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 seven 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 LATEX `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 TEX 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`. On many TEX-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 Setting up the Module

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

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

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

4 The `style` Key

There are nine options for the `style` key:

4.1 blueframe

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. There is a "progress meter" at the bottom, a shaded blue area which will grow with each slide.

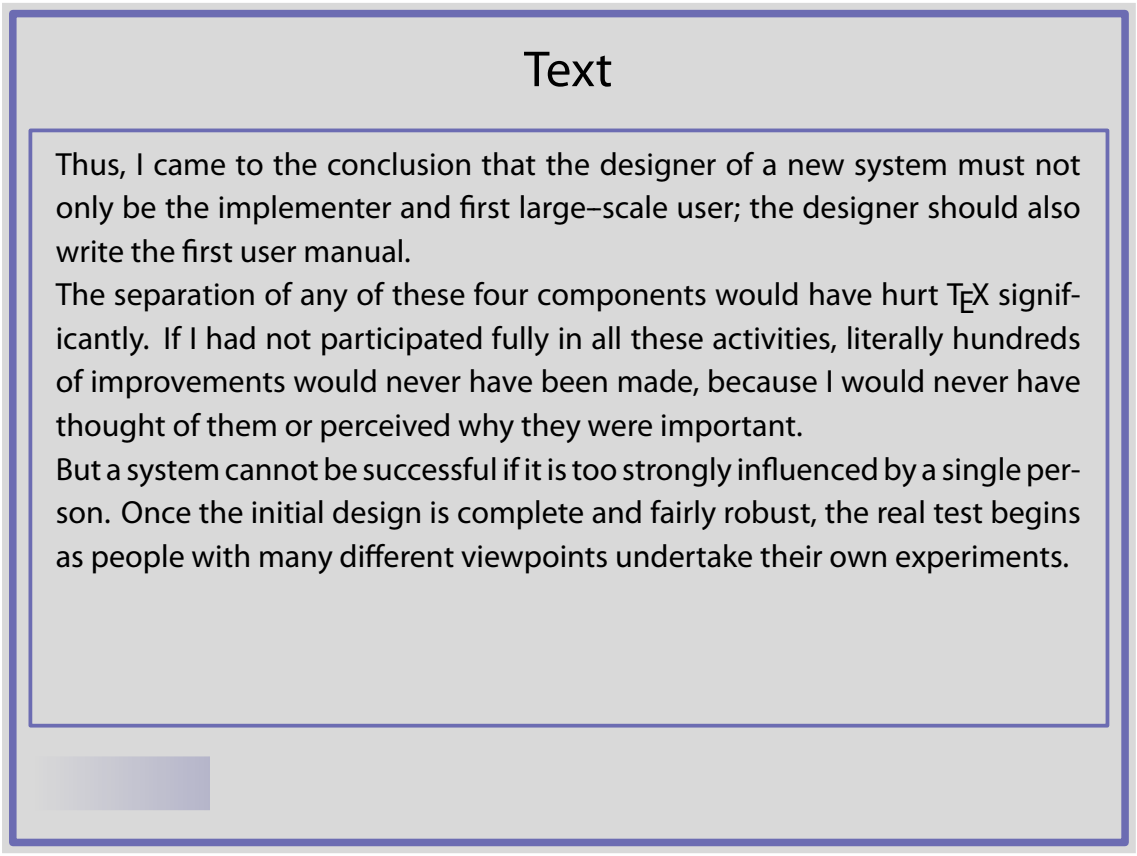

Figure 1 The blueframe style

4.2 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`).

