

DevDays User Test

Conclusions

Summary

This report contains the results from the documentation user testing performed at DevDays in Munich 09. The test was very well received and completed by close to 15% of the DevDays attendees. To a certain extent, this should be representative for how the DevDays attendees think about documentation and how they prefer to learn.

The conclusion is based on the test feedback is not surprisingly quite similar to the feedback given on the blog posted earlier this fall. The users want a design quite similar to the current one. In addition they want a fresher looking layout, using colors and icons. Navigation features like bread crumbs and an Ajax based search suggestion function, is also rated high. As good as all social concepts have been rejected.

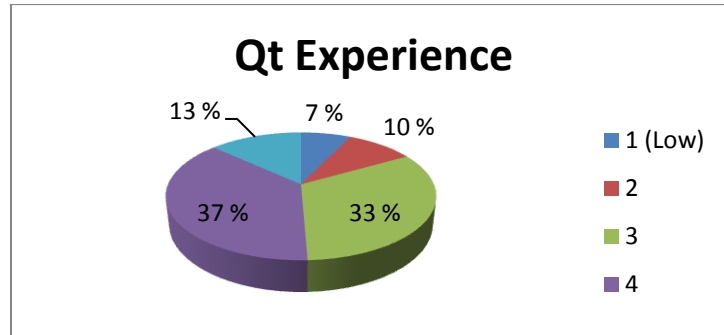
In addition to the goals this feedback generates, a number of other goals appear as a consequence. The structure of the documentation must be updated and the content, layout and functionality of the documentation must be split up into separate parts, keeping the content in a HTML format while all layout, is pushed into a CSS document. As far as any scripts will be used, these will have to be included in a JS file included in the page at page load. This will to a certain extent require some changes in QDoc.

I am currently working on mapping requirements and features to the new design. I'll put down suggestions for some short term goals and some long term goals. I want to get these done before I send them, so this document only contain the results of the test, plus some interpretation of these.

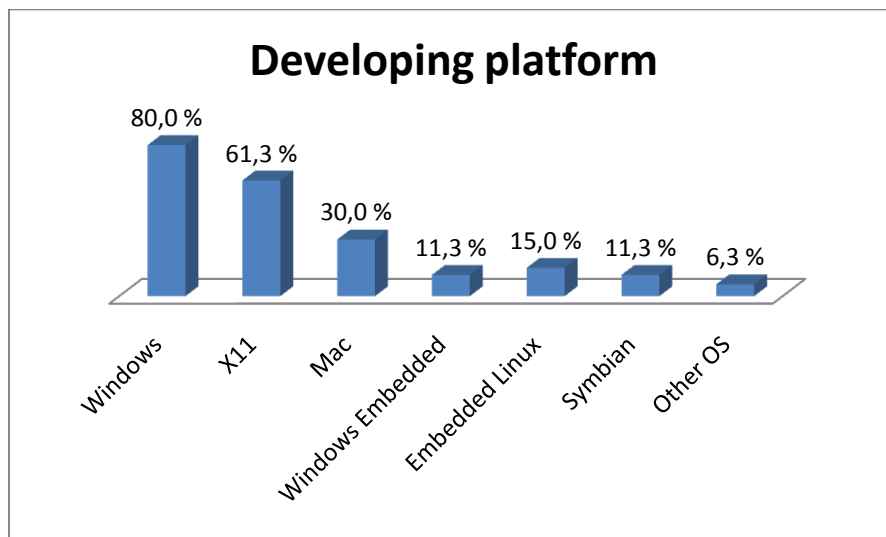
Morten;

Test User Profiles

The average user completing the survey mainly works as a programmer and/or designer. On a scale from 1 to 5, where one is the lowest and five is the highest, the majority (70%) state that their knowledge about Qt is somewhere between 3 and 4. The vast majority (85%) of this group uses the documentation every day, and almost 30 % do lookups at least once an hour.

