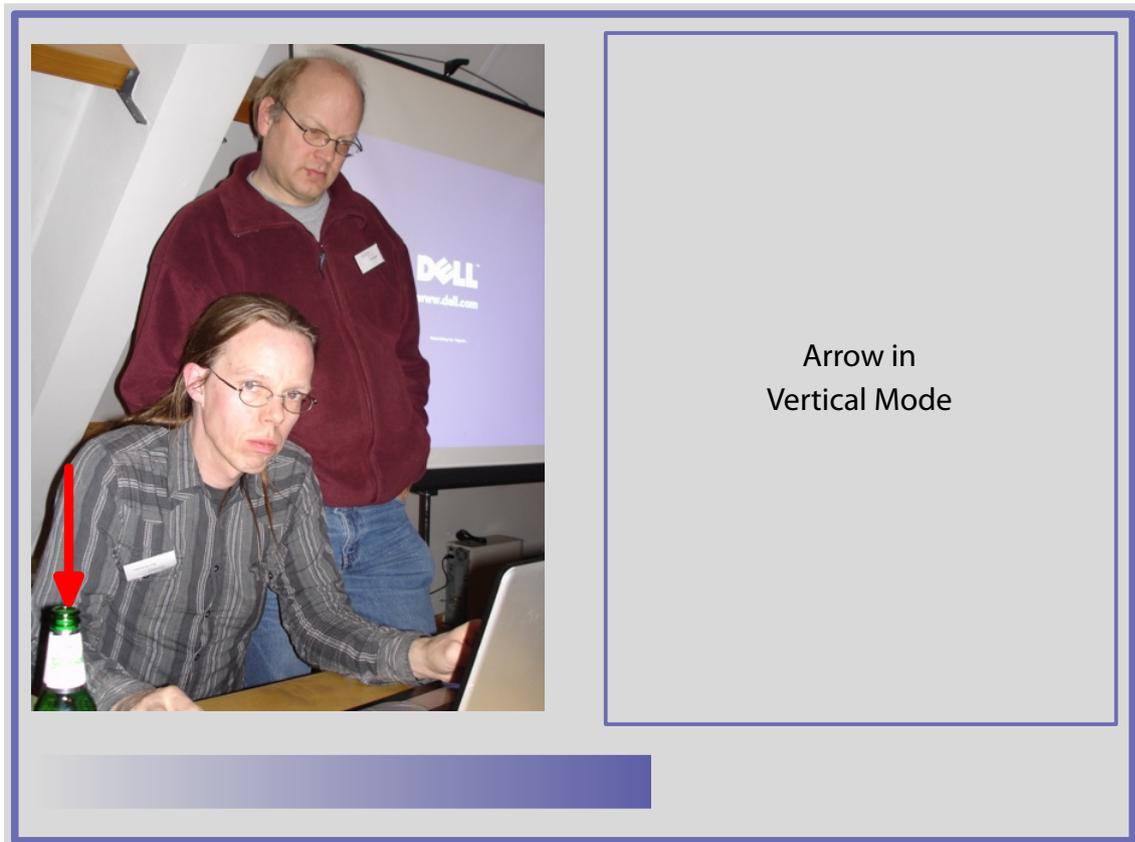
Of course, there are also circles and arrows for “vertical” pictures; again, the first argument is the position of the circle/arrow:

```
\CircVert [scale=22,x=23,y=25]%
[vert]%
[width=\NormalWidth]%
{Circle in \\ Vertical Mode}
```

```
\ArrowVert [direction=90,x=7,y=23]%
[vert]%
[width=\NormalWidth]%
{Arrow in \\ Vertical Mode}
```



**Figure 26** Vertical picture with red circle



**Figure 27** Vertical picture with red arrow

## 8.9 Background

Some of the styles provide up to three backgrounds: for titles, for slides with vertical image, and for normal slides with text or horizontal images. Switching the backgrounds also adjusts parameters like margins or headers, where this is necessary. There are three commands for setting the background for title slides, “horizontal” slides and “vertical” slides respectively:

```
\titback  
\lecback  
\picback
```

## 9 Handouts

The easiest way to make handouts from your slides is post-processing the pdf-file of your presentation. If you want to make the handouts from the slides such as

they appear, just run this command in the directory where your slides are located (of course, put the name of your own presentation where you see `demo.pdf` in the example; the entire command has to go in one long line):

```
texmfstart texexec --pdfcombine --combination='2*3' --nobanner  
--result=handout demo.pdf
```

This way, you will get a handout with six slides typeset in two columns per page. If you prefer to have the handouts without the colored background, typeset your presentation and leave the *style* key unset; that way, you will get a default version without backgrounds and fancy frames.