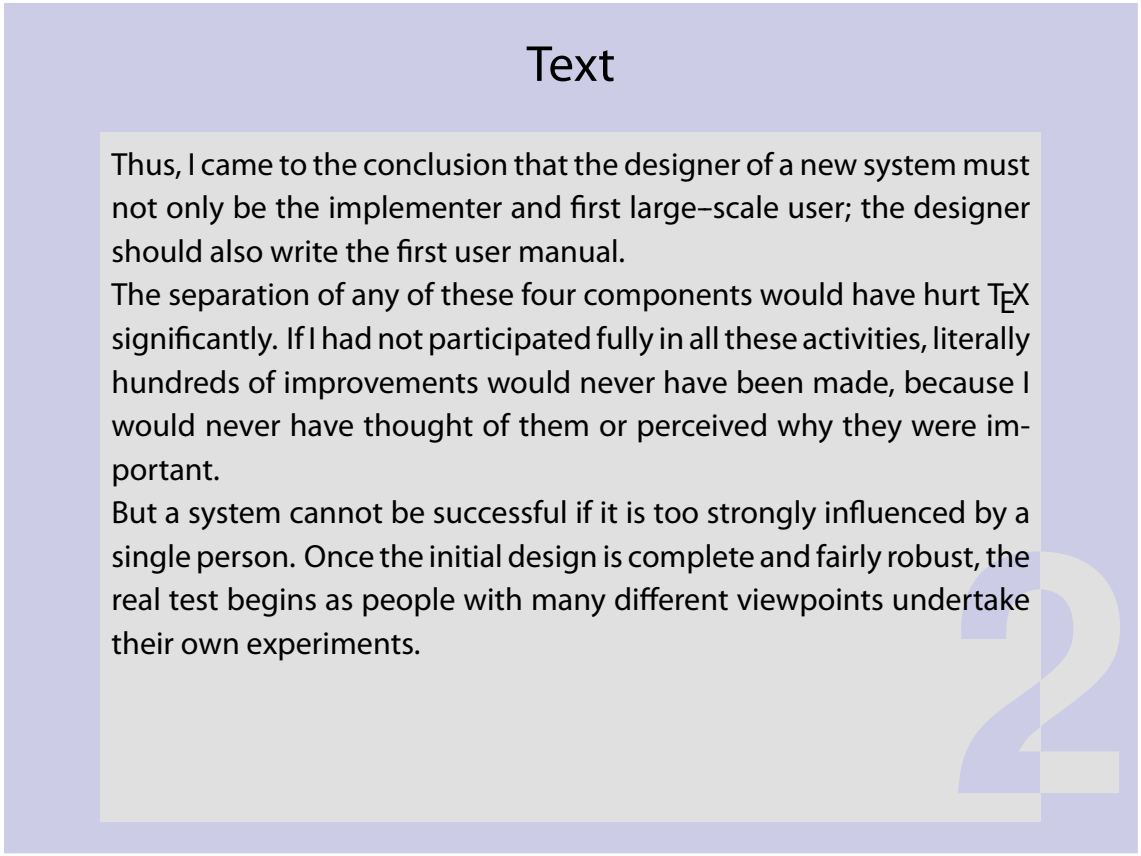

Figure 2 The bluegray style

4.3 blueshade

The only ornament to this style is the dark blue, shaded background. It uses ConTEXt's `interactionbar` mechanism to show the progress of the presentation. It provides much space for text.

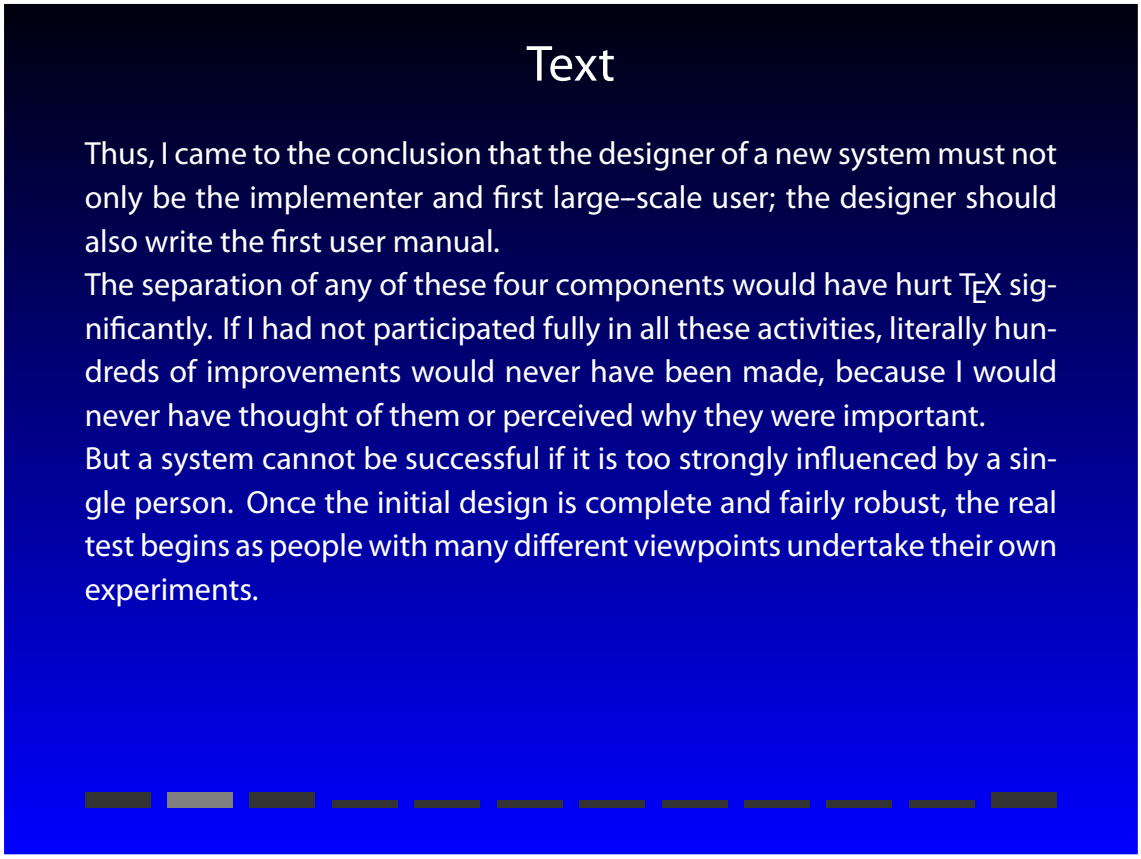

Figure 3 The blueshade style

4.4 bluestripe

This theme is inspired by the "Berkeley" style of the LaTeX `beamer` package. Apart from the blue sidebars, it has no ornaments.