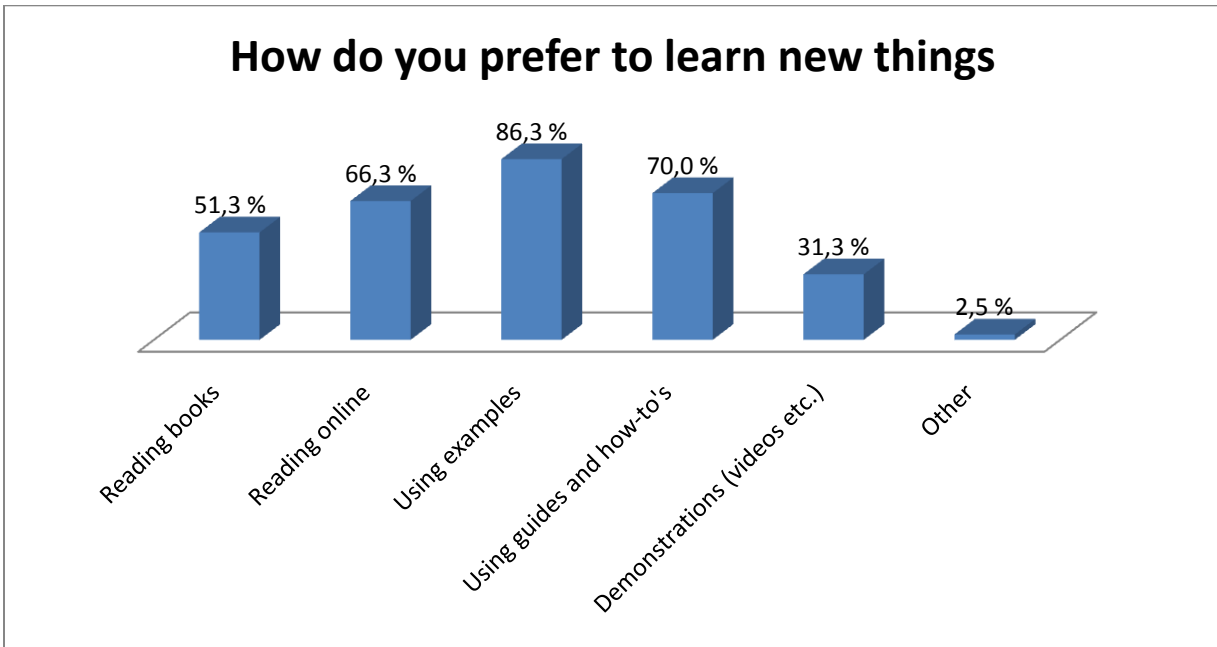


Further the users mainly develop for Windows and X11, while only a third develops for Mac. On the embedded side the degree of popularity amongst the developers are quite even and often the developers are building for more than one of the embedded platforms.



Learning patterns

The preferred way to learn amongst the users is by looking at code examples and “how-to” instructions. Further the test shows that the users prefer to use reference lists and keyword searches when looking for specific classes, functions and function specific information. On the other hand, they like to browse and read about the subject when looking for details on modules and key technologies. This is also the case when they are looking for new and popular features and solving problems they encounter.



Design rating and feedback.

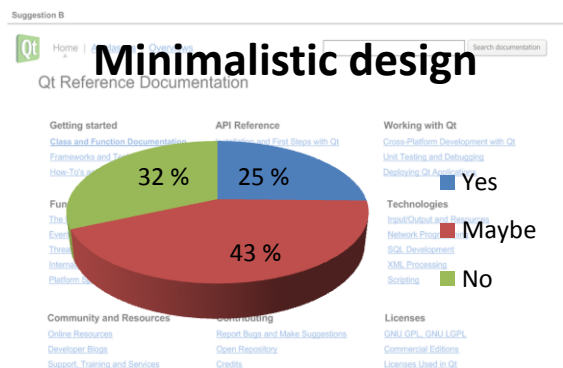
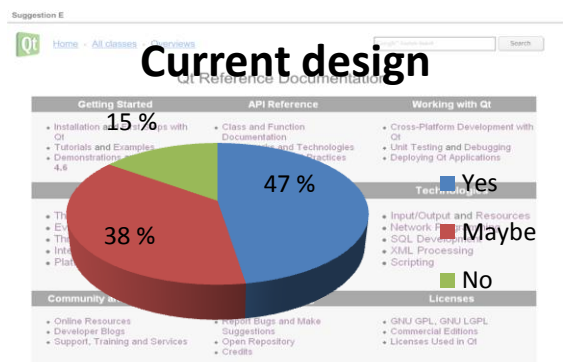
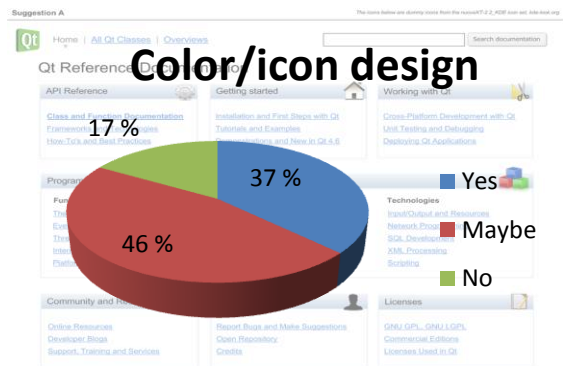
Design concepts

