MetaPost
how we adapt
ConTeXt Meeting 2011
Development Stage

I. We started with simple usage (logos) and PostScript output.

II. Then we moved on to conversion to PDF using TeX macro solution.

III. This has the advantage that fonts are handled by TeX.

IV. For a long time this was a generic solution (later became the MkII variant).
Development Stage

I. We added some extensions (transparency, CMYK, etc.) and MetaFun showed up.

II. That extension mechanism uses special colors as signals.

III. We always collected btext ... etex in order to speed up processing.

IV. In addition, we added textext and similar features.

V. Communication between MetaFun and ConTeXt became more advanced over time.
3

Development Stage

I
when LuaTEX showed up a substitution based lua converter was written

II
later when lpeg came around an experimental lpeg converter showed up

III
some changes were made to textext processing and run management
Development Stage

I. The arrival of mplib had rather big consequences.
II. Integration of MetaFun became less of a runtime burden.
III. A more definitive lpeg converter was written.
IV. Text handling was kept internal (but still needs two passes).
Development Stage

I. Some extensions were changed to use the new pre/postscripts functionality.

II. The lpeg converter was adapted accordingly.

III. Multiple (independent) MetaPost instances were now supported.

IV. The chemical code was overhauled and moved to the core.
Development Stage

I by now all extensions use pre/postscripts

II this made it easier to add more extend functionality

III again the lpeg converter was adapted (simplified)

IV it became possible to do some color trickery with text

V but (till now) color spaces are more complex (mp has a mixed model)

VI so we might move towards a slightly different approach

VII a couple of helpers were added for Mojca (some more will follow)
Development Stage

I. We're now splitting the code in MpII and MpIV code bases.

II. From now on the focus will be on MpIV.

III. The (rather old) MetaFun code will be cleaned up.

IV. Where possible namespaces will be added.

V. As MetaFun is loaded runtime we see an impact on startup time (quite some files).

VI. So maybe we will use a packed and/or zipped pseudo format file for faster loading.
Development Stage

I. An overhaul of the flowchart code is on the agenda.
II. We also want to finish (and cleanup) the chemical related code.
III. Simple data/graphics helpers will be provided (graph replacement).
IV. And of course we keep moving on (who knows what MetaPost 2 will bring us).
V. Some examples: tests/mkiv/metapost/plugins-*