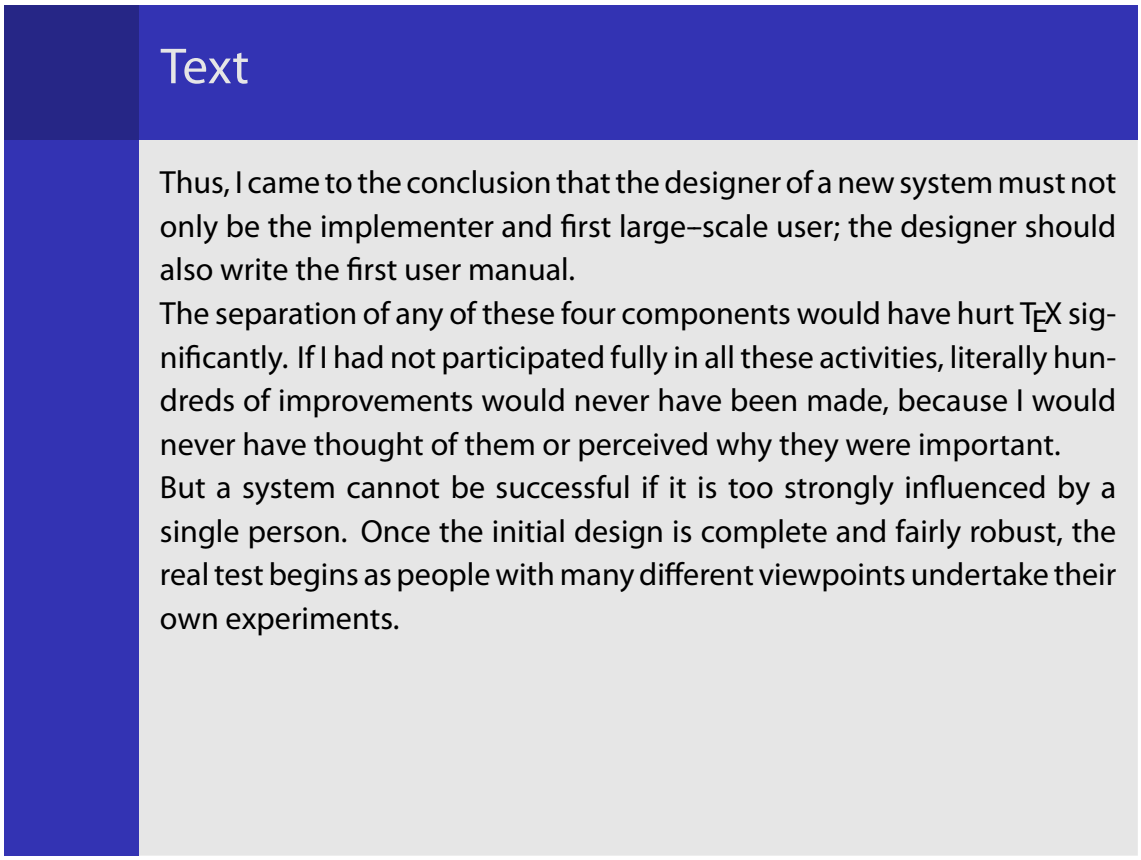

Figure 4 The bluestripe style

4.5 doubleshade

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

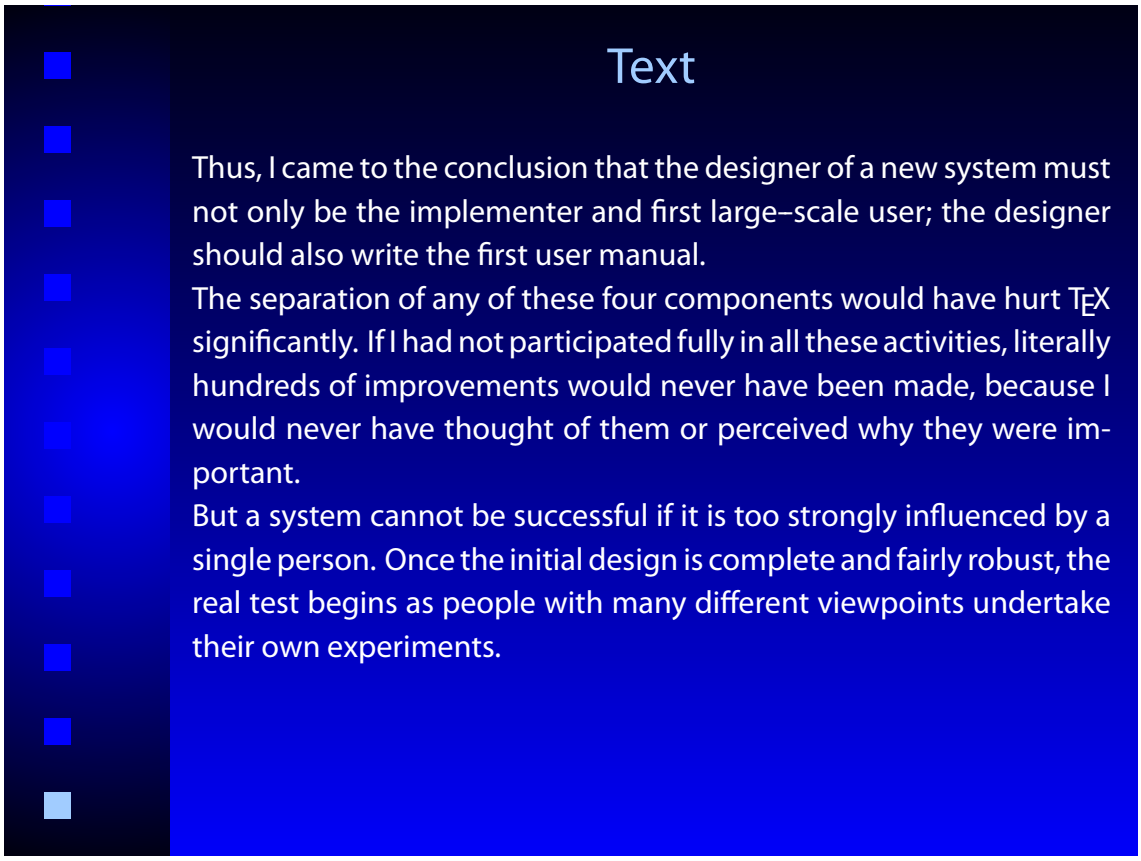

Figure 5 The doubleshade style

4.6 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.

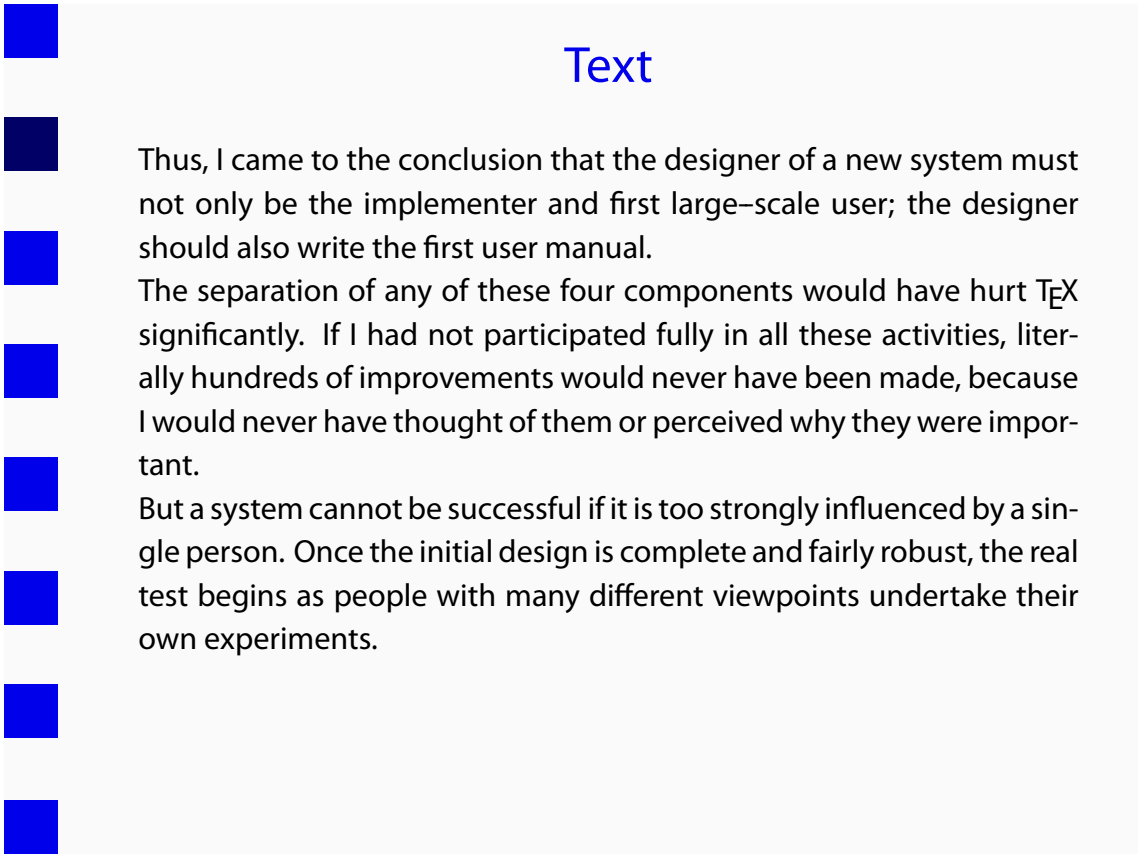

Figure 6 The quadblue style

4.7 redframe

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

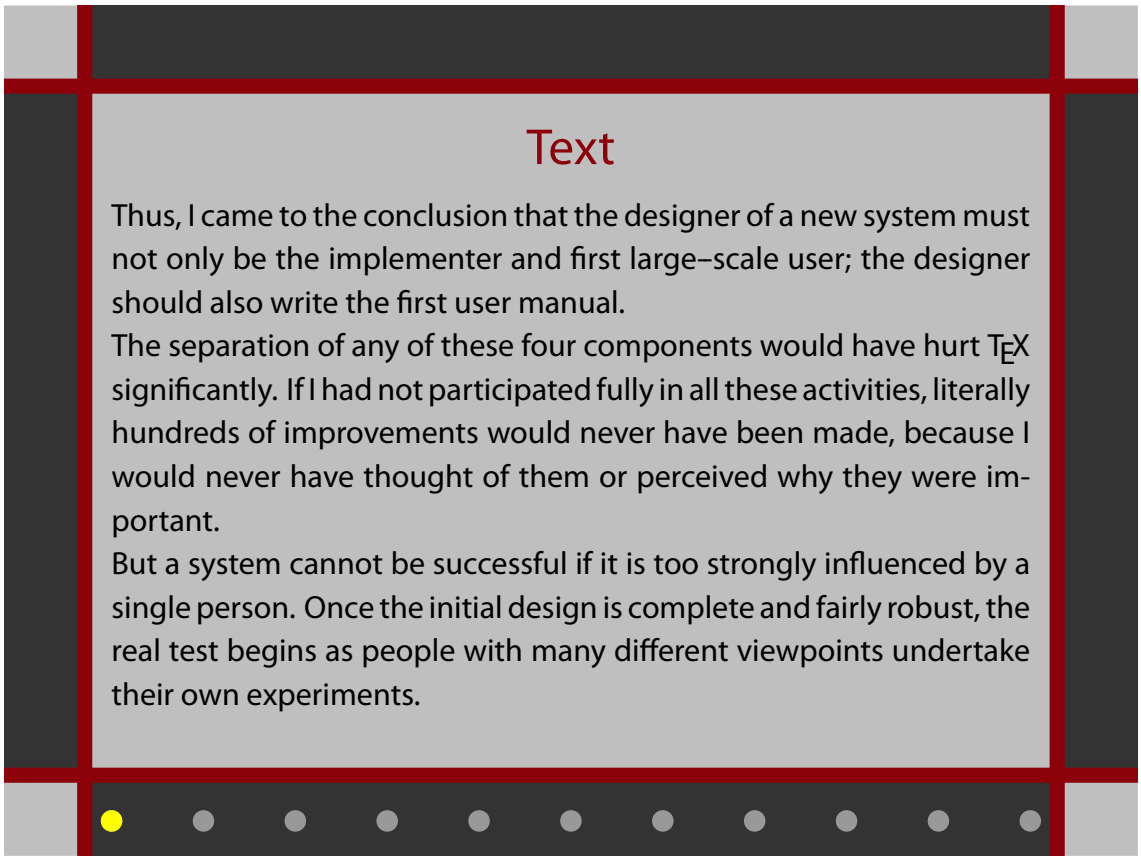

Figure 7 The redframe style

4.8 squareframe

This style is identical to the blueframe style, but it has blue squares at the bottom; they also measure the presentation's progress.

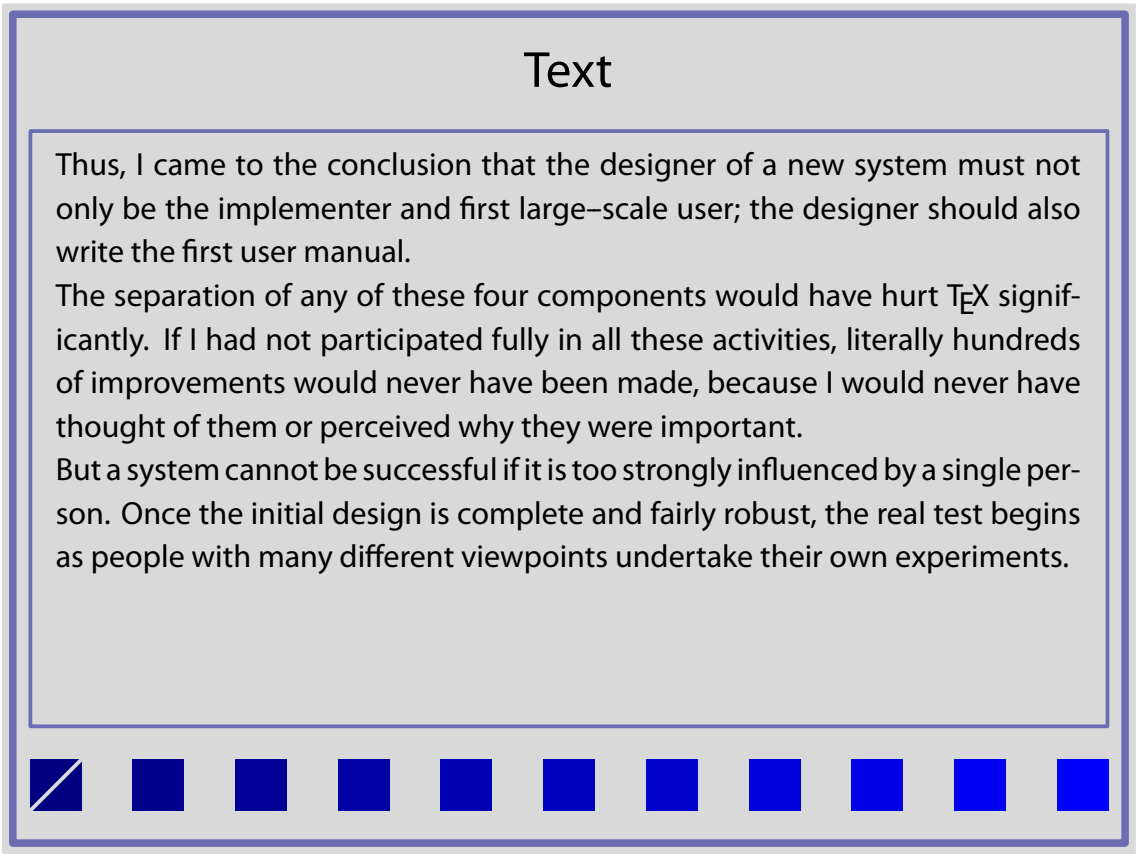

Figure 8 The squareframe style

4.9 Customization

The style parameter allows easy customization. If you want to do that, I would suggest copying one of the style files to another name, say `MyStyle.tex`, and modifying it to your heart's content: define a different background, 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 TEX can find it and instruct the module to use your file:

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

5 The `font` Key

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

`LatinModern`	typesets in LatinModern Serif
`LatinModernSans`	typesets in LatinModern Sans
`Times`	the free clone of TimesNew Roman
`Helvetica`	the free clone of Helvetica
`Pagella`	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 "unknown command"). For example, for the samples included here, I have used my own typescript which defines the Adobe MyriadPro font:

```
\usetypescriptfile[type-myriadpro]
\usetypescript[MyriadPro] [texnansi]
\setupbodyfont[MyMyriadPro,ss,\Normalsize]
```

6 The `size` Key

This selects the Normalsize for the main text and defines a corresponding Titlesize for titles.

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

Figure 9 Text and title sizes

7 Macros

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

7.1 \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.

7.2 \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.

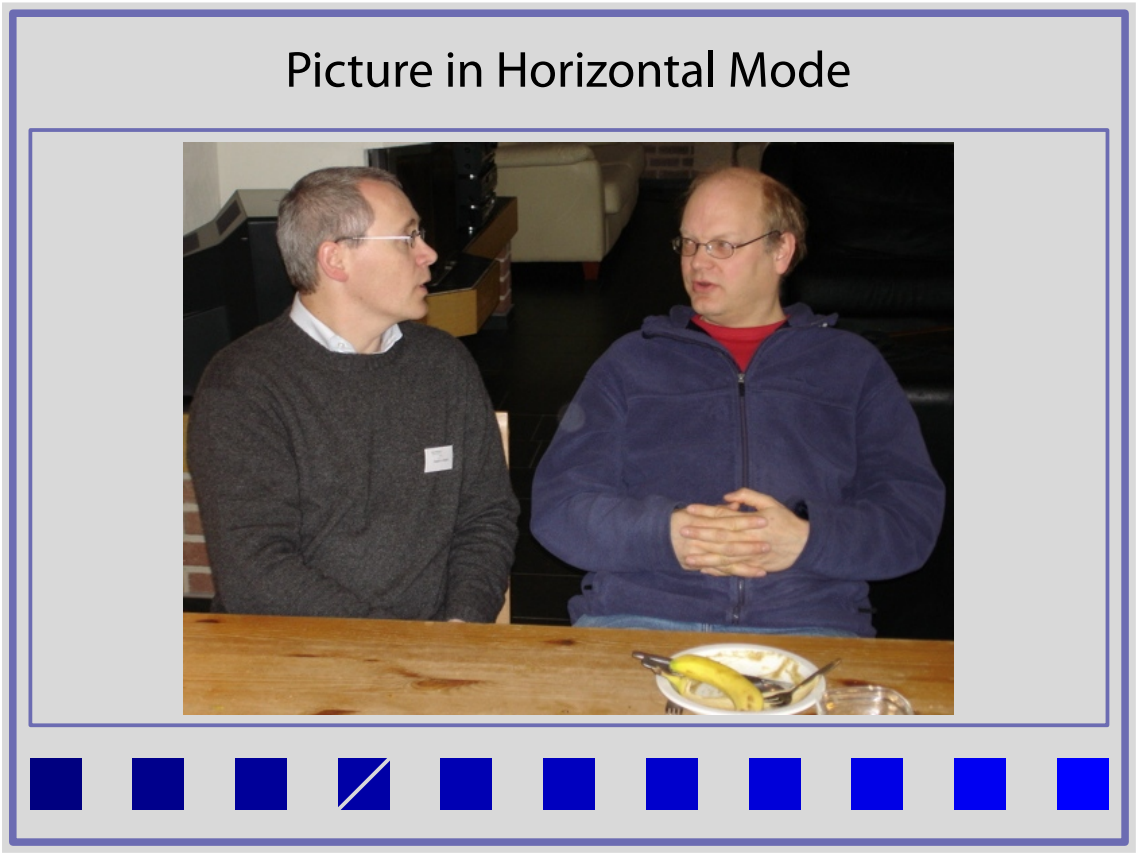

Figure 10 Placement of a horizontal picture

7.3 \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 hight, you should set height=\textheight. The third argument is the text that will be placed opposite the picture.

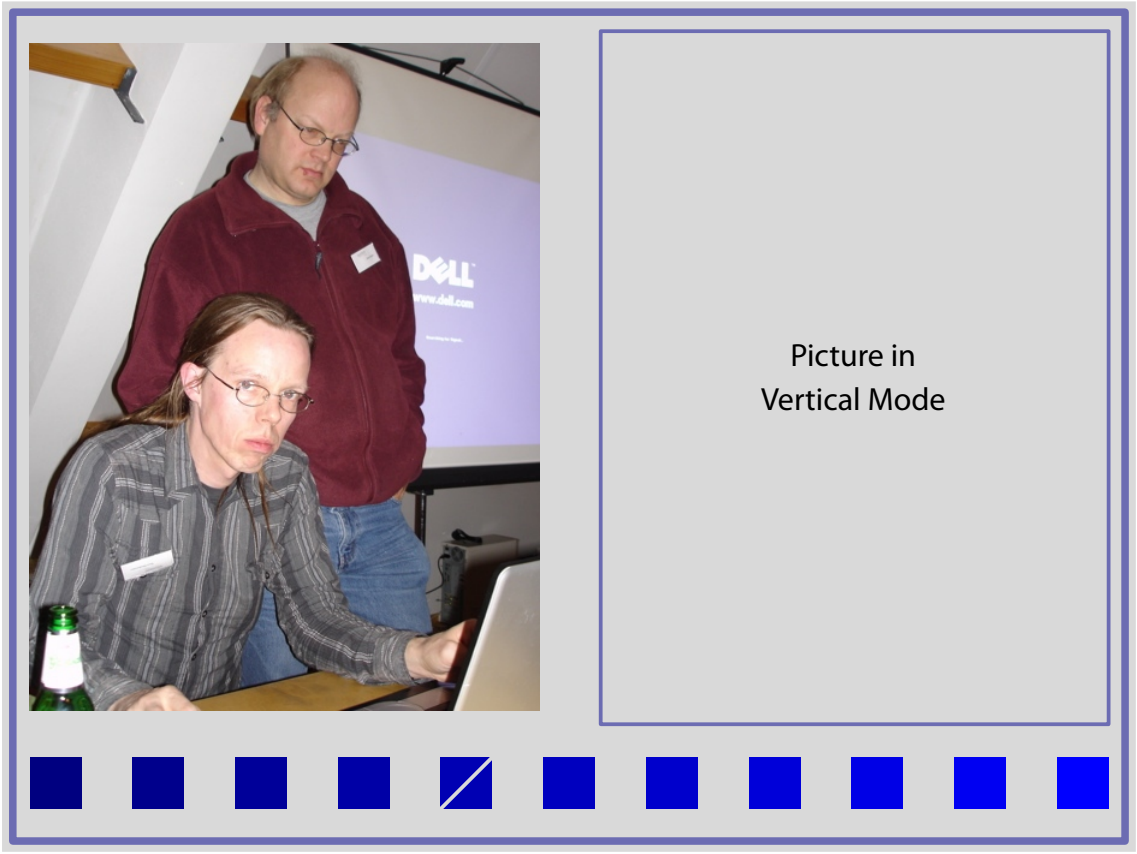

Figure 11 Placement of a vertical picture

7.4 \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.

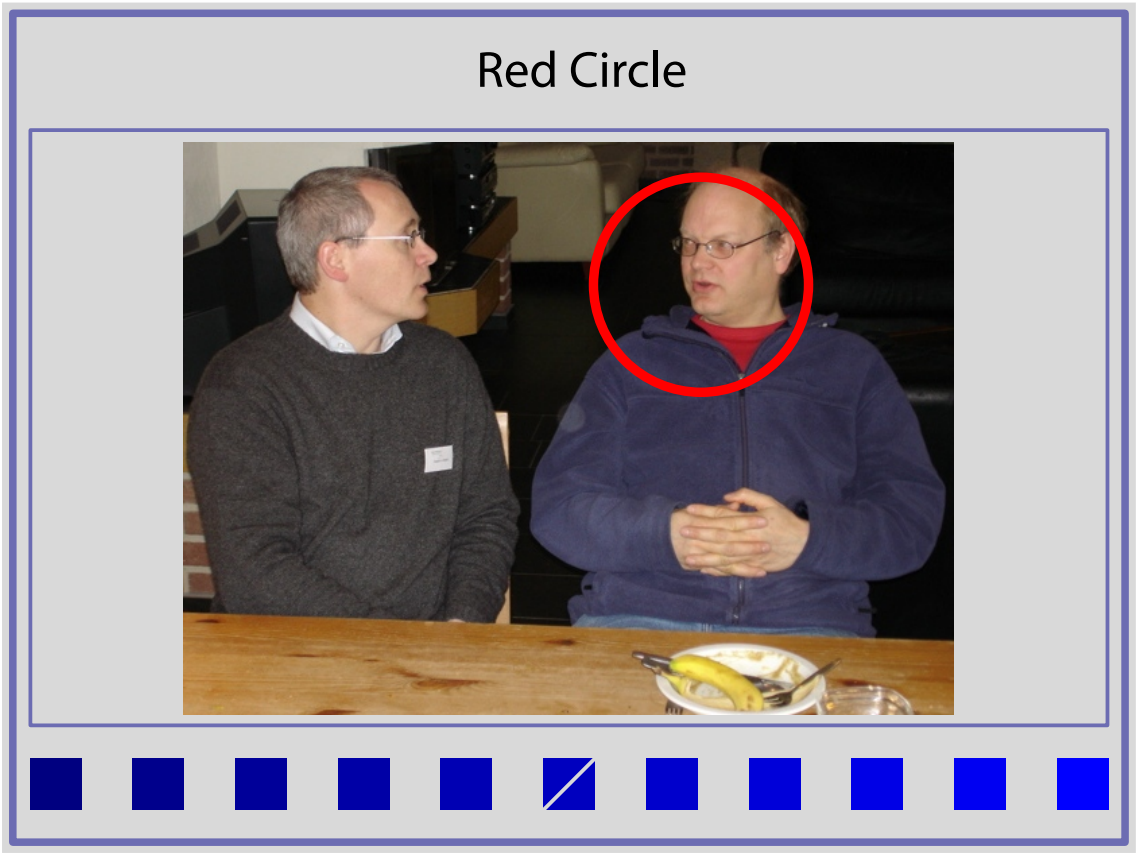

Figure 12 A picture with a red circle

7.5 \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 circle exactly where you want it.

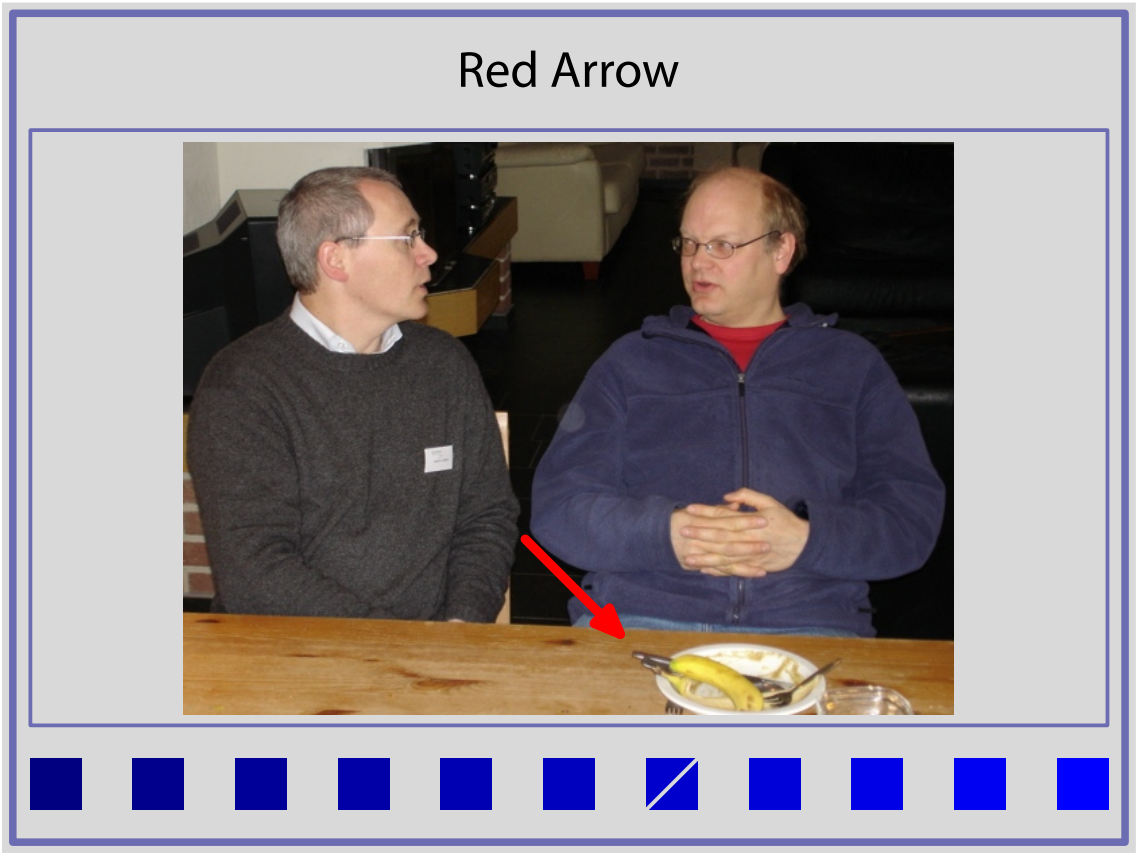

Figure 13 A picture with a red arrow

7.6 \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}
```

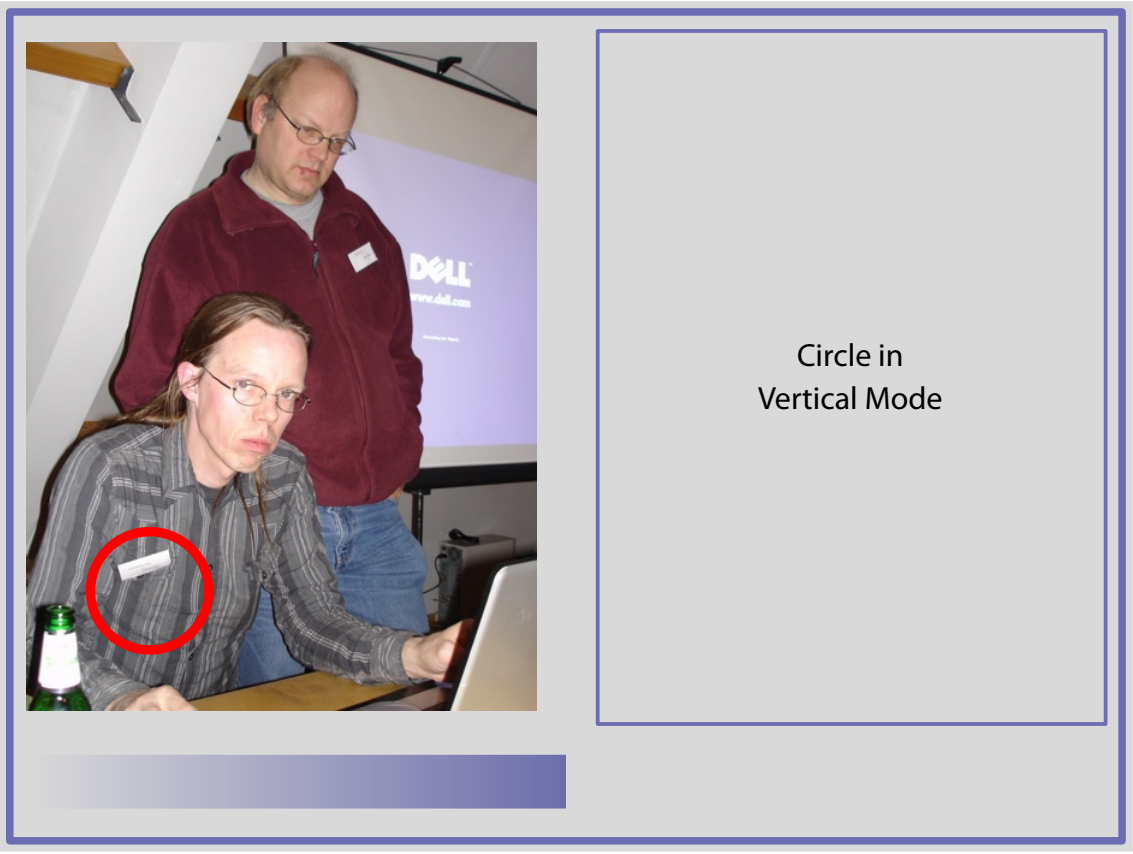

Figure 14 Vertical picture with red circle

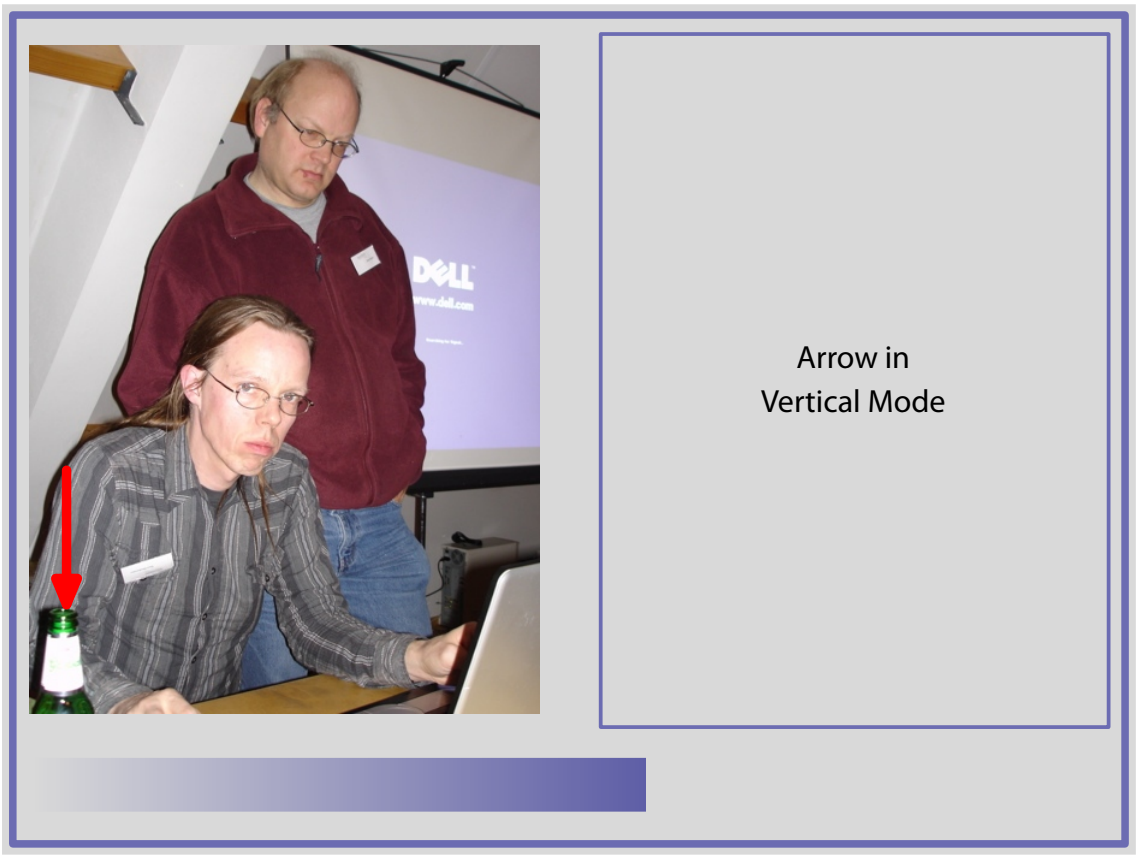

Figure 15 Vertical picture with red arrow

7.7 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
```