The users were presented with a set of design concepts which they could comment and rate(Yes,Maybe,No). The concepts stretched from super-minimalistic to super-social, with the current design in the middle. The motive behind this was to get a clearer feedback on what kind of documentation concept our users want. From reading the feedback comments on the different concept this strategy seems to have worked as hoped. When reading comments from users that rated “Maybe” on a concept, it is often easy to see what the user prefers and not.

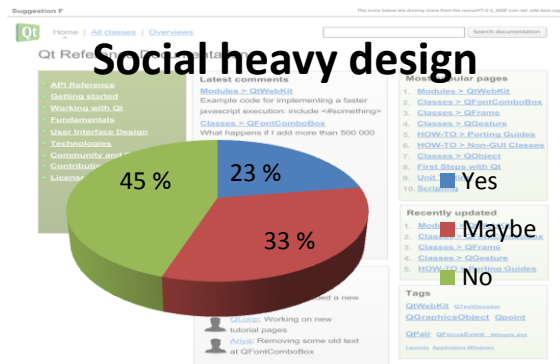
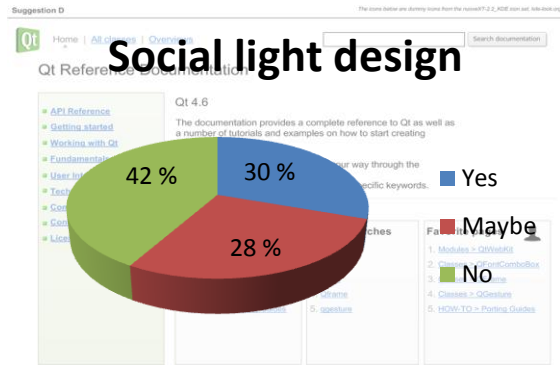
The first three concepts are quite similar displaying a set of categorized link menus. They are also the concepts that are closest to the current design.

The difference between them is the way they use colors, icons and visual structures like borders and bullet points. These were not surprisingly the best received concepts. The reason for this is the similarity to the concept that the users already are familiar with, and if it is one thing people usually don't like, it is changes. Still the current design was criticized as boring and badly categorized.

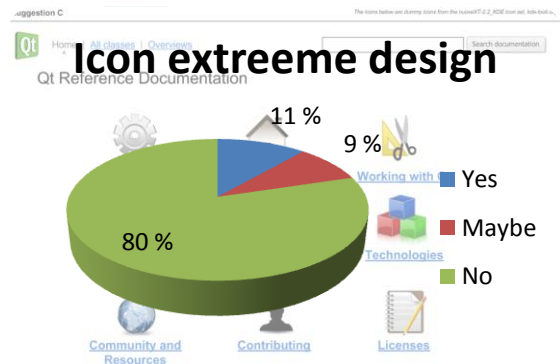
The minimalistic version was said to be too simple and had a lack of structure. The design including icons and a broader color palette was said to be more fresh, but in need of better categorization.



The next two designs display a more social inspired way of organizing the documentation. The feedback on these concepts was that many of the elements did not belong in such a page. The general message was that these kinds of elements would fit on a developer site like the coming Qt Developer Network pages. Still features like a list of previously browsed pages, was well received.

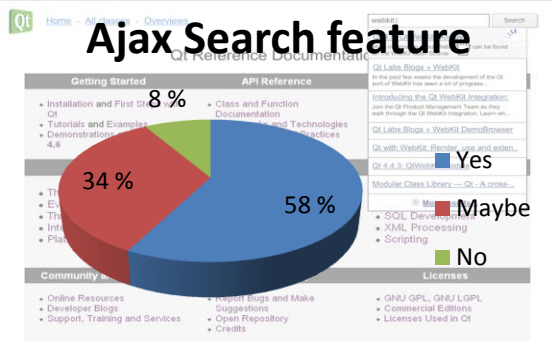


The most minimalistic design was totally rejected by most of the users. Even the users voting in favor of such a design argued against it in their comments. This was not unexpected either, but the concept was included in the test to show an extreme minimalistic design.



