



ROYAL INSTITUTE
OF TECHNOLOGY

Can free software replace proprietary software for graphic production?

Investigating to which extent
free software can be used
for book and magazine production.

2013-04-25

Staffan Melin, D89, staffan.melin@oscillator.se

Master's thesis in Media Technology,
School of Computer Science and Communication
at the Royal Institute of Technology, Stockholm, Sweden

This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

Abstract

Free and open source software is widely used. At the same time there are several areas where it is not. One of these is graphic production where the applications from Adobe - **Indesign, Photoshop** and **Illustrator** - dominates.

In this thesis I start by describing a workflow for graphic production. Next I research the field of free software and put together a set of tools that fit into this workflow. The choices are made on the basis of functionality and how they work together.

I then go on to apply these tools to two real world scenarios: production of a book and a magazine.

The results show that free software can be used for graphic production without any loss of quality and only minor problems compared to the proprietary tools. A look into future development shows that the bulk of these problems are being taken care of by the open source community.

Keywords: free software, open source, proprietary software, graphic production, graphic design, FLOSS, FOSS, layout, libre, Scribus, Inkscape, GIMP, GNOME Color Manager.

Sammanfattning

Fri och open source programvara används på många områden. Samtidigt finns det många områden där det inte används. Ett av dessa är grafisk produktion där programmen från Adobe - **Indesign, Photoshop** och **Illustrator** - dominerar.

I denna uppsats börjar jag med att beskriva ett arbetsflöde för grafisk produktion. Genom efterforskningar av fri programvara sätter jag sedan ihop en verktygslåda som passar för detta arbetsflöde. Valen görs baserat på funktionalitet och hur de fungerar tillsammans.

Därefter tillämpar jag dessa verktyg på två verkliga fall: produktion av en bok och en tidskrift.

Resultaten visar att fri programvara kan användas för grafisk produktion utan någon kvalitetsförlust och med endast mindre problem jämfört med proprietära verktyg. En blick in i framtiden visar att majoriteten av dessa problem håller på att åtgärdas av open source-rörelsen.

Nyckelord: fri programvara, open source, proprietär programvara, grafisk produktion, grafisk design, FLOSS, FOSS, layout, libre, Scribus, Inkscape, GIMP, GNOME Color Manager.

Acknowledgement

Thanks to Thomas Melin for valuable thoughts and advice.

Table of contents

| | |
|--|----|
| 1. Introduction..... | 1 |
| 1.1. Objective..... | 1 |
| 1.2. Research questions..... | 1 |
| 1.3. Delimitations..... | 1 |
| 1.4. Intended audience..... | 1 |
| 1.5. Definitions used..... | 2 |
| 2. Method..... | 3 |
| 2.1. Reliability..... | 3 |
| 2.2. Repeatability..... | 3 |
| 2.3. Validity..... | 3 |
| 3. Theory..... | 4 |
| 3.1. Workflow..... | 4 |
| 3.1.1. Step 1: Manuscript preparation..... | 5 |
| 3.1.2. Step 2: Image preparation..... | 6 |
| 3.1.3. Step 3: Template construction..... | 7 |
| 3.1.4. Step 4: Page layout..... | 8 |
| 3.1.5. Step 5: Cover layout..... | 9 |
| 3.1.6. Step 6: Proofing..... | 9 |
| 3.1.7. Step 7: Print preparation..... | 9 |
| 3.2. Color management..... | 11 |
| 3.2.1. Color management crash course..... | 11 |
| 3.2.2. Create a color managed workflow..... | 12 |
| 4. Software..... | 13 |
| 4.1. Workflow..... | 13 |
| 4.1.1. Step 1: Manuscript preparation..... | 13 |
| 4.1.2. Step 2: Image preparation..... | 14 |
| 4.1.3. Step 3–5: Layout..... | 14 |
| 4.1.4. Step 6: Proofing..... | 14 |
| 4.1.5. Step 7: Print preparation..... | 15 |
| 4.2. Color management..... | 16 |
| 4.2.1. Step 1: Install and enable a CMS..... | 16 |
| 4.2.2. Step 2: Calibrate the screen..... | 17 |
| 4.2.3. Step 3: Configure GIMP..... | 17 |
| 4.2.4. Step 4: Configure Inkscape..... | 19 |
| 4.2.5. Step 5: Configure Scribus..... | 19 |
| 4.3. Other tools and resources..... | 20 |
| 4.3.1. Font management..... | 20 |
| 4.3.2. Font creation..... | 20 |
| 4.3.3. Fonts..... | 20 |
| 4.4. Operating system..... | 20 |
| 4.4.1. Recommendation..... | 20 |
| 5. Case study: Paperback production..... | 21 |
| 5.1. Workflow..... | 21 |
| 5.1.1. Step 0. Color management..... | 21 |
| 5.1.2. Step 1: Manuscript preparation..... | 24 |
| 5.1.3. Step 2: Image preparation..... | 26 |
| 5.1.4. Step 3: Template construction..... | 26 |
| 5.1.5. Step 4: Page layout..... | 28 |
| 5.1.6. Step 5: Cover layout..... | 30 |
| 5.1.7. Step 6: Proofing..... | 33 |
| 5.1.8. Step 7: Print preparation..... | 34 |
| 6. Case study: Magazine production..... | 35 |
| 7. Result and analysis..... | 37 |
| 7.1. Workflow..... | 37 |
| 7.1.1. Step 0. Color management..... | 37 |
| 7.1.2. Step 1: Manuscript preparation..... | 37 |
| 7.1.3. Step 2: Image preparation..... | 37 |
| 7.1.4. Step 3: Template construction..... | 37 |

| | |
|---------------------------------------|----|
| 7.1.5. Step 4: Page layout..... | 37 |
| 7.1.6. Step 5: Cover layout | 39 |
| 7.1.7. Step 6: Proofing | 39 |
| 7.1.8. Step 7: Print preparation..... | 39 |
| 7.2. Conclusions..... | 39 |
| 7.3. The future..... | 41 |
| 8. References..... | 42 |

Can free software replace proprietary software for graphic production?

1. Introduction

In the commercial area of graphic production (making books, newspapers, magazine, leaflets, folders and advertising) proprietary software clearly dominates over libre software. (See Appendix A. Usage statistics.)

The typical workflow involves using **Microsoft Word** for text entry, **Adobe Photoshop** for bitmap graphics and photo editing and **Adobe Illustrator** for vector graphics. The content is imported into **Adobe Indesign** for layout. The final product is a print-ready PDF document which is sent to the print shop.

While these are capable programs, they all suffer from the drawbacks of proprietary software: they are expensive, they tie the user to a single company (and the file formats of that company), support is hard to get and/or expensive and it's not easy (or often even possible) to add new functionality [1, 2].

1.1. Objective

The purpose of this report is to find out if these proprietary tools can be replaced by free alternatives. Is it possible to produce a print-ready PDF using only free tools? If it is, what software can (and should) be used and how? What are the drawbacks, if any? How can problems in handling and producing results be overcome?

1.2. Research questions

To find a solution we have to start by defining the problem. What are the needs of a graphic designer and what should free software be able to accomplish? To answer this question I have constructed a typical workflow for the production of a paperback and thereby identified the main steps and requirements of the graphic production process. This workflow is also applicable to other graphic production processes.

What free software tools can be used to accomplish this workflow? Are there any problems or drawbacks? How can they be solved or overcome?

1.3. Delimitations

This report does not address all cases of graphic design. The field is too large. It is also based on the assumption that the user is running an operating system that supports these applications and let them work together.

1.4. Intended audience

This report is written mainly for a graphic design professional or knowledgeable amateur familiar with the concepts of graphic design.

Can free software replace proprietary software for graphic production?

1.5. Definitions used

The following definitions are used in this thesis [3, 4]:

- Free: “the users have the freedom to run, copy, distribute, study, change and improve the software” (from the Free Software Foundation); free as in “free speech” not “free beer”.
- Libre: same as “free” (often used instead of “free” as this also means “gratis” in English).
- Open source: in day-to-day use, the same as “free”. There are a range of open source licenses, from Apache license where you are free to do anything in terms of modifications and reselling to the more strict GPL where you need to reveal your own source code.
- Proprietary: (closed source) software where the user pays a license fee to be able to use the software under certain conditions. Proprietary software can be offered for free but there is no possibility for the user to inspect and change the source code.
- FLOSS, FOSS: free (libre) open source software.

Can free software replace proprietary software for graphic production?

2. Method

I started with a literature study. This focused on two areas: What steps are involved in graphic production, and what free tools are available? The latter step also involved finding problems with the tools.

The results from the first researched area was used to construct a workflow. The tool research ended in several software suggestions. The workflow was reviewed by three persons who work professionally with layout and graphic design [5, 6, 7]. They read the report and validated the workflow according to their own experience with proprietary software. They also contributed with suggestions for the workflow and capabilities of the tools.

There is no theoretical method to decide whether the suggested software “solves” the workflow or not, so I decided on two case studies, These qualitative studies would let med find out if I could produce the expected results.

The studies were both real world cases. The first was volume 2 of a book I previously did for an end-customer using proprietary software. The second was an issue of a magazine for which I previously had constructed a graphic identity and template using proprietary software.

2.1. Reliability

Some of the software choices could be different. There are several options for text, bitmap and vector editing. An exhaustive test of all available software could be an interesting future project. For this report I tried to find the best software based on experience and literature studies.

The end result, two physical products, are the proof of concept in that it is impossible to tell whether they are made with proprietary or free tools.

2.2. Repeatability

If the process would be repeated the end result would stay the same. The software is still developed, including the operating systems, and as long as no features used in the production are removed from the tools, the same steps could be applied again.

2.3. Validity

Based on my ten year experience as a freelance graphic artist and journalist, I would say that the two test cases are both relevant and involves the most common aspects of graphic production. My three professional reference people supported this. That said, there are of course steps involved in graphic production not addressed in the two cases. These could for example be advanced use of vector graphics and typography.

I am both ideologically and practically in favor of open source solutions and to say that this has not affected my report would simply be untrue. The physical proof of the proposed free solution makes my bias a lesser problem.

Can free software replace proprietary software for graphic production?

3. Theory

3.1. Workflow

The graphic production workflow is, even with proprietary software, modular [8, 9]. There is no single application that does it all. Instead, several different programs take care of different parts or aspects of the creation. The same goes for free software, maybe even more so for two reasons: the modular approach is part of the open software ideology/movement and also because the applications are often not as extensively developed as their proprietary counterparts.

Step 1 of the workflow is performed by the author of the manuscript, while steps 2 and onwards are performed by the layout person. These are *roles*, and can both be performed by several people.

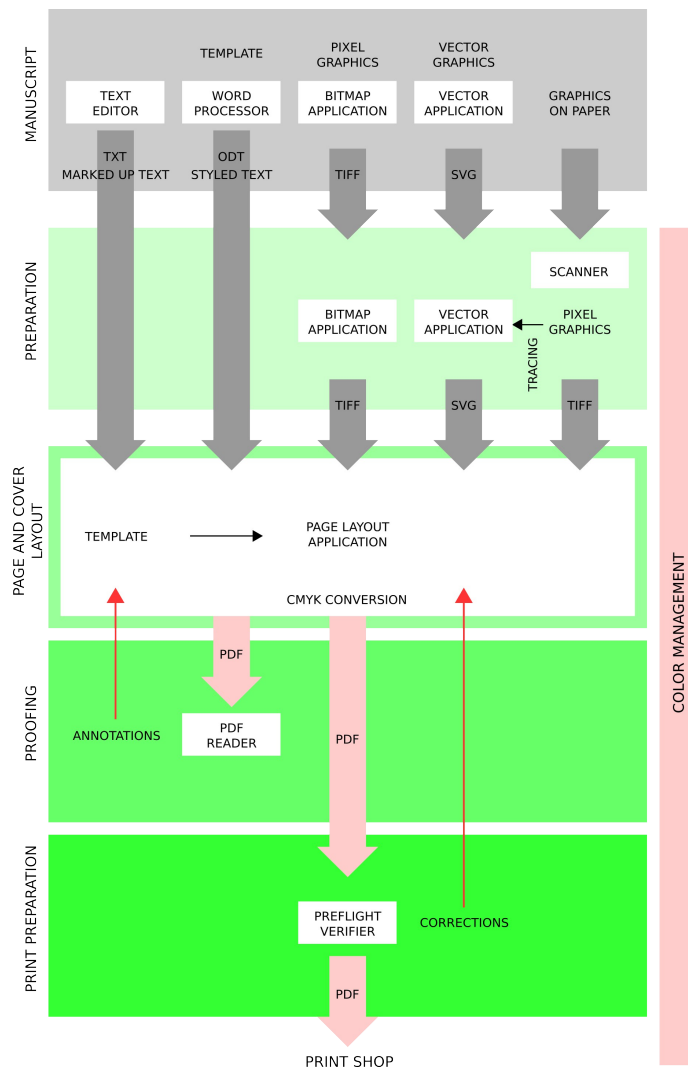


Figure 1. Workflow for preparing text and images in order to assemble material for a traditional four color printing process in CMYK.

Can free software replace proprietary software for graphic production?

3.1.1. Step 1: Manuscript preparation

A manuscript is in this thesis both the text and the graphics that are going to be source for the graphic production.

Entering and editing text

It's important at this stage to consider the import process into the page layout program. If the author and the layout person can agree on some rules as to how the text should be entered, like not styling the document and supplying images separately, much time can be saved. The author should also run a spell checker on the document. See Appendix B. Rules for entering and editing text.

The author should preferably tell what kind each paragraph (and optionally words or even characters) in the document is. Note that this is not a question of form but of function. How this step later translates into a layout is a task for the graphic designer (and the page layout software).

There are two different paths to follow. You can either tag the text with paragraph prefixes (and optionally suffixes) or using styles defined in the text editor/word processing software. If text is entered using a word processor with some kind of WYSIWYG capabilities, it's very useful if the layout person can provide a template with predefined styles for paragraphs, lists etc.

Text file formats

Save documents containing style information in a standard format, preferably Open Document Format for Office Applications (ODF), using the extension .odt [10]. If text is entered in a text editor (and tagged), save as a text file with UTF8-encoding as this encoding can represent every character in the Unicode character set.

Image production

When it comes to software for producing images, there are two different kind of graphics: bitmap graphics or vector graphics. Note: Image *production* is not about adjusting digital photographs for printing (see 3.1.2. Step 2: Image preparation).

Bitmap graphics are especially suited for photographic quality images and natural media images (digitally painted pictures where the software tries to mimic the behavior of traditional media). Bitmap software should preferably be able to handle all the actions described below (see 3.1.2. Bitmap image editing) but minimally let you: [5, 6, 7]

- Work in the correct resolution (a rule of thumb is a factor of 2 multiplied by the screening resolution of the finished product, so 300 dots per inch corresponds to 150 lines per inch of screening frequency, typical magazine printing).
- Save in a non-destructive format (non-compressed or lossless compression).
- Set the color space for correct color reproduction. Newsprint has a much smaller color space - colors that can be faithfully represented on paper - than fine artwork reproduction.

Vector graphics are useful for logotypes, maps, diagrams and technical drawings. Vector software

Can free software replace proprietary software for graphic production?

should have: [5, 6, 7]

- geometric tools: lines, circles, squares
- basic vector graphics editing capabilities
- curve-creation and editing
- path operations including boolean set operations on objects.
- text tools
- support for a standardized and widely used vector graphics file format, for example SVG or EPS (preferable SVG as it is an open standard format) [11]
- the ability to handle different color spaces so that it is possible to specify colors in both RGB and CMYK color models.
- the ability to auto-trace bitmap graphics and create a vector representation

Bitmap and vector file formats

See 3.1.2. Step 2: Image preparation for a description of file formats.

3.1.2. Step 2: Image preparation

The author should supply images in a “raw” format, that is without any applied image manipulation such as cropping or sharpening. If images are supplied as paper copies, they should first be scanned. “Graphic art”, ie logotypes, should be traced in the vector graphics software and transformed (traced) into vector art.

Bitmap image editing

The following actions are typically applied to an image before including it in the layout: [12]

1. Cropping.
2. Setting resolution.
3. Adjusting size.
4. Adjusting levels (setting black and white point).
5. Adjusting curves and adjusting brightness and contrast.
6. Adjusting the color balance and adjust hue/saturation.
7. Performing touch up: removing unwanted details.
8. Masking.

Can free software replace proprietary software for graphic production?

9. Sharpening.
10. Separation (if this is not handled by the page layout software, ie when creating the print-ready PDF).

Good selection tools and layer capabilities helps the workflow [5].

Bitmap file format

Desirable qualities of an image file format is that it

- uses non-destructive compression (or is non-compressed)
- handles clipping paths or transparency (or both)
- handles different color spaces
- preferably handles large color depths; more than 8 bits per color (channel) so that image manipulation has a lot of dynamic “room” to operate in.

Work is currently being done on the Open Raster format. The **Scribus** documentation team recommends TIFF and PNG [13, 14, 15].

Vector image editing

It's useful to open the vector document in a vector graphics application and do the following:

- Check and if necessary (and possible) adjust the color space.
- Clean up: remove stray points and lines.
- Check that no white frames (areas) are left that look like they are transparent (but are not).
- Perform simplifications to increase performance, like reducing the number of points in paths and removing stray point.

Vector file format

For vector graphics there is an open standard format, SVG. Color space in SVG is always sRGB [11, 16].

3.1.3. Step 3: Template construction

Using pen and paper or working directly in the page layout program, the graphic artist constructs a graphical identity for the product. This consists of all elements used in the finished product such as:

- page format and page size (including margins and columns)
- headers and footers

Can free software replace proprietary software for graphic production?

- typography: fonts, styles, paragraphs and their use
- graphical elements on the pages

3.1.4. Step 4: Page layout

The layout person imports text and images into the template using the page layout software. The page layout program should be able to: [5, 17]

- handle template pages (with headers and footer, including automatic page number generation)
- handle margins and columns
- handle grids (grid lines, align to grid functionality etc) for easy positioning
- allow for WYSIWYG editing including zoom in and out
- import text and images and place them in frames
- link text frames and allow text to flow between them
- be able to manipulate frames: move, rotate, align etc
- be able to arrange objects and move them along the z-axis (depth), ie change the stacking order
- employ basic graphic tools for creation of lines and boxes
- handle paragraph styles and preferably also character styles
- adjust the typographic content of text-frames: font style, size, kerning, text-wrapping, leading, hyphenation, justification, lists (numbered and bullet)
- generate table of contents and indexes
- run a preflight verifier (check that the document is ready for printing)
- handle color separation and ICC-profiles (see 3.2. Color management)
- create a print-ready PDF

Layout file format

The file format is not that important since this is the final stop before printing – the page layout program should be able to export to PDF. A human readable file format is preferable as it can be modified using a text editor and also allows for other programs to more easily manipulate the document.

If the layout person wish to change from one software to another there is no standardized and

Can free software replace proprietary software for graphic production?

open layout file format for print production. One could speculate that HTML and CSS will move into print production in the future. The current solution would be to export the layout as vector documents (PDF, EPS or SVG format) that can later be imported into other software.

3.1.5. Step 5: Cover layout

In book production the page and cover layout are separate steps, due to the fact that the pages are printed in black and white while the cover is in color on thicker paper of higher quality. In magazine production the practical difference does not exist.

3.1.6. Step 6: Proofing

Checking that the whole manuscript has been included and represented in a correct way in the document. This includes finding: [12, 6]

- spelling errors (should preferable already have been done at the manuscript preparation)
- cut-off or missing text (text frames too small or not linked)
- bad text flow: bad hyphenations as well as widows and orphans
- multiple blank characters in a row
- incorrect use of styles
- low readability (a hard copy printout is necessary for this)
- bad placement of images, text frames and graphic elements that decrease readability or put them too close to the edge of the paper or the binding.

The layout person creates a PDF version of the document and sends it for proof reading. The proof reader should be able to annotate, make notes of the errors, directly in the PDF document, and return this to the layout person.

3.1.7. Step 7: Print preparation

This is a three-part process:

1. Checking that the document will be able to print correctly, print proofing.
2. Creation of a print-ready PDF.
3. Checking the quality of the PDF.

Part 1: Print proofing

The preflight verification is a technical check on the content of the layout document: [12, 19]

- Are font files present and in the correct format?

Can free software replace proprietary software for graphic production?

- Are all image files present and of the correct resolution and color space?
- Is the page size correct?
- Is the color handling correct (is the document and the included images in the correct color space)?

This is Adobe's list of the top 10 problems with PDF documents: [20]

1. The resolution of images is too low.
2. Fonts are not embedded in the PDF.
3. The wrong color space is used.
4. The information about trim or bleed are incorrect.
5. There is an inconsistency with the native file. The original page, as viewed in the layout software looks different from the PDF. Hairlines might be different or gradients have changed.
6. A spot color is misnamed or it is accidentally converted to a process color.
7. Images are compressed too much. This causes a quality loss and in some cases artifacts appear inside or around the images.
8. The page size is incorrect.
9. There are problems with transparent objects.
10. ICC profiles are missing or incorrect.

Although all of them should be possible to avoid with the proper configuration of color management and layout software, you can not be sure that the PDF document is correct until you run a preflight verifier on it (and not even then - it still has to go through a RIP, Raster Images Processor).

Part 2: PDF creation

The page layout software is used to create a print-ready PDF document which is sent to the print shop. Settings for the PDF creation has to be done according to what kind of print process is used. (For Adobe software the settings are sometimes included in a so called joboptions-file supplied by the print shop.) The CMYK conversion, necessary for professional printing, is done at the same time (except for PDF/X-3). This step needs an ICC-profile, supplied by the print shop.

PDF file format

The document is sent to the print shop as a PDF document. There are several versions of the PDF format: [21]

- PDF 1.4 and 1.5 both handle transparency.

Can free software replace proprietary software for graphic production?

- PDF/X-1a specifies that all colors are CMYK or spot colors.
- PDF/X-3 tags all images with the right ICC profile and leaves the CMYK conversion to the RIP (Raster Images Processor) at the print office.

X-1a and X-3 are based on PDF 1.3 which in turn does not handle transparency.

Part 3: PDF quality control

A PDF preflight program can check many things inside the PDF, like image resolution, font embedding etc. It can also check that the PDF is of the right version. For very reliable color reproduction the print shop can produce a paper proof before printing all copies.

3.2. Color management

From step 2 and onwards color management has to be applied. This ensures that the original colors are kept throughout the layout process and that the printed end result is predictable.

3.2.1. Color management crash course

How can we keep the colors consistent all through the digital graphic design workflow? How do we make sure that what we see on the screen is what will come out of the printing press? The answer is to use a Color Management System (CMS) which in turn uses color profiles (ICC profiles) that describe how color numbers in a document translate to actual color appearances. This means that each input and output device have their own ICC profile, that is. all scanners, screens, cameras, printers and foreign imports such as ads from third parties must be tagged with their respective ICC profile. In the case of printers, several profiles could be used depending on the type of paper and ink used.

All images exist in a given color space. For example, a camera attaches it's own color profile to the image it produces (it »tags« the image). This means for example that the image (ie raw data, color numbers) combined with the camera (input) ICC profile and the monitor (output) ICC profile will reproduce faithfully on the screen.

Can free software replace proprietary software for graphic production?

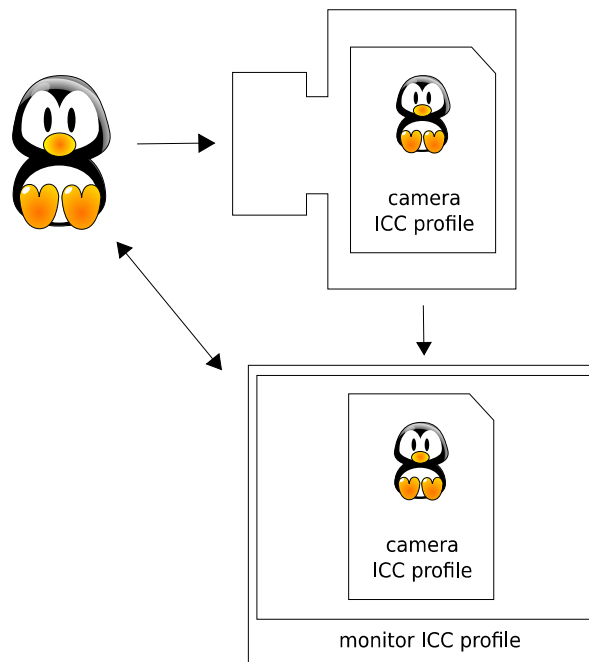


Figure 2. The color management tries to reproduce reality faithfully on a computer monitor or printer.

The color space can also be a device independent “working space”, for example sRGB IEC61966-2.1 or Adobe RGB, which is used when converting between different color spaces. The CMS handles the color management and also supports on-screen WYSIWYG previewing. Color managed applications can convert images from one device ICC profile to another (a conversion to a different color space). ICC profiles are based on an open specification and can be used on different operating systems. Note that ICC-profiles for printing should be unique for every printer, paper and ink combination (in practice often only for printer and coated/uncoated paper). [12, 22, 23, 24]

3.2.2. Create a color managed workflow

The following steps are necessary to set up a color managed workflow:

1. Install and enable a CMS.
2. Calibrate the screen (and the scanner).
3. Configure the software.

Can free software replace proprietary software for graphic production?

4. Software

The purpose of this chapter is to identify free software fulfilling the specifications, that is, match the workflow requirements. Experience gained during the case study will possibly have an impact on these choices - the recommendations in this chapter are based on findings in literature and on the Internet as well as from my own experiences as a layout professional and graphic designer.

4.1. Workflow

4.1.1. Step 1: Manuscript preparation

Entering and editing text

There are several free word processing applications available, allowing the user to input and edit text. Among the most widely used are:

- **OpenOffice Writer**
- **LibreOffice Writer** (a fork of OpenOffice)
- **Calligra** (former KOffice)

If using paragraph prefixes a pure text editor is also an option. The word processor or text editor should have a spell checker.

Recommendation

LibreOffice Writer due to the rich feature set, wide following and active development.

Image production: Bitmap graphics

GIMP (<http://gimp.org>) is feature-rich, has color management but no native color separation. The last is not a problem if we let the page layout software do the color separation at PDF creation time. **GIMP** only supports 8-bit color channels although 16-bit support is on the way [25]. RAW photographs can be opened with the **UFRaw** plugin (<http://ufraw.sourceforge.net/>). If necessary, **GIMP** can color separate using **Separate+** (<http://cue.yellowmagic.info/software/separate-plus/index.html>).

Krita (<http://www.calligra.org/krita>) of Calligra (former KOffice) is feature rich, offers color separation and 16-bit color channels, but is heavy and has no multi document support. It also focuses more on painting than image manipulation.

MyPaint (<http://mypaint.intilinux.com>) is an application for natural media painting, not photo editing. MyPaint has no selections and no transforms.

CinePaint (<http://www.cinepaint.org>) supports 8-bit, 16-bit and 32-bit color channels and uses LittleCMS for color management. It is an early fork of GIMP but the development is sporadic and it is not easy to install [26].

Phatch (<http://photobatch.stani.be/> and <http://photobatch.wikidot.com/>) can be used for batch

Can free software replace proprietary software for graphic production?

operations, as can **ImageMagick** (<http://www.imagemagick.org/script/index.php>).

There is also several applications for manipulation of digital photos: **Darktable** (<http://www.darktable.org/>), **RawTherapee** (<http://rawtherapee.com>) and **DigiKam** (<http://www.digikam.org/>). They support more than 8 bit colors but doesn't allow for any retouch work.

Recommendation

GIMP. [28] It has many plugins and a large community. Only 8-bit color support which could pose a problem but probably not [27, 28, 29]. **Darktable** can be used when necessary.

Image production: Vector graphics

Inkscape (<http://inkscape.org>) is a SVG editor with tracing and some limited CMYK handling [30].

Karbon (<http://www.calligra.org/karbon>) is the vector drawing application of Calligra (former KOffice). No support for tracing but it handles CMYK.

LibreOffice Draw/OpenOffice Draw has basic vector functions and SVG support. No support for tracing.

Tracing

Trace functionality can be handled with **potrace** (<http://potrace.sourceforge.net>), which is what **Inkscape** is doing "in the background". There is a graphical frontend for **potrace** called **Potracegui** (<http://potracegui.sourceforge.net>).

Recommendation

Inkscape. It has a rich feature set including tracing [31].

4.1.2. Step 2: Image preparation

See 4.1.1. Image production: Bitmap graphics for a list of software for bitmap graphics.

4.1.3. Step 3-5: Layout

There is only one option for professional page layout using free tools: **Scribus**.

Scribus (<http://www.scribus.net>) is an application for WYSIWYG page layout with color management and PDF export. It has one major problem (for our book project): it can't handle long documents (ie many linked text frames). The solution is to split the project into many smaller documents and then concatenate the resulting PDF files before sending them to the print shop. [32, 33]

Recommendation

Scribus. The only option. [17, 34, 35]

4.1.4. Step 6: Proofing

We need software that can show PDF documents and let the user make annotations (notes of errors) in the PDF document. If this is not possible (or desirable) a separate text document can be

Can free software replace proprietary software for graphic production?

used for noting corrections. [36, 39]

Jarnal (<http://jarnal.wikispaces.com/>) is not fully open source (dependent on JPedal library which is partly proprietary).

PDF Edit (<http://pdfedit.cz/en/index.html>) has no functionality for annotations.

Okular (<http://okular.kde.org/>) saves annotations outside of the PDF file which makes it harder to send corrections between people.

Xournal (<http://xournal.sourceforge.net/>) is a notetaking application which supports annotation of PDF files.

Evince (<http://projects.gnome.org/evince/>) is also an option which supports annotation of PDF files. It is included in the default Ubuntu installation.

Recommendation

Xournal as it is free, supports annotations and keeps them in a PDF document. [37, 38]

4.1.5. Step 7: Print preparation

1. Print proofing

A preflight verifier exists in **Scribus**. It checks that all content in the document is visible and that images are present and of the correct resolution. [40]

Examining the previously mentioned Adobe's list of the top 10 problems with PDF:s, we can see that different methods have to be used for the problems:

1. The resolution of images is too low. This is checked with the Scribus preflight verifier.
2. Fonts are not embedded in the PDF. In Scribus, check the **Save as PDF > Fonts** tab: Make sure you have embedded all available fonts.
3. The wrong color space is used. In Scribus, check **File > Document Setup > Color Management**. Are the correct ICC profiles assigned?
4. The information about trim or bleed are incorrect. Make sure you have assigned (the correct) bleed values in the **Document Setup** window (or when creating the **Scribus** document). Also remember to check **Use Document Bleeds** in the **Save to PDF** window **Pre-press** tab.
5. There is an inconsistency with the native file. The original page, as viewed in the layout application looks different from the PDF. Hairlines might be different or gradients have changed. This can only be checked after the PDF document creation. A printout can possibly expose some problems. Otherwise, order a printed proof from the printer.
6. A spot color is misnamed or it is accidentally converted to a process color. If you *do* want spot colors (which you most often don't want when doing normal four color CMYK

Can free software replace proprietary software for graphic production?

printing), uncheck **Convert spot colors to process colors** in the **Colors** tab in the **Save as PDF** window.

7. Images are compressed too much. This causes a quality loss and in some cases artifacts appear inside or around the images. In the **General** tab in the **Save as PDF** window set the **Compression Method** to **Automatic** and the **Compression Quality** to **Maximum** (which corresponds to 95%).
8. The page size is incorrect. Make sure you have assigned the correct width and height in the **Document Setup** window (or when creating the **Scribus** document). Also remember to use the correct values for bleed and the correct printer marks in the **Save to PDF** window **Pre-press** tab.
9. There are problems with transparent objects. Try to avoid using transparency if possible.
10. ICC profiles are missing or incorrect. Examine the images in **Scribus** and check that they are tagged with the correct ICC profiles. Use the **Image** tab in the **Properties** window.

2. PDF creation

PDF creation is fully supported in **Scribus** [41]. **Pdftk** and **PDF Chain** (<http://pdfchain.sourceforge.net>), a GUI for **Pdftk**, can be used to concatenate PDF pages.

3. PDF quality control

Libre PDF preflight software is not available except for a somewhat limited tool written in **Ruby** [42]. It's possible to check that the PDF document is of the right version using **Pdftk**.

4.2. Color management

Create a color managed workflow using the following steps:

Step 1. Install and enable a CMS

Step 2. Calibrate the screen

Step 3. Configure **GIMP**

Step 4. Configure **Inkscape**

Step 5. Configure **Scribus**

If you are using a scanner it too has to be calibrated but this is outside of this report.

4.2.1. Step 1: Install and enable a CMS

Argyll CMS is a free CMS consisting of a set of command line tools. Ubuntu (12.04) has color management built in (and installs Argyll when needed) and supports some color calibration hardware out of the box. Ubuntu uses the **GNOME Color Manager** (which utilizes both the **Argyll CMS** and **colord**). [43, 44, 45, 46, 47, 48]

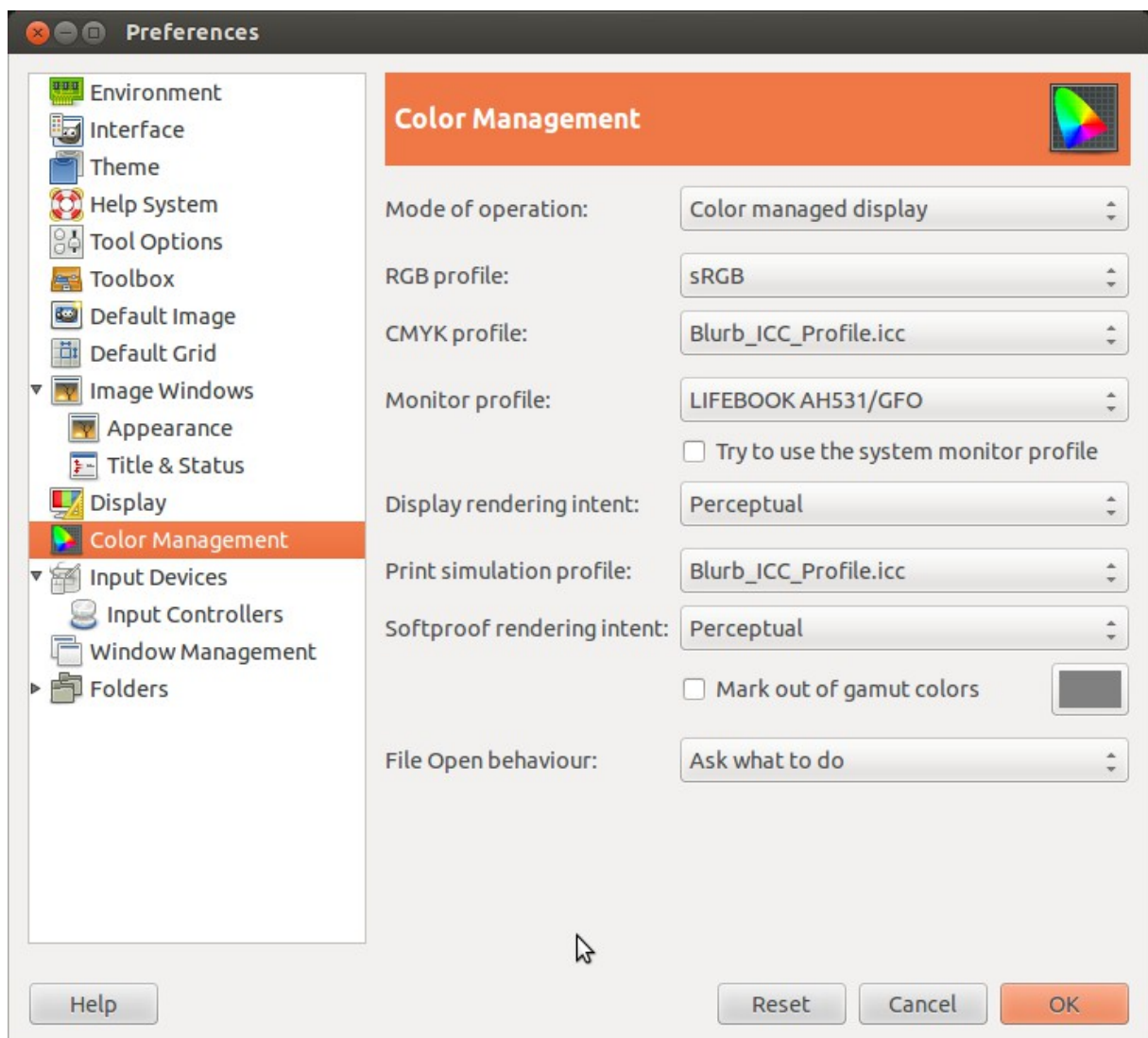
Can free software replace proprietary software for graphic production?

4.2.2. Step 2: Calibrate the screen

The calibration soft- and hardware examines your screen and creates an ICC profile describing how it represents colors. The **GNOME Color Manager** can be used to calibrate both screens and input devices like scanners [54]. For screen calibration a colorimeter is also needed, a hardware device which measures the color output of the display. Color Hug is an open hardware colorimeter [49, 50, 51, 52, 53].

4.2.3. Step 3: Configure GIMP

The photo editing will be done in the RGB color space. There are two things necessary to enable color management in **GIMP** (2.8). [60, 61, 62, 63, 64]



Figur 3. GIMP color managment.

1. In the **Color Management** section of the **Preferences** (**Edit > Preferences > Color Management**):

Can free software replace proprietary software for graphic production?

- Set **Mode of operation** to **Color managed display** (the default). Set it to **Print simulation** for soft proofing.
- Set **RGB profile**. This is the working profile that **GIMP** will use with the open images. sRGB or Adobe RGB are good choices.
- Set **CMYK profile** used for color conversion. (Not really necessary as we are only going to work in the RGB color space.)
- Set **Monitor profile** to the ICC profile for the monitor created in step 2 (“Calibrating the screen”). (Or use the **Try to use the system monitor profile.**)
- Set **Rendering Intent** to **Perceptual** (best for photographs).
- Set **Print simulation profile** to the CMYK (output) profile. This is the profile used during soft proofing (when **Mode of operation** is set to **Print simulation**).
- Set the **Softproof rendering intent** to **Perceptual**.
- Uncheck **Mark out of gamut colors**. Although it's useful to be aware of colors that cannot be printed **GIMP** doesn't seem to handle this correctly. [56, 57]
- Set **File Open Behavior** to **Ask what to do**.

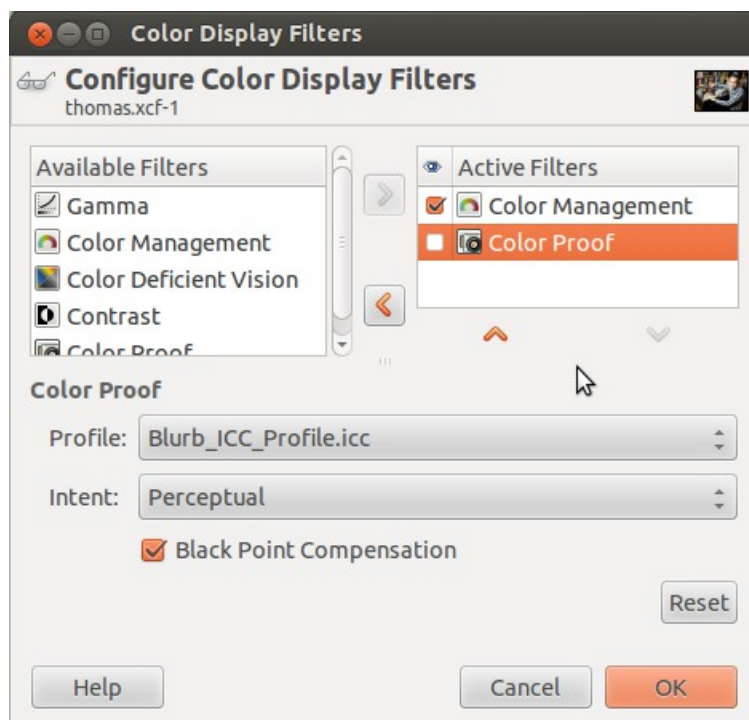


Figure 4. GIMP display filters.

2. In the **Color Display Filters** dialog box (**View > Display Filters**), make sure **Color Management** is active (placed in the right hand list and checked).

Can free software replace proprietary software for graphic production?

For soft proofing enable the **Color Proof** display filter and tell it to use the **Print simulation profile** mentioned in above and the **Perceptual rendering intent**. This filter has been receiving some criticism and it is recommended not to use it and go for the **Print simulation Mode of operation** instead. [58, 59]

4.2.4. Step 4: Configure Inkscape

Inkscape works in sRGB. SVG doesn't handle CMYK colors. This is not a problem since the color separation will be handled by **Scribus**. Set up a proper display of colors by choosing **File > Inkscape Preferences > Color Management**: [16, 30]

- **Display profile**: select the monitor profile.
- **Display rendering intent**: **Perceptual**.
- **Simulate output on screen**: Checked
- **Mark out of gamut colors**: Checked.
- **Device profile**: Set this to the CMYK (output) profile.
- **Device rendering intent**: **Perceptual**.
- **Black point compensation**: On.

View > Color-managed view should be checked.

4.2.5. Step 5: Configure Scribus

1. Setting up color management

In **Scribus**, chose **File > Preferences > Color Management** and check **Activate Color Management**. Specify the following: [24]

- **RGB Images**: select the profile used in the creation of images existing in the RGB color space. This is the profile used in the image editor, camera etc. If the images are already tagged with an ICC color profile, that one will be used instead.
- **CMYK Images**: select the profile used in the creation of images existing in the CMYK color space. Remember to use the same profile as target profile if you chose to do the CMYK conversion in the image editor program.
- **RGB Solid Colors**: the color profile for any RGB solid colors used in **Scribus**.
- **CMYK Solid Colors**: the color profile for any CMYK solid colors defined and used in **Scribus**.
- **Monitor**: select the monitor profile.
- **Printer**: Contact the print shop and ask for the printing press ICC profile for the project.

Can free software replace proprietary software for graphic production?

- To see what the final printing press output will be, check ***Simulate printer on the screen*** and ***Convert all colors to printer space***.

2. Producing a PDF

Scribus is used to produce a PDF for on-screen viewing as well as for the print shop [24, 65].

Chose **File > Export > Save As PDF**.

4.3. Other tools and resources

4.3.1. Font management

FontMatrix (<https://github.com/fontmatrix>) is a free font manager. [66]

4.3.2. Font creation

FontForge (<http://fontforge.sourceforge.net/>) is a tool for font creation and editing.

4.3.3. Fonts

Here are some places to find free open source fonts: [67]

- Open Font Library, <http://www.openfontlibrary.org>,
http://en.wikipedia.org/wiki/Open_Font_Library
- The League of Movable Type, <http://www.theleagueofmoveabletype.com/>
- OSP Foundry, <http://ospublish.constantvzw.org/foundry/>
- VTF Type Foundry, <http://velvetyne.fr/>

4.4. Operating system

Any free OS capable of running the recommended software applications is suitable.

4.4.1. Recommendation

Ubuntu. It has built-in color management and can run all needed applications. The user experience is close to what a graphic designer expects and is used to. It is easy to set up, stable and has a large user base. [68]

Can free software replace proprietary software for graphic production?

5. Case study: Paperback production

The first case study is a collection of blog posts expected to result in a book of about 250 printed pages (which later turned out to be almost 100 pages more). The book is targeted for print-on-demand publishing on Blurb (blurb.com), who require a PDF/X-3 compliant document.



Figure 5. Cover of book produced with Adobe Indesign.

As I have already produced volume 1 of the book using the Adobe Indesign suite, it was an opportunity to compare the two solutions: proprietary and free software. I chose Ubuntu (12.04 LTS) as the operating system.

5.1. Workflow

5.1.1. Step 0. Color management

Calibrate the screen

I started up the **GNOME Color Manager** and attached a Spyder 3 Express calibration device to the computer. It was automatically detected and I was asked whether I wanted to install the additional software required for calibration.

Can free software replace proprietary software for graphic production?

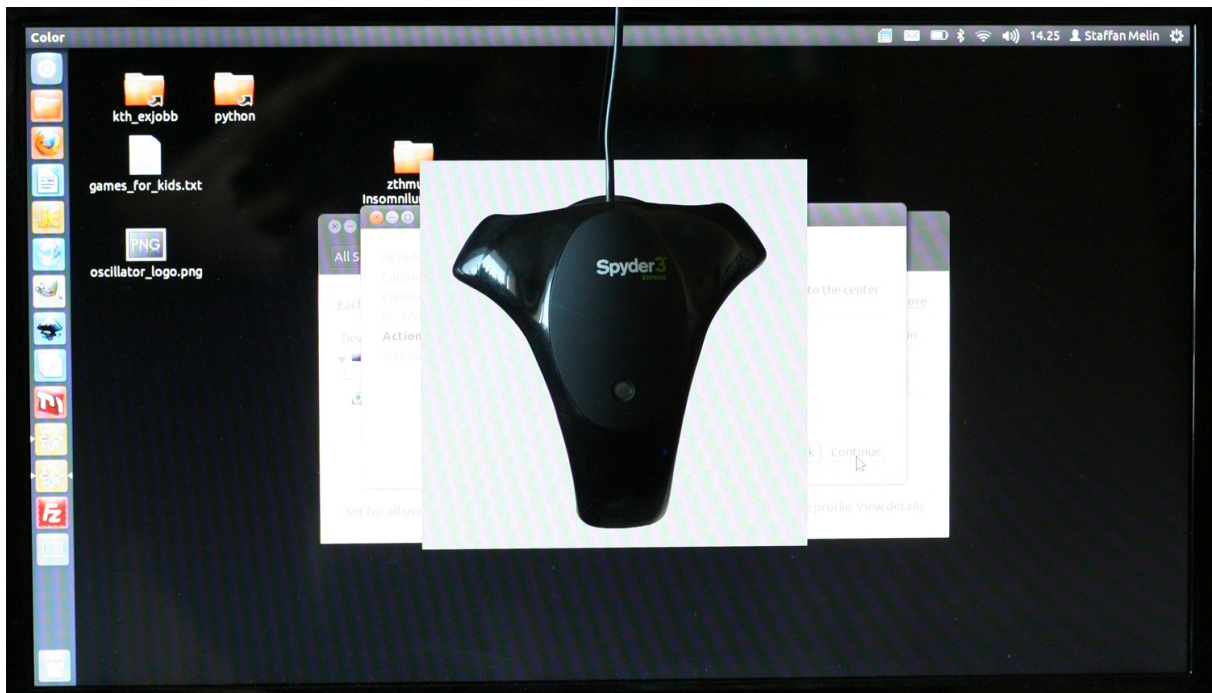


Figure 6. The GNOME Color Manager calibrating a laptop screen using the Spyder 3 Express calibration device.

I was then guided through the calibration process which took approximately 10 minutes (I choose the “normal” calibration process). My screen target white point was 6500K [69, 70]. When finished I was asked to name the new profile (an ICC file) [71]. On **Ubuntu**, ICC files are saved in `~/.local/share/icc`.

I first carried out a visual comparison using two different monitors both calibrated using the Spyder (one in **Ubuntu** and one in **Windows 7**). The test image was a little bit more yellow and a bit lighter on the **Ubuntu** screen but the difference was small (and the differences can possibly be attributed to the low-cost laptop screen on the **Ubuntu** computer).

Can free software replace proprietary software for graphic production?

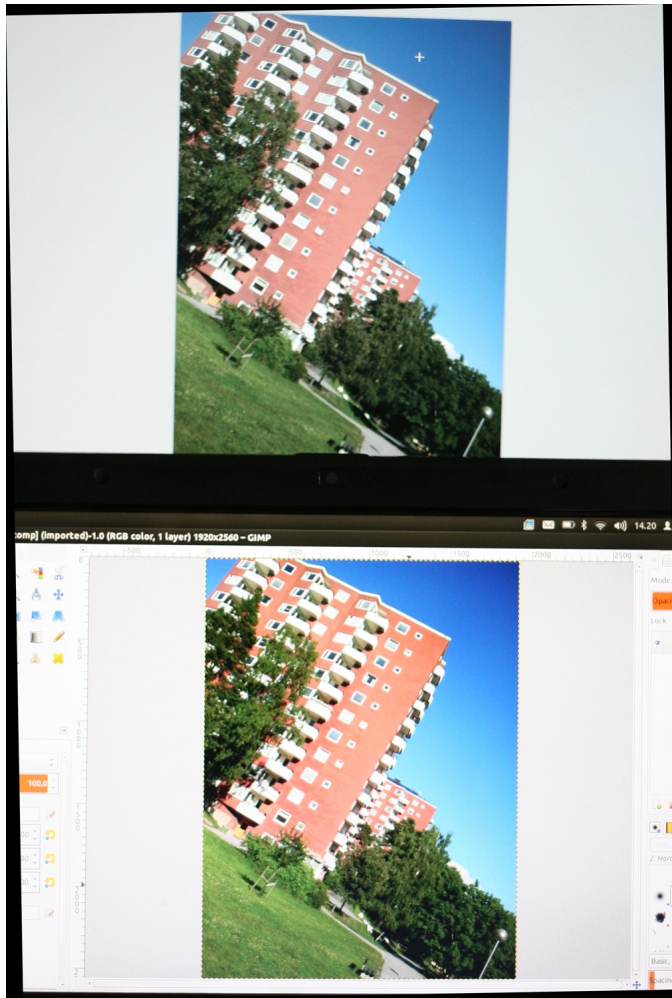


Figure 7. Comparison of a two calibrated monitors. Windows 7 (top) and Ubuntu 12.04 (bottom).

I calibrated the same screen using the two different operating systems which yielded better results: overall brightness was the same - the image was a bit more yellow on the **Windows 7** system.

Configure Scribus

Scribus is installed via **Ubuntu Software Center** with the additional add-on “ICC-profiles for use with color profile aware software (icc-profiles)”.

For output I needed Blurb's ICC profile so I downloaded it directly from http://www.blurb.com/downloads/Blurb_ICC_Profile.icc and saved it to `~/color/icc` [80]. For **Inkscape**, I copied the Blurb profile to `~/local/share/color/icc`. I copied the monitor profile obtained in the calibration step from `~/local/share/icc`. to `~/color/icc`. For **Inkscape**, I copied the monitor profile to `~/local/share/color/icc`.

I then had to restart **Scribus** so it found the ICC profiles. I chose **File > Preferences > Color Management** and checked **Activate Color Management** and applied the following settings: [24]

Can free software replace proprietary software for graphic production?

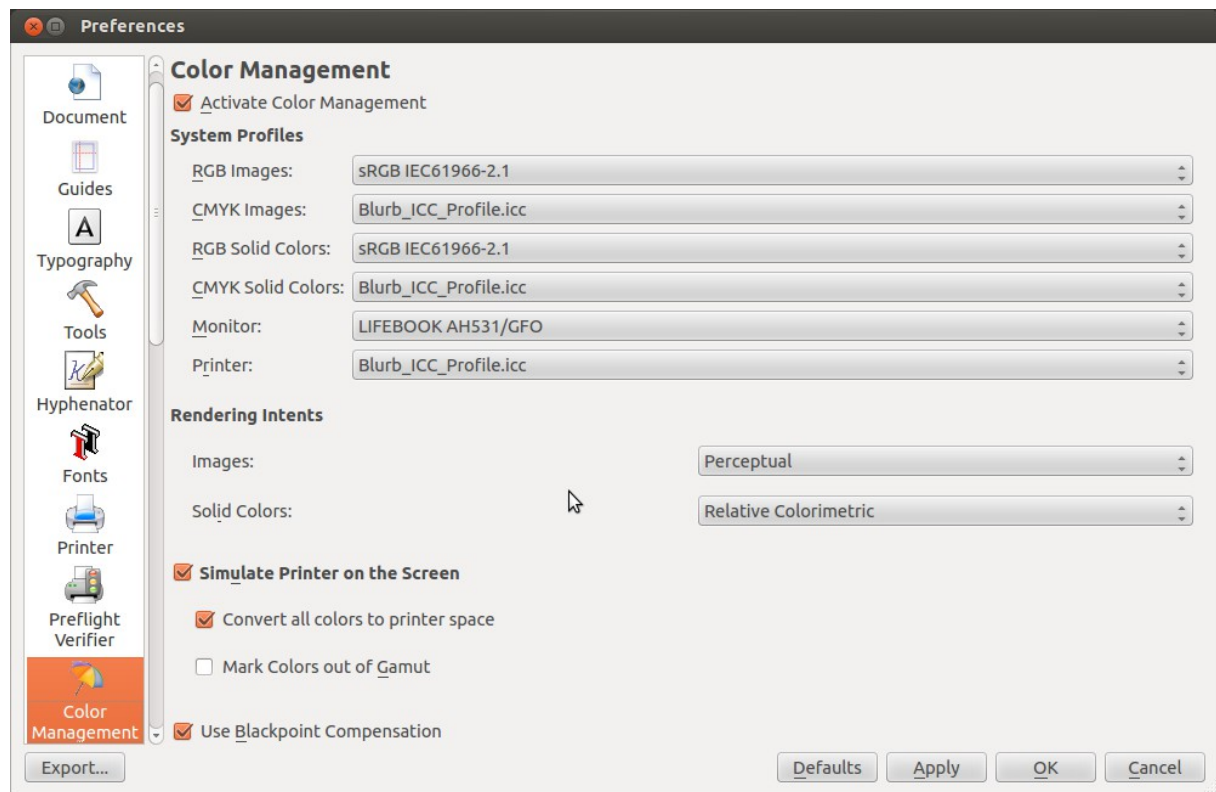


Figure 8. Scribus color management settings (see Appendix C, section 1).

Configure GIMP

First of all I configured the **GIMP** color management (see 4.2.3. Step 3: Configure GIMP) and went into single window mode (**Windows > Single-window mode**).

Configure Inkscape

I then turned on color management in **Inkscape** (see 4.2.4. Step 4: Configure Inkscape) using the “Blurb_ICC_profile.icc” profile as **Device profile**.

5.1.2. Step 1: Manuscript preparation

The manuscript preparation stage consisted of an initial meeting where I met the author of the book to discuss what kind of information was in the script. He was going to use **OpenOffice** for text entry.

Can free software replace proprietary software for graphic production?



Figure 9. Typical blog post.

We ended up with the following paragraph styles:

- **Heading 2:** Blog entry title (a predefined OpenOffice style).
- **date:** date line, eg "Thursday, August 16, 2007".
- **Text body:** body text (a predefined OpenOffice style).
- **quote:** for quoted paragraphs.
- **listing:** for lists of items.

Character styles:

- **title:** For record titles.
- **title in quote:** For titles mentioned in quoted text.

We also agreed on the following rules:

- Just hit the Enter key once for a new paragraph (ie, no blank lines). The **Text body** style was configured for displaying a blank line between paragraphs for easy reading.
- Use - (two dashes) for long dash.

The author tested out the template and, after a second meeting where we adjusted and expanded the template (with a style for lists), used it to enter the complete manuscript. The manuscript was delivered as an .odt file made in **Open Office 3.5**.

Can free software replace proprietary software for graphic production?

5.1.3. Step 2: Image preparation

Two images (record covers) were delivered as a source for the cover. Using these I was going to reproduce the cover of the first volume using a different color scheme (the process is described in 3.1.5. Step 5: Cover layout). No images or graphics were used on the inner pages.

5.1.4. Step 3: Template construction

Fonts

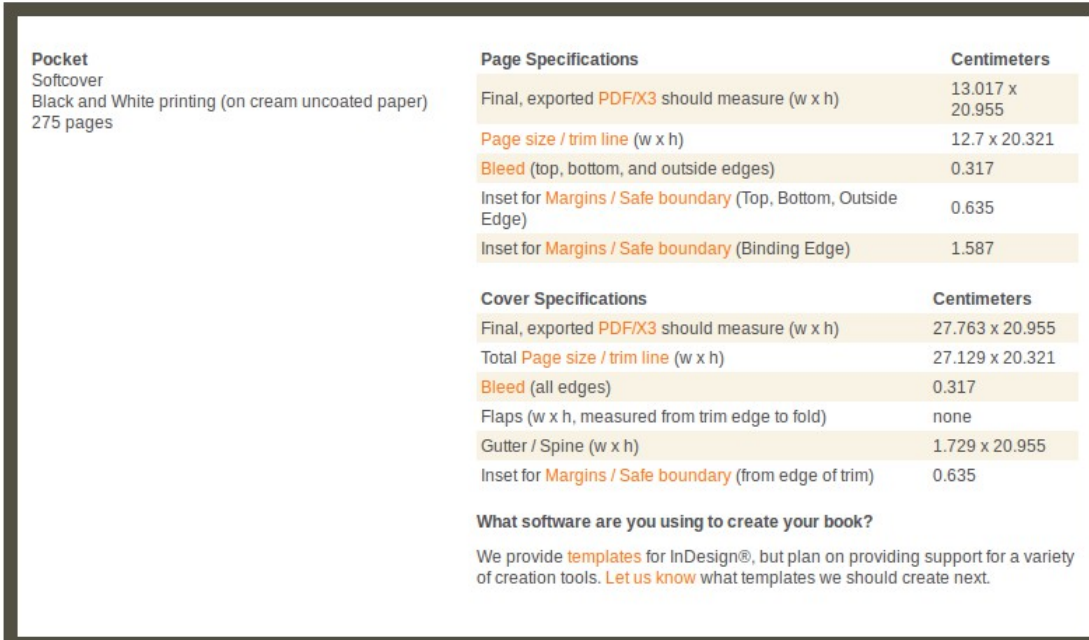
The layout was going to use the DejaVu font family, the same used in volume 1. They are distributed under a free license, so I downloaded the fonts and installed them. [73]

Template measurements

Using the Blurb booksizer calculator for “Pocket 5x8 inches (13x20 centimeters)” [74] I entered the following values:

- Cover type: Softcover
- Page count: 275 (final count was 328, so I had to redo this step later on)
- Paper type: Black and white printing
- Unit of measurement: centimeters

The calculator returned the necessary size of both pages and cover.



The screenshot shows the Blurb calculator interface. On the left, it lists the book details: Pocket Softcover, Black and White printing (on cream uncoated paper), and 275 pages. The main area contains two tables of specifications, both in centimeters.

| Page Specifications | Centimeters |
|---|-----------------|
| Final, exported PDF/X3 should measure (w x h) | 13.017 x 20.955 |
| Page size / trim line (w x h) | 12.7 x 20.321 |
| Bleed (top, bottom, and outside edges) | 0.317 |
| Inset for Margins / Safe boundary (Top, Bottom, Outside Edge) | 0.635 |
| Inset for Margins / Safe boundary (Binding Edge) | 1.587 |

| Cover Specifications | Centimeters |
|---|-----------------|
| Final, exported PDF/X3 should measure (w x h) | 27.763 x 20.955 |
| Total Page size / trim line (w x h) | 27.129 x 20.321 |
| Bleed (all edges) | 0.317 |
| Flaps (w x h, measured from trim edge to fold) | none |
| Gutter / Spine (w x h) | 1.729 x 20.955 |
| Inset for Margins / Safe boundary (from edge of trim) | 0.635 |

What software are you using to create your book?
We provide templates for InDesign®, but plan on providing support for a variety of creation tools. Let us know what templates we should create next.

Figure 10. Blurb calculator.

Can free software replace proprietary software for graphic production?

Scribus setup

I started up **Scribus** and created a new double sided document with the settings described in Appendix C, section 2.

Master pages

To keep the look of the pages consistent, **Scribus** has a template function called “Master pages”. All elements (text, images, guide lines) placed on these pages appear on all the pages that use them. They are therefore useful for guidelines (which won't print), page numbers, headers and footers (which will print). [24]

I selected **Edit > Master Pages** and the **Edit Master Pages** dialog appeared. I clicked **Normal Left**. The **Scribus** main window now showed the master page for a left page. I selected **Page > Manage guides**. A **Guide Manager** window appeared. I clicked the **Single** tab. I wanted to insert one vertical guide (lines that won't print but will aid us when positioning elements on the page) and clicked **Add** and entered the following value: 6.35 mm. I next added two horizontal guides (for page header and line number) at 6.35 mm and 196.86 mm. I then checked **Lock guides** and closed the **Guide Manager** window.

I next repeated the procedure for **Normal Right**. I selected **Page > Manage guides** and clicked the **Single** tab. I clicked **Add** and entered the following values: 120.65 mm (127 - 6.35 mm). I finally added the two horizontal guides (for page header and line number) at 6.35 mm and 196.86 mm. I then checked **Lock guides** and closed the **Guide Manager** window. I finally closed the **Edit Master Pages** window.

Baseline grid

The baseline grid is a set of horizontal lines, one for each text line, that I used to align the body text. I chose **File > Document setup** and clicked on the **Guides** icon: [24]

- **Show baseline grid:** Checked
- **Baseline grid:** 12 pt (the same as the linespacing of the **Text body** paragraph style)
- **Baseline Offset:** 15 mm (the same as the top margin of the template)

I checked **Page > Snap to guides**. and **View > Show Baseline Grid**.

Styles

Styles can be of two kinds: paragraph styles and character styles. A paragraph is all text entered before the user presses the Enter key and a paragraph style applies to all the characters. A character style on the other hand, can be applied to as few as a single character (but is normally used for one or a few words). [24]

I brought up the **Style Manager** by choosing **Edit > Styles** and redefined the two default styles, defined the paragraph styles and the character styles. (See Appendix D. Style definitions in Scribus.)

Headers and footers

On the **Edit > Master Pages**, I entered the header “The Elvis Today Blog Volume 2” and positioned it in the upper left and right corner (I used the **header** style, snapped to the guides, and right

Can free software replace proprietary software for graphic production?

aligned the header on the right master page). For automatic page numbers, I created a text frame and used **Insert > Character > Page Number**. I applied the **footer** paragraph style and aligned the frame to the guide line. I saved the document as "elvistoday_vol2_pages.sla". This was going to be the template.

5.1.5. Step 4: Page layout

I already knew that **Scribus** would have a problem handling the full manuscript in one document. The idea was to divide the book into several pieces which would later be exported as PDF documents and stitched together. As the book didn't contain any chapters (which would have been good places to split the document) I used the following strategy:

1. Copy the **Scribus** template and open it.
2. Insert about 10 pages of text.
3. Check for a suitable page to end the part of the book. A suitable page is where there is no space left at the bottom to start a new blog entry (two blank lines, header, date, blank line, two lines of text).

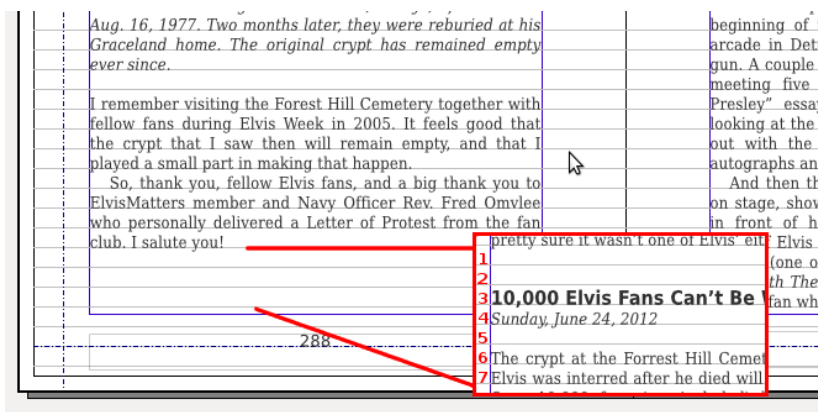


Figure 11. Suitable place to end a part of the book.

4. Delete everything after this (text and pages).
5. Repeat for the rest of the manuscript.

First try: Import of formatted text

I started by copying the manuscript, opened the copy, deleted everything except the 20 first blog entries and saved it.

In **Scribus**, I inserted 20 pages. Next, I started the import by choosing **Insert > Frames**. The import function created new style names, which unfortunately didn't have the same names as the original styles (even though the names were identical in both templates).

This could be solved by deleting the new styles and replacing them (this is handled by the delete style function). Not all styles imported correctly. The **quote** style was applied to the correct

Can free software replace proprietary software for graphic production?

paragraphs but it didn't look correct until I re-applied the style. The plan was to go through the document and fix all formatting, but then I found out that the import-function does not handle character styles correctly so I abandoned this approach. [72, 73]

Second try: Import of unformatted text

Next I tried to import the document as a text file (I checked **Import Text Only** in the **Insert > Frame** function). I applied the **Text body** style to the entire document. I then went through the document using the **Story Editor** (by selecting the first text frame, right click and select **Edit Text**) and applied the correct styles. This worked but it took a long time to create the text documents to import.

Third try: Copy and paste unformatted text

I re-evaluated the process and settled with not importing but copying-and-pasting text from the manuscript into the **Scribus** document. The copy-and-paste process was easier than continually creating new text documents.

The final process

I ended up with the following workflow for getting the text from **LibreOffice** into **Scribus**:

1. Copy the **Scribus** template to a new sequentially numbered copy.
2. Open the copy and adjust the page number: **Document Setup > Sections**. If the first page is a left page: **Document Settings > Document: First page is: Left Page**.
3. Create 19 new pages (for a total of 20 pages). Insert linked frames on all pages (**Insert > Frames**).
4. Copy the next 10 blog entries from **LibreOffice** and paste them into **Scribus**.
5. Select all text and apply the **Text body** paragraph style to the whole text.
6. Replace “-” with an em-space (which can be found using the find function of the **Ubuntu Character map** application, “Inherited” script for DejaVu font). (It would have been easier to do this once in the manuscript.)
7. Apply the correct paragraph styles using the **Story Editor: Heading 2, date, Text body - first paragraph**.
8. Apply paragraph and character styles by visually scanning the manuscript: **quote, list, title**.
9. Locate suitable places to end: a page where there are less than 7 rows left on a page after a blog entry ends. If a good place to end can't be found, repeat from step 3 (but insert something like 5 pages and 5 blog entries).
10. Remove any remaining text and the empty pages at the end. Mark up used blog entries in the manuscript (I used the highlighting color tool).

I did a final check of the text, looking for and adjusting widows (sometimes resorting to adjusting

Can free software replace proprietary software for graphic production?

the letter-spacing for a good looking text flow) and quotes inside of quotes. Using this workflow I ended up with 5 separate **Scribus** documents, between 40 and 120 pages long. 120 pages is a maximum, anything more makes the editing too slow.

5.1.6. Step 5: Cover layout

The idea from the author was to replicate the cover of the first volume which was inspired by the Elvis record cover (more correctly a CD reissue):

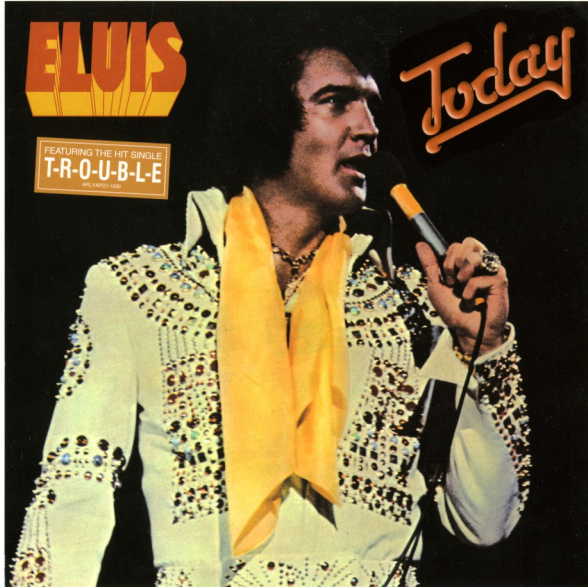


Figure 12. Elvis record cover.

On the back of the original LP album the logo was in blue, so this was a natural choice for the second volume:

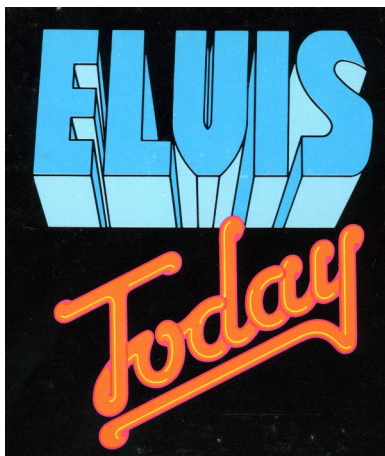


Figure 13. Blue logo for volume 2.

I also had a digital photograph of the author.

Can free software replace proprietary software for graphic production?

Setting up GIMP and Inkscape

I made sure **GIMP** was in single windows mode (*Windows > Single-window mode*).

Elvis silhouette

I first started to work on the Elvis silhouette. Using **GIMP** I cleaned up the scanned Elvis photograph and prepared it for the manual trace in **Inkscape**. I increased the brightness and contrast to show the hair and erased some background to make Elvis stand out more from the background. Using **Inkscape** I manually traced Elvis' silhouette using the Bezier path tool. I filled it and saved it as a SVG file.

Title logo

I next started to work on the title logo, which consisted of three parts: "ELVIS", "today" and "BLOG". For the "ELVIS" part I first cleaned up the original using **GIMP**. I then manually traced it in **Inkscape** and saved it as a SVG file.

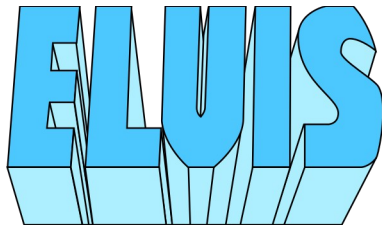


Figure 14. Elvis logo created in Inkscape.

For the "today" text I first cleaned it up using **GIMP** and also expanded the pink outer border 2 pixels using a second layer. I traced the image in **Inkscape** using the *Path > Trace Bitmap* tool. The options used for the scan was:

- **Multiple scans: Colors**
- **4** scans because we want our result to have 4 colors (Inkscape scans once for each color in the result)



Figure 15. Today lettering traced in Inkscape.

For the "BLOG" part I decided to test the **Adobe Illustrator** import function in **Inkscape** and

Can free software replace proprietary software for graphic production?

imported the graphic from volume 1 of the book. I cleaned it up in **Inkscape** (I ungrouped the graphic and removed the clipping path) and re-colored it in blue.

Cover layout using Scribus

The width of the spine depends on the number of pages in the book. Using the numbers obtained from the Blurb calculator (see 5.1.4. Step 3: Template construction) I created a new **Scribus** document.

Insert Image Frame / Get Image doesn't work for SVG files so I had to use **Import Vector File**. I got a message saying the "File contains some unsupported features." but ignored it so I could place and resize (using Ctrl + Shift) the graphics. I converted the imported colors (**Edit > Colors**) to CMYK which removed them from being out of gamut. I also defined new styles for the typography.

Author photograph

When the cover layout began to finalize, I knew the size of the photograph that was going to feature on the back. I edited the jpg-file in **GIMP** (see 3.1.2. Bitmap image editing). I made sure to keep the Adobe RGB (1998) ICC profile contained within when I opened it.

1. Cropping: No cropping done, I wanted to use the whole photograph.
2. Setting resolution (often 300 dpi for offset printing): I checked **Image > Print Size** and the resolution was already 300 dpi.
3. Adjusting size: I used **Image > Scale Image** so as not to change the resolution. I needed a width of 101 mm's and let the height adjust to this.
4. Adjusting levels (setting black and white point): **Colors > Levels**.
5. Adjusting curves and adjusting brightness and contrast: **Colors > Curves**.
6. Adjusting the color balance and adjust hue/saturation: **Colors > Hue-Saturation** (not needed).
7. Performing touch up; removing unwanted details: Not needed.
8. Masking: Not needed.
9. Sharpening: For **Filters > Enhance > Unsharp mask** I used the following values:
 - **Radius:** 1,2
 - **Amount:** 0,5
 - **Threshold:** 8
10. Separation: Not done in **GIMP**.

I finally saved it as a TIFF-file. I imported it into **Scribus** and cropped it vertically.

Can free software replace proprietary software for graphic production?

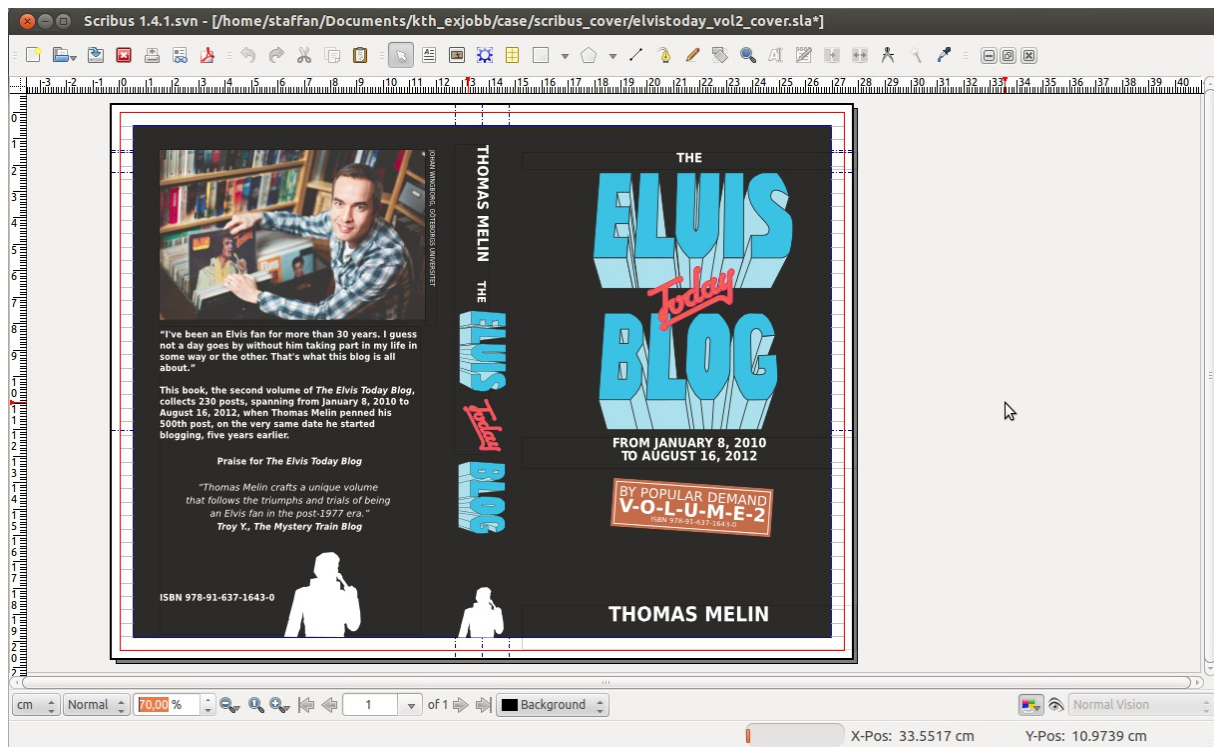


Figure 16. Final cover in Scribus.

5.1.7. Step 6: Proofing

I produced PDF documents in **Scribus** for on screen viewing (**File > Export > Save As PDF**) using the settings described in Appendix C, section 3.

All **Scribus** pages (except for the cover) was concatenated using **PDF Chain** (a GUI for **PDFTK**). I recommended that the author used **Xournal** for proofing the cover, and a text document for noting corrections to the pages. We met for a discussion of the proofing process and an opportunity for me to show him **Xournal**.

Receiving the proof

I received the result as two files: a PDF document produced by **Xournal** for the cover and a ODT document for the pages.

Can free software replace proprietary software for graphic production?

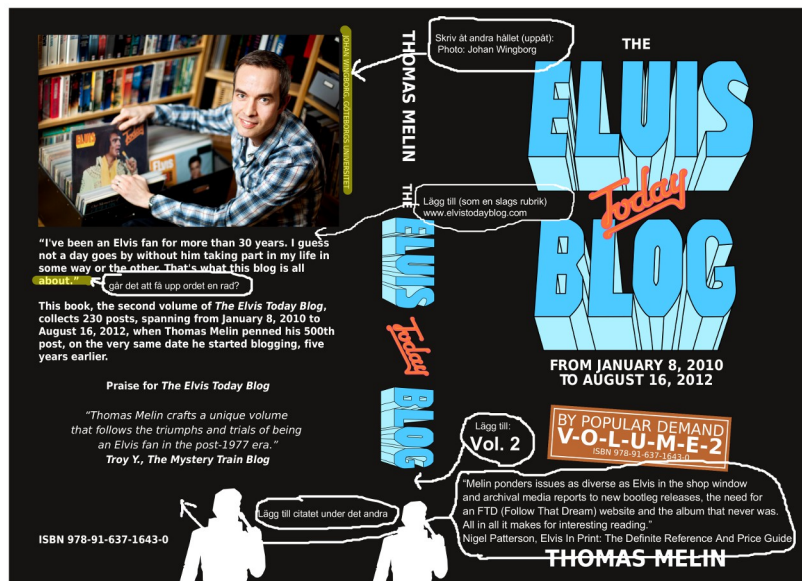


Figure 17. PDF created in Xournal.

Discussing the proof

The proof of the pages could roughly be categorized into typos and “bad” linebreaks. As bad linebreaks often are subjective, I met with the author to go through the proof.

Making the changes

Next I made all the changes, generated a new PDF document and sent the author another set of proof which he approved after a small number of corrections.

5.1.8. Step 7: Print preparation

Part 1: Print proofing

For the cover, I went through the list (see . Part 1: Print proofing). Next I activated the preflight verifier in **Scribus** (**Windows > Preflight verifier**) and selected the profile **PDF/X-3**. It reported “No problems found” [20, 40]. I went through the same process for the pages.

Part 2: PDF creation

As PDF creation is fully supported in **Scribus** this was only a question of producing the right kind of PDF documents that Blurb requested, PDF/X-3. The settings described in Appendix C, section 4, were used both for the cover and for the pages. I named it and clicked the **Save** button. The PDF documents for the pages were merged using **PDF Chain**.

A note about the **Use Color Profile** option (**Save as PDF > Color** tab). If **Use Color Profile** is not checked (and **Output intended for** is set to **Printer** in the **Save as PDF** dialog), every image will be converted to CMYK according to the color management setup. This increases most probably the size of the final PDF – CMYK uses 4 color channels (C, M, Y and K) as opposed to the three channels of RGB (R, G and B). If it is checked, images are not converted but the color profile embedded in the image will be embedded in the PDF too. This is why this option is checked and grayed out when creating a PDF/X-3.

Can free software replace proprietary software for graphic production?

Part 3: PDF quality control

I wanted to check the quality of the PDF documents using **Adobe Acrobat X Pro**. I ran the preflight check **Verify compliance with PDF/X-3** which for the cover gave me the result “No problems found.”

Checking the PDF of the pages with **Adobe Acrobat X Pro** told me that the document no longer was a PDF/X-3 formatted file. The merging had transformed the original formatting into PDF 1.4, making it impossible to upload the document to Blurb. No information on this behavior was found on **PDFTK**'s homepage (**PDF Chain** is only a GUI for **PDFTK**) so I contacted the author which confirmed that **PDFTK** didn't produce PDF/X-3 compliant documents. [78]

I began the search for another solution. The help came with **Ghostscript**'s ability to merge PDF documents [79]. While the result of this process still produced an error with the **Adobe Acrobat X Pro** preflight check (“Page boxes not nested properly”), the author of the part of **Ghostscript** which handles the conversion, doesn't consider this an PDF/X-3 validation error. [80]

Note. I also tried **PDFjam** (<http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic-research/firth/software/pdfjam/>) but the software didn't keep the PDF/X-3 format either.

Print office quality control

When uploaded to Blurb.com, the website does PDF preflight checks on the documents. They passed.

6. Case study: Magazine production

In 2010 I created a new graphic profile for the Swedish organization Critics of Hunting (including a new website made with the open source CMS **Drupal**). I used **Adobe Indesign** for the magazine (an important consideration for the choice of tool was that the regular layout person only knew **Indesign**). Because she didn't have the time to do #3 this year I did it using **Scribus**.

The magazine is in A5-size, b/w with a color cover. This issue was 40 pages. I followed the steps in this thesis with two exceptions:

1. The graphic template was re-created by running **Ubuntu/Scribus** in the virtualization software **VirtualBox** in a separate window on a **Windows 7** machine (which also ran **Indesign**). By observation of the template in **Indesign** I could transfer the template manually but quite easily to **Scribus** (resulting in 3 template pages, 29 paragraph styles and 5 character styles).
2. Proofing was done by producing a screen optimized PDF and sending it to the editor. He sent back his comments by e-mail and didn't use **Xournal**.

The final print ready PDF was PDF 1.5 and offset printing was used.

Can free software replace proprietary software for graphic production?

Åke Espegren: Djursafari i Sverige SID 5

JAKT DEBATT 3/2012

TIDSKRIFTEN SOM VÄRNAR OM DE VILDA DJUREN JAKT KRITIKERNA

RÄVEN
LIV OCH
FÖRFÖLJELSE
AV VÅRT
SYMBOLDJUR
SID 18

En äventyrare med empati SID 8

SEN SIST

Trots ejderns betydande minskning i svenska vatten väljer Landsbyggsdepartementet att låta jakten fortgå. Dess konsumtionsvärde är minimalt, dykänder är knappast goda att äta. Många ejdrar skadeskjuts och plågade försöker de då komma undan genom att dyka.

Sen sist

Hans Ryttman

... har föreningen haft sitt årsmöte. Jag meddelade att jag avgår som ordförande trots att valberedningen än en gång inte kunnat finna någon ersättare. Jag hoppas på så sätt att det blir större press på valberedningen att finna en ny ordförande till nästa årsmöte. Försök görs nu med att finna någon person både inom föreningen och bland sympatisörer. Jag är övertygad om att det snart kommer att ordna sig med en kandidat. Styrelsen verkar nu kollektivt med att sköta föreningens verksamhet vilket fungerar utmärkt

JAKTDEBATT 3/2012 3

Figure 18. Jaktdebatt final PDF made in Scribus.

Can free software replace proprietary software for graphic production?

7. Result and analysis

Can free software replace proprietary software for graphic production? By succeeding in creating print ready PDF:s for the two cases – using only libre software – and receiving the final printed products I have found that the answer to this question is yes.

7.1. Workflow

In this section I will describe and analyse the result and performance of the different steps of the workflow.

7.1.1. Step 0. Color management

The integrated color management in **Ubuntu** worked very well. The difference between screen and paper reproduction of colors was very small upon visual inspection.

7.1.2. Step 1: Manuscript preparation

The experience of working with text in **LibreOffice** was the same as working in Microsoft Office (or any other proprietary office suite).

No original graphics were produced in this step.

7.1.3. Step 2: Image preparation

This step was integrated into the 5th step, cover layout, as I didn't know the size of the graphics until the page (cover) was constructed. In practice one can view step 2 as being performed in parallel to step 5.

7.1.4. Step 3: Template construction

It was possible to create a template with the same ease in **Scribus** as in **Indesign** using grid lines and master (template) pages.

I found one thing lacking in **Scribus**. In **Indesign** it is possible to give a paragraph of text an “away from spine” adjustment property, very useful for page numbers and other template elements as it does not matter if the element is on a right or left page.

7.1.5. Step 4: Page layout

Page layout (and also cover layout) is about bringing all the pieces together and this means that the performance of the layout software is central to the workflow performance. As **Scribus** is the only available libre tool, the shortcomings are important as is the ways to handle them.

Problem: Text length

Scribus can't handle large documents. What is “large”? In my case edit operations (inserting and deleting characters) were slow, taking a couple of seconds to finish when the document reached

Can free software replace proprietary software for graphic production?

about 100 pages. This delay seems to increase exponentially so faster hardware or more memory is no real solution. In itself, this leads to problems with consistency, because I had to do changes to the template and text styles in several documents. I also had to find places to break up the book. In my case, a book with continuous text, this wasn't so easy. But the worst problem was the problems that arose when I had to change this break due to proof reading/corrections. I then had to move text between two **Scribus** documents until a new suitable page break was found.

I consider this a worst case scenario as many books often have natural breaks such as chapters.

Missing: Lists and bullets

Bullets and numbered lists are missing in **Scribus** [89]. I solved it by not using any bullets in the lists (meaning the technical limitations impacted on the graphic design). Bullets can be handled manually and seems to be an upcoming feature. [82]

Missing: Widow and orphan control

Functions for keeping paragraphs and lines together, keeping characters together and widow/orphan control are used to create good looking page breaks. They are missing in **Scribus** [91]. It now has to be done manually using Ctrl+Return (**Insert > Spaces & Breaks > Frame Break**) and by adjusting the letter-spacing. As this is left this to manual adjustment it takes time. On the other hand automatic handling of this still has to be manually checked. Some of this functionality is on the way [84].

Missing: Tables

Scribus has a primitive handling of tables: creating a grid of text boxes. For more complex tables, the **Scribus** wiki recommends using **OpenOffice** for constructing a table which is exported as a SVG document to be converted in **Inkscape** and later imported into **Scribus**. Feature is on the way [84].

Missing: Drop shadows

Drop-shadows are missing in **Scribus**, but that was not a problem in my two cases. The solution is to create the shadows in **GIMP** and importing the graphics into **Scribus**. [87]

Missing: Keyboard shortcuts for styles

It is not possible to assign keyboard shortcuts to styles in **Scribus**. This, in combination with the time consuming placement of style assignments ("palette"), slows down the workflow. Feature is on the way. [88]

Missing: Footnotes

Scribus doesn't handle footnotes. Feature is on the way. [89]

Missing: Palettes

The handling of object properties in **Scribus** is cumbersome. This is especially true for text properties, which is something a user frequently changes. A proper palette system would solve this. Feature is on the way. [84]

Bug: Importing text from LibreOffice to Scribus

There are several problems with the import of text into **Scribus** from ODT documents:

LibreOffice style names are not imported, the import of character styles does not work as well as several more [77, 92, 93, 94, 99]. This renders the import functionality useless.

Can free software replace proprietary software for graphic production?

7.1.6. Step 5: Cover layout

Inkscape (vector graphics)

The bitmap trace worked fine, resulting in a vector image better (closer to the original bitmap) than the one **Adobe Illustrator CS2** produced. One problem is that **Inkscape** works in RGB because this is what the SVG file format can handle. This is not really true. As of version 0.48 **Inkscape** can work with colors in the CMYK color space. This means that you can assign an object a CMYK color using a given color profile and it will import correctly into **Scribus**. [30]

GIMP (bitmap graphics)

GIMP performed very well and no problems were encountered.

LibreOffice SVG export (diagrams)

A strange artifact was discovered by **Adobe Acrobat Pro X**. I got several "Transparency used (transparency group)" warnings on every page of the magazine PDF. They seemed to be objects lying at the bleed border of the PDF. I discovered that they came from the three SVG diagrams placed inside of the magazine. I had constructed them from **LibreOffice Calc** diagrams copied and pasted into **LibreOffice Draw** and later saved as SVG documents. It turns out that the SVG documents produced by **LibreOffice** are not reliable and a final tidying-up step in **Inkscape** is required (or saving the diagrams as bitmap documents). [95, 96]

7.1.7. Step 6: Proofing

For proof reading comments on the book pages, a text document was used. For the cover **Xournal** was tried, making it easy to draw and type on the top of the PDF of the cover (annotation). When copying text from a **Xournal** generated PDF to **Scribus**, strange text encoding made copy-paste of corrections impossible. This is a known bug of **Xournal**. [90]

7.1.8. Step 7: Print preparation

All print ready PDF:s were finally tested with **Adobe Acrobat X Pro** without complaints. The book PDF:s were passed by the automatic quality control at Blurb. The magazine PDF:s were passed by a professional offset printer. Quality-wise, this means that free software can replace the proprietary alternatives.

One problem I encountered was that no user friendly tools for joining PDF/X-3 documents existed and I had to resort to command line operations. The joining operations were made necessary by the inability of **Scribus** to handle large documents.

7.2. Conclusions

What are the needs of a graphic designer and what should free software be able to accomplish?

The graphic production workflow was described in 7 steps (8 with color management setup) and this workflow was used in the two test cases. With the exception of step 2 (Image preparation) being performed later in the process (parallel to the layout steps), the workflow handled the cases well.

Can free software replace proprietary software for graphic production?

What free software tools can be used to accomplish this workflow?

Libre tools exist for all 7 steps, as proved by the successful production. The main software choices, the “open source creative suite” matching Adobes' proprietary creative suite are: **LibreOffice** for text editing, **GIMP** for bitmap editing, **Inkscape** for vector editing and **Scribus** for layout. All tools ran on **Ubuntu** with **GNOME** color management.

Are there any problems or drawbacks?

My case studies show several problems when you work with all of the programs simultaneously, with the main problem being import of styled text into **Scribus**. Integration is of course a question about file formats and import functionality, but also about how recognizable the workflow is, and how keyboard shortcuts and mouse control work. For example, zooming in and out has different keyboard/mouse controls in all three programs. To solve this, some kind of “open source creative suite” project is needed.

Vector and bitmap editing are the strong and trouble-free areas if you use **Inkscape** and **GIMP**. The import and text-handling functionality in **Scribus** leaves some things to be desired. Only unformatted text can be imported properly. That **Scribus** cannot handle large volumes of text is a problem with certain projects, both in the layout and PDF creation steps.

Other important features that are missing in **Scribus** are lists, footnotes, proper tables and widow and orphan control. They are all being worked on. But as with all open source software development it is depending on people voluntarily working on the code.

How efficient are the free tools? It took 32 hours to produce the book with **Scribus**, compared to volume 1, produced with **Indesign**, which took 20 hours. The additional time can be attributed in part to the fact that I have 10 years of experience with **Indesign**, whereas this was my first big project with **Scribus**. The new book is also 60 pages longer. Another time consuming step was the problem with large documents in **Scribus**. For the magazine, it took 3 hours to produce the template and 10 hours to produce the layout including photo editing. This is about the same time it would have taken to produce it with **Indesign**.

A final note on the argument that proprietary software is more expensive than free software, ie has a licensing fee. This is valid only as long as the free software don't lead to an increased production time and larger salary costs.

How can they be solved or overcome?

As the first case study showed all of the encountered obstacles can be overcome. The import problems are handled by importing styled text as unformatted text and applying the formatting in **Scribus**. The text length problem is handled by dividing long documents into several smaller. The solutions add time to the production, but this overhead is small compared to the total production time.

One of the three professionals consulted for the work with this thesis felt that the **Adobe Creative Suite** was a safe choice as the software was from one company and thus designed for compatibility [5]. This leads to the insight that it is important to communicate that open source is as future-proof as proprietary software and thus safe to use, and that open formats leads to compatibility. It also means that guides using libre solutions, written for graphic designers and layout people, should be produced. Knowledge is an important factor, as shows in my case studies

Can free software replace proprietary software for graphic production?

where I several times have to try different solutions before finding one that worked.

Conservatism and accumulated knowledge are two obstacles to change. Some effort has been made with this in mind, with guides and tutorials directed at graphic design professionals using Adobe products. [97, 98]

7.3. The future

Scribus is currently the only layout candidate and also the weak link in the production workflow. It must constantly be developed to be able to handle new PDF standards, but at the same time receive more features to be able to compete with **Adobe Indesign**.

Although the direct manipulation features of page layout software is something I think will survive, perhaps some parts of the future lies in another direction? CSS3 has support for the handling of pages [99], so maybe it's the web designers which will have to start writing CSS code for book production, until the layout tools do this for us. I am reminded of the way that people had to manually code web sites in HTML before the web development software and content management systems did that for us.

If CSS has the support for printer-ready book production, with the support of HTML to PDF software, it doesn't matter how this CSS and HTML code is produced. It can be graphic software as well as hand coding. It will also automatically create a standardized open file format for layout.

It also reminds of the "traditional" tools for typesetting, **TeX** and **LaTeX**, where you specify how the pages should look in a text file. They are widely used mainly in the scientific community. They existed before the desktop publishing revolution, so maybe we will in some aspect go back to where we started? This will certainly be an interesting area to research.

Can free software replace proprietary software for graphic production?

8. References

Printed in order of appearance for easier lookup.

1. *Why Open Source?*, <http://listman.redhat.com/about/whyopensource/>, December 16, 2012
2. Noyes, Katherine, *10 Reasons Open Source Is Good for Business*, http://www.pcworld.com/businesscenter/article/209891/10_reasons_open_source_is_good_for_business.html, December 16, 2012
3. Free Software Foundation, <http://www.fsf.org>, December 16, 2012
4. *Proprietary software*, http://en.wikipedia.org/wiki/Proprietary_software, December 16, 2012
5. Melin, Thomas, personal communication, August 4, 2012
6. Hellstrand, Vendela, personal communication, August 9, 2012
7. Ekström, Tomas, personal communication, August 4, 2012
8. Martin Howse, *Free Time in FLOSS + Art*, 2008, ISBN 978-1-906496-18-0
9. Pagola, Lila, *Applying F/LOSS as a final user and not dying in the attempt*, Libre Graphics Magazine 1.1 (page 43)
10. *Open Document Format for Office Applications (OpenDocument) v1.1*, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59302, January 19, 2013
11. *Scalable Vector Graphics (SVG)*, <http://www.w3.org/Graphics/SVG/>, January 10, 2013
12. Johansson, Kaj, *Grafisk kokbok*, Stockholm 2001, ISBN 9789178431618. Also available in English as *A Guide to Graphic Print Production*.
13. *Some notes about image format*, http://documentation.scribus.net/index.php/Working_With_Images#Some_Notes_About_Image_Formats, December 16, 2012
14. *Manual importing*, http://wiki.scribus.net/canvas/Help:Manual_Importing, December 16, 2012
15. Bruce Byfield, *Working with images in Scribus*, <http://www.linuxjournal.com/content/working-images-scribus>, December 16, 2012
16. Lilley, Chris, *Color Management in SVG2*, http://www.youtube.com/watch?v=e2J3k_S1eb0&list=PL095A40A9E5123EF6&index=3&feature=plpp_video (from Libre Graphics Meeting 2012), December 16, 2012
17. Lechner, Tom, *DTP Features*, <http://laidout.org/dtpcompare.html>, December 16, 2012

Can free software replace proprietary software for graphic production?

18. *Why should I use a human readable file format?*,
<http://stackoverflow.com/questions/568671/why-should-i-use-a-human-readable-file-format>,
December 16, 2012
19. *Pre-flight printing*, http://en.wikipedia.org/wiki/Pre-flight_%28printing%29, December 16, 2012
20. Leurs, Laurens, *PDF troubleshooter*, <http://www.prepressure.com/pdf/pdf-troubleshooter>,
December 16, 2012
21. *pxd/x3 export : apparently contains RGB images*,
<http://comments.gmane.org/gmane.comp.graphics.scribus/35449>, December 16, 2012
22. *Linux color management*, http://en.wikipedia.org/wiki/Linux_color_management, December 16, 2012
23. *Gamut*, <http://en.wikipedia.org/wiki/Gamut>, December 16, 2012
24. Melin, Staffan, *How to make a four page leaflet using Scribus*, 2010,
<http://oscillator.se/opensource>, December 16, 2012
25. *Surprise: Working 16-bit mode landing in GIMP!*, <http://www.gimpusers.com/news/00422-16-bit-goat-invasion-ready>, December 16, 2012
26. *Is the CinePaint project dead?*, <http://ubuntuforums.org/showthread.php?t=1866130>,
December 16, 2012
27. *13 Reasons to choose GIMP over Photoshop!*, <http://www.unixmen.com/13-reasons-to-choose-gimp-over-photoshop/>, December 16, 2012
28. *Comparison of raster graphics editors*,
http://en.wikipedia.org/wiki/Comparison_of_raster_graphics_editors, December 16, 2012
29. Miranda, Ramon, *Digital Painting with open source*, http://www.youtube.com/watch?v=nQDDxzAS_oE&list=PL095A40A9E5123EF6&index=10&feature=plpp_video, December 16, 2012
30. Prokoudine, Alexandre, *Getting CMYK colors from Inkscape to Scribus for printing*,
<http://libregraphicsworld.org/blog/entry/getting-cmyk-colors-from-inkscape-to-scribus>, December 16, 2012
31. *List of vector graphics editors*, http://en.wikipedia.org/wiki/List_of_vector_graphics_editors,
December 16, 2012
32. Andersson, Hans, personal communication, July 8, 2012
33. Andersson, Hans, *Pinball book*, <http://forums.scribus.net/index.php/topic,185.0.html>,
December 16, 2012

Can free software replace proprietary software for graphic production?

34. Laidout imposition software, <http://laidout.org/>, December 16, 2012

35. *Editing/Creating documents/Books*, <http://wiki.scribus.net/canvas/Editing#Books>, December 16, 2012

36. *Finally, real PDF annotating under Linux! (with help from Wine)*, http://www.gnurou.org/blog/2008/09/09/finally_real_pdf_annotating_under_linux, December 16, 2012

37. *Xournal: pdf annotator*, http://www.youtube.com/watch?v=WifbpKqS_-U&feature=related, December 16, 2012

38. *Xournal on Ubuntu Precise Pangolin 12.04*, <http://www.youtube.com/watch?v=YT8Qbqf5Tcc>, December 16, 2012

39. *List of PDF software/Viewers*, http://en.wikipedia.org/wiki/List_of_PDF_software#Viewers_3, December 16, 2012

40. *The Preflight Verifier*, http://documentation.scribus.net/index.php/The_Preflight_Verifier, December 16, 2012

41. *Scribus Output*, http://wiki.scribus.net/canvas/Scribus_Output, December 16, 2012

42. Healy, James, *Preflight in Ruby*, <http://yob.id.au/2011/07/29/preflight-in-ruby.html>, December 16, 2012

43. *Screen Calibration on Ubuntu 12.04 with Spyder2 [workaround]*, <http://nxadm.wordpress.com/2012/05/06/screen-calibration-on-ubuntu-12-04-with-spyder2-workaround/>, December 16, 2012

44. Prokoudine, Alexandre, *Richard Hughes on color management in Linux and GNOME*, <http://libregraphicsworld.org/blog/entry/richard-hughes-on-color-management-in-linux-and-gnome>, December 16, 2012

45. Prokoudine, Alexandre, *GNOME Color Manager, preliminary review*, <http://libregraphicsworld.org/blog/entry/gnome-color-manager-preliminary-review>, December 16, 2012

46. *GNOME color manager*, <http://projects.gnome.org/gnome-color-manager/>, December 16, 2012

47. *colord = color daemon*, <http://www.freedesktop.org/software/colord/index.html>, December 16, 2012

48. *Argyll Color Management System Home Page*, <http://www.argyllcms.com/>, December 16, 2012

49. Hughes, Richard, *ColorHug*, <http://youtu.be/GVH3ykJiNt0> (from Libre Graphics Meeting 2012), December 16, 2012

Can free software replace proprietary software for graphic production?

50. *ColorHug*, <http://www.hughski.com/>, December 16, 2012

51. Lonien, Wolfgang, *Hughski ColorHug field report, or “test”, or “review”*, <http://wolfgang.lonien.de/2012/05/hughski-colorhug-field-report-or-test-or-review/>, December 16, 2012

52. Willis, Nathan, *Sizing up the ColorHug open source colorimeter*, <https://lwn.net/Articles/499231/>, December 16, 2012

53. Hughes, Richard, *ColorHug: filling the F/LOSS colour calibration void*, Libre Graphics Magazine 1.4 (page 43)

54. *Calibrating a Monitor in Linux*, http://www.brighthouse.com/computing/linux/articles/75216.aspx?cid=parsely_rec, December 16, 2012

55. *Monitor Calibration: Linux Tips*, <http://www.brighthouse.com/computing/linux/articles/110682.aspx>, December 16, 2012

56. *Soft Proofing*, <http://www.flickr.com/groups/gimpusers/discuss/72157624240896808/>, December 16, 2012

57. *Gamut warning huge compared to Photoshop*, <http://gimpforums.com/thread-gamut-warning-huge-compared-to-photoshop>, December 16, 2012

58. *1. Display an image — the Color Management Settings dialog*, <http://ninedegreesbelow.com/photography/conversion-intent-settings.html#1>, December 16, 2012

59. *About CMYK color and Gimp*, https://wiki.archlinux.org/index.php/CMYK_support_in_The_GIMP#About_CMYK_color_and_Gimp, December 16, 2012

60. *GIMP 2.4 Color Management*, <http://www.gimp.org/release-notes/gimp-2.4-cm.html>, December 16, 2012

61. *GIMP Soft proof mode*, <http://www.gimptalk.com/index.php?/topic/33122-soft-proof-mode/>, December 16, 2012

62. *Chapter 11. Color Management with GIMP*, <http://docs.gimp.org/en/gimp-imaging-color-management.html>, December 16, 2012

63. Section 1.14.1. of the GIMP built-in help system

64. Willis, Nathan, *Quality Printing with GIMP*, <http://blog.worldlabel.com/2010/quality-printing-with-gimp.html>, December 16, 2012

65. *Preparing a document for printer: CMYK PDF*, <http://blog.tsoots.fi/2009/02/preparing-document-for-printer-cmyk-pdf.html>, December 16, 2012

Can free software replace proprietary software for graphic production?

66. Crossland, Dave, *Pierre Marchand talks Fontmatrix, responsiveness and user engagement*, Libre Graphics Magazine 1.1 (page 21)
67. *The top 10 open source web fonts*, <http://www.netmagazine.com/features/top-10-open-source-web-fonts>, December 16, 2012
68. *7 Best Linux Distributions for Multimedia Enthusiasts*, <http://www.junauza.com/2008/08/7-best-linux-distributions-for.html>, December 16, 2012
69. *What white point temperature should I set my LCD monitor to?*, <http://photo.stackexchange.com/questions/4611/what-white-point-temperature-should-i-set-my-lcd-monitor-to>, December 16, 2012
70. *Color temperature/Desktop publishing*, http://en.wikipedia.org/wiki/Color_temperature#Desktop_publishing, December 16, 2012
71. *Where does Gnome Color Manager store the ICM or ICC files?*, <http://askubuntu.com/questions/155552/where-does-gnome-color-manager-store-the-icm-or-icc-files>, December 16, 2012
72. *Blurb Color Management Resource Center*, <http://www.blurb.com/b3info>, December 16, 2012
73. *DejaVu fonts*, <http://dejavu-fonts.org>, December 16, 2012
74. *Blurb booksize calculator*, http://www.blurb.com/make/pdf_to_book/booksize_calculator (only available to registered users), December 16, 2012
75. *Cannot embed a font into a pdf document*, <http://blog.tsoots.fi/2009/02/cannot-embed-font-into-pdf-document.html>, December 16, 2012
76. *Scribus and OpenOffice.org/OpenDocument*, http://wiki.scribus.net/canvas/Help:Manual_Importodt, December 16, 2012
77. *Scribus bug report on character styles ignored/not-imported*, <http://bugs.scribus.net/view.php?id=9858>, December 16, 2012
78. Steward, Sid, personal communication on Pdftk, November 6, 2012
79. *Creating a PDF/X-3 document*, <http://ghostscript.com/doc/9.05/Ps2pdf.htm#PDFX>, December 16, 2012
80. *pdfwrite - check annotation placement and Box sizes for PDF/X-3 validity*, <http://git.ghostscript.com/?p=ghostpdl.git;a=commit;h=7167ebccea78be8dcec24a059f936a861769cd5>, December 16, 2012
81. *How to create a list in Scribus*, http://wiki.scribus.net/canvas/How_to_create_lists_in_Scribus, December 16, 2012

Can free software replace proprietary software for graphic production?

82. Bullets in Scribus, <http://www.scribus-ece.info/News/BULLETS!!!/>, December 16, 2012

83. Feature request for Scribus: Support non-breakable text blocks, <http://bugs.scribus.net/view.php?id=1155>, December 16, 2012

84. New features and fixes in Scribus 1.6, <http://wiki.scribus.net/canvas/1.6-new-features-and-fixes>, December 16, 2012

85. Scribus bug report: Impossible selection with mouse through multiple linked frames, <http://bugs.scribus.net/view.php?id=6241>, December 16, 2012

86. Scribus bug report: Import pages with linked text frames will break the link that acrosses the pages, <http://bugs.scribus.net/view.php?id=8056>, December 16, 2012

87. How to create drop shadows in Scribus, http://wiki.scribus.net/canvas/Drop_Shadows, December 16, 2012

88. Scribus bug report: Make shortcuts for styles working, <http://bugs.scribus.net/view.php?id=10289>, December 16, 2012

89. Scribus patch: footnotes, <http://bugs.scribus.net/view.php?id=11032>, December 16, 2012

90. Xournal bug report: In PDF export, fonts do not appear on Apple Preview, http://sourceforge.net/tracker/index.php?func=detail&aid=3477005&group_id=163434&atid=827733, December 16, 2012

91. Scribus bug report: Import of ODT should use "display name" of style, <http://bugs.scribus.net/view.php?id=11157>, December 16, 2012

92. Scribus bug report: Lost bold text in text imported from ODT file, <http://bugs.scribus.net/view.php?id=9080>, December 16, 2012

93. Scribus bug report: Some styles imported from OpenOffice loose their names, <http://bugs.scribus.net/view.php?id=6573>, December 16, 2012

94. Scribus bug report: Some style attributes are lost when importing from OpenOffice text, <http://bugs.scribus.net/view.php?id=6574>, December 16, 2012

95. *Transparency used (transparency group)" warnings*, <http://forums.scribus.net/index.php/topic,720.0.html>, December 16, 2012

96. Scribus documentation page on tables, <http://documentation.scribus.net/index.php/Tables>, December 16, 2012

97. *Inkscape for Adobe Illustrator Users*, <http://www.angelfire.com/mi/kevincharles/inkscape/pa-illustrator.html>, December 16, 2012

98. *FLOSSMETRICS/OpenTTT guides*, http://guide.conecta.it/index.php/Main_Page, December 16,

Can free software replace proprietary software for graphic production?

2012

99. McKesson, Nellie, *Building books with CSS3*, <http://www.alistapart.com/articles/building-books-with-css3/>, January 19, 2013

Can free software replace proprietary software for graphic production?

Appendices

Appendix A. Usage statistics

How many people use open source software? How many people use proprietary software? I have pursued several different paths to try and find this information but have failed.

The numbers doesn't exist but there are maybe some ways to get an approximate:

- Downloads at a host (eg sourceforge for Scribus), but as the software can be downloaded in other ways (eg Synaptic) this is not reliable
- Facebook likes, which will give an indication on the ratio but not the absolute numbers. Scribus has 772 likes, while Indesign has 263 282 (as of the 24th of November 2012).
- Analysing data from Popcon, the popularity-contest package, for different linux distributions (eg <http://popcon.ubuntu.com/>).

I tried to contact both Canonical (the company that supports Ubuntu) and Adobe but failed to receive any numbers. The only "solid" numbers I have found is Ubuntu's claim that is has "more than 20 million people preferring it to commercial alternatives" while there are over 1 billion Windows users. While this doesn't say anything about what software they are running on those platforms (and a lot of libre software is available for Windows), it does give some indication as to what path the users are threading.

I have been in personal contact with Carlo Daffara (<http://carlodaffara.conecta.it/about/>), an italian open source consultant that has worked for the EU amongst others. He told me about reports that say that 1 to 1.8% of desktop users run linux. And that Popcon for Debian reports that 35% have installed GIMP.

The statistics at NetApplications show a 90 to 1 relation between the Windows and Linux platforms for desktop use.

Sources and additional reading:

- <http://www.ubuntu.com/> "Ubuntu is the world's favorite free operating system, with more than 20 million people prefering it to commercial alternatives." (November 7, 2012)
- <http://winsupersite.com/windows-8/windows-charge-few-numbers>
- <http://marketshare.hitslink.com/>
- http://en.wikipedia.org/wiki/Usage_share_of_operating_systems

Can free software replace proprietary software for graphic production?

Appendix B. Rules for entering and editing text

Typical rules looks like this:

- Never style the document using different fonts, font sizes and bold, italic etc.
- Never try to layout the document using spaces (or tabs). If creating columns (for tables), use tabs, and only input one tab for each column.
- Use only one space after periods.
- Only hit the return or enter key once after each paragraph. Don't try to layout the document using blank lines.
- Don't insert graphics or images into the document. Make a note in the document where the image should go (“[image introduction.jpg]”).
- Never use the standard “automatic” formatting tools such as drop caps, lists, footnotes, endnotes, and indexes. Make notes in the document instead.

Sources and additional reading:

- <http://www.peachpit.com/articles/article.aspx?p=1352550&seqNum=2>
- Hellstrand, Vendela, personal communication, August 9, 2012

Can free software replace proprietary software for graphic production?

Appendix C. Settings

1. Settings for Scribus color management

- **RGB Images:** sRGB IEC61966-2.1
- **CMYK Images:** Blurb_ICC_Profile
- **RGB Solid Colors:** sRGB IEC61966-2.1
- **CMYK Solid Colors:** Blurb_ICC_Profile
- **Monitor:** I selected the monitor profile obtained in the previous step.
- **Printer:** Blurb_ICC_Profile
- **Rendering intents**
 - **Images: Perceptual**
 - **Solid colors: Relative Colorimetric.**
- **Simulate printer on the screen:** checked
- **Convert all colors to printer space:** checked
- **Mark colors out of gamut:** checked
- **Use blackpoint compensation:** checked

2. Scribus setup

Settings for a new document (**File > New**):

- **Document layout: Double sided**
- **Size: Custom**
- **Orientation: Portrait**
- **Width:** 127mm
- **Height:** 203,21mm
- **First page is: Right page**

Can free software replace proprietary software for graphic production?

- **Number of pages:** 1
- **Default unit:** Millimeters
- **Automatic text frames:** Unchecked
- **Show document settings after creation:** Unchecked
- **Margin guides:**
 - **Inside:** 18mm
 - **Outside:** 12mm
 - **Top:** 15mm
 - **Bottom:** 14mm
- **Bleeds:** 3,17 mm

Scribus document setup (**File > Document setup**):

- **Hyphenator**
 - **Language:** English
- **Fonts**
 - **DejaVu** fonts: **Subset** unchecked

This last step is necessary to be able to embed the fonts into the final PDF. Note that “subset” in **Scribus** really means outlining (see *Cannot embed a font into a pdf document*, <http://blog.tsoots.fi/2009/02/cannot-embed-font-into-pdf-document.html>, December 16, 2012).

3. Scribus export settings

- **Output one file for each page:** unchecked.
- **General** tab: **All pages, PDF 1 .4, Maximum image resolution:** 72 dpi.
- **Fonts** tab: **Fonts to embed:** <all fonts>.
- **Extras** tab: **Enable presentation effects** unchecked.
- **Viewers** tab: Select **Single Page** and **Use viewers default**.
- **Security** tab: unchecked.

Can free software replace proprietary software for graphic production?

- **Color** tab: **Output intended for: Screen/web**.
- **Pre-Press** tab: no printer marks and no bleed (0 mm).

4. Scribus print PDF settings

- **Output one file for each page**: unchecked
- **General** tab: **All pages**, **Compatibility: PDF/X:3**, **Compress text and vector graphics** checked.
- **Fonts** tab: **Embed all**.
- **Extras** tab: **Don' t use presentation effects**.
- **Viewers** tab: **Single Page** and **Use viewers default**.
- **Security** tab: Don' t use security.
- **Color** tab: **Output intended for: Printer** is already checked (and grayed out), **Convert Spot Colors to Process Colors** checked.
- **Pre-Press** tab: **No Printer Marks**; **Use Document Bleeds** checked; **Output profile**: Blurb_ICC_Profile.icc; **Info string**: name of Scribus document.

Can free software replace proprietary software for graphic production?

Appendix D. Style definitions in Scribus

I used the following styles in the first case (the book):

- **Heading 2:** Blog entry title
- **date:** date line, eg “Thursday, August 16, 2007”.
- **Text body:** body text, first line is indented
- **Text body - first paragraph:** first paragraph of body text (not indented, but with a blank line above it).
- **subheading:** Subheading.
- **Text body - after subheading:** As “Text body - first paragraph” but without the blank line above.
- **quote:** for quoted paragraphs, first line is indented. Use “title in quote style” for titles in quotes.
- **quote - first paragraph:** first paragraph of quote (not indented)
- **quote - song line:** single line quote
- **quote - song line first p:** single line quote with one blank line above
- **list:** for lists of items that are several lines long. One blank line between/before each
- **list - titles :** for lists of items that are are single line
- **list - titles first p:** for lists of items that fit on a single line. A blank line before the first (and last, but that is automatically taken care of since this is always followed by the **Text body - first paragraph** style).
- **header:** page header (ie title of book)
- **footer:** page footer (ie page number)

Character styles:

- **title:** Use the title style for titles.
- **title in quote:** Use the title style for titles mentioned in quotes.
- **strike out:** For things that are striked out.
- **title - strike out:** For titles that are striked out.

Can free software replace proprietary software for graphic production?

I brought up the **Style Manager** by choosing **Edit > Styles** and redefined the two default styles:

- **Default character style:** DejaVu Serif, Condensed, 10 pt, Language: English
- **Default paragraph style:** Distances and linespace: Align to baseline grid. Character style: Default character style.

In Scribus, I created the following paragraph styles:

- **Text body:** Based on: Default Paragraph Style, Alignment: Left justify (fourth icon), First Line Indent: 12 pt (same as the line space).
- **Text body - first paragraph:** Based on: Text body, First Line Indent: 0 pt, Space before: 12 pt.
- **Heading 2:** Based on: Text body, Space before: 24 pt, Align: Left. Character style: Based on: Default character style, Family: DejaVu Sans, Style: Condensed Bold, Font size: 12 pt.
- **date:** Based on: Text body, First Line Indent: 0 pt, Align: Left. Character style: Based on: Condensed italic.
- **header:** Based on: Default Paragraph Style, Distances and Alignment: Fixed Linespacing, 12pt, Align: Left. Character style: Based on Default character style, DejaVu Sans, Book, All caps.
- **footer:** Based on: Default paragraph style, Distances and Alignment: Fixed Linespacing, 12pt, Align: Center. Character style: Based on: Default character style, DejaVu Sans, Book.
- **quote:** Based on: Text body, Align: Left, First line indent: 12 pt. Character style: Condensed Italic.
- **quote - first paragraph:** Based on: quote, First line indent: 0 pt.
- **list:** Based on: Text body, Align: Left , Left indent + Right indent + Space before: 12 pt.
- **list - titles:** Based on: list, Space before: 0pt
- **list - titles first p:** Based on: list - titles, Space before: 12 pt
- **quote - song line:** Based on: quote; Left indent: 0 pt
- **quote - song line first p:** Based on: quote - song line firsts p, Space before: 12 pt.
- **subheading:** based on Default paragraph style, Space above: 12pt. Character style: Default character style, DejaVu Sans, Condensed Bold
- **Text body - after subheading:** based of Text body - first paragraph, Space above: 0pt.

Finally, I created the character styles:

Can free software replace proprietary software for graphic production?

- **title**: Based on: Default Character Style, Family: DejaVu Serif, Style: Condensed Italic.
- **title in quotes**: Based on: Default character style, Style: Condensed.
- **strike out**: Based on Default Character Style; Strike out attribute
- **title - strike out**: Based on title; Strike out attribute