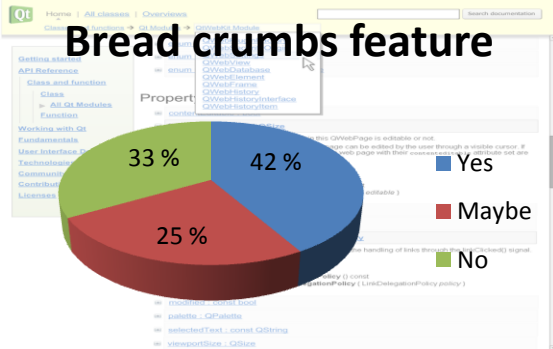
Design features

Suggestion E-A - AjaxSearch

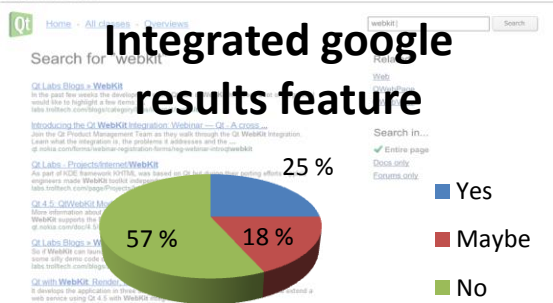


The remaining design concepts mainly demonstrate features that could be included in any of the main concepts. The first one displays an Ajax based Google search, based on a custom search engine, going through the Qt documentation and displaying suggestions related to the keywords as they are typed into the search box. This feature was very well received by most of the users. A similar feature was demonstrated in combination with a bread crumb line enabling the user to navigate directly to other documentation pages on the same level, without going up to the parent level. Whereas the bread crumb line was very well received, the users were more skeptical to the Ajax menu. If this feature was to be used, it would have to be more intuitive.

Suggestion D -> A



Suggestion E-B - Search



The second last feature was simply to integrate search results into the Qt documentation page. This was surprisingly unwell received. The users argued that this was a bad mix, and that they would prefer to go to the Google page when searching. The last feature included a dropdown menu displaying a selection of links, as the one on the current front page. This feature was also turned down by the users as uninteresting and a bit too much.



Conclusions and targets

These results give us quite clear advice in which direction we should take our documentation design. For the top-level index page, the number shows that our users want a design which is close to the one Qt currently has, only freshen it up with color and to some extent discrete icons. In addition the categories and listed links must be chosen carefully. They must cover all the different sides of Qt and at the same time not grow to a number that will make the front page chaotic. On the feature side, we should include some kind of auto-suggest search box in the design, as well as bread crumbs for easier navigation.

On pages on the second $+n$ level, we should look into a number of issues. First, we should look into navigational features. One example could be to include a navigational menu on the left side. Most of the pages follow a structure that easily be included in such a menu, removing such links from the content. This will also leave space underneath the menu for features like history, related sites etc. The users did not like the popup menu on the bread crumb line, so navigation to classes in the same module as the one the user is browsing could be put in such a box. Still the number of boxes should be limited to avoid overfilling the page.

The next part will contain a suggestion on how we can design the documentation differently based on the feedback from the test and from the feedback given in the Labs blog "Giving the doc a facelift" posted September 28, 2009.

Design ideas:

The new design should have a simple style using a limited set of colors and icons. Text and images should follow a streamlined style to make the design clean and the different elements easy to recognize. Common structural elements like borders, tables, lists and figures should also follow a set of design rules.

The documentation can be divided into the following groups of pages:

Page type	Content/Structure	Requirements
All pages	<ul style="list-style-type: none">- Header including list of top links, logo and search box.- Footer including version, Nokia Copyright statement and feedback box.- All pages should have a menu and/or a bread crumb line for navigation.	<ul style="list-style-type: none">- Logo and top links appear on a straight line in the top left corner. Search box in the opposite corner. Search box should only search one version of the docs.- Footer should appear as the current one. Feedback box should appear on all pages with the exception of the front page.
Overview pages (i.e. front page, list of classes, functions)	<ul style="list-style-type: none">- Categorized links	<ul style="list-style-type: none">- Categories shaped as boxes with a thick top border showing the category name and an icon.- Lists of links. No underline, but color changing when hovering.
Theme/ Subject pages (i.e. How-to, examples and modules)	<ul style="list-style-type: none">- Introduction- Table of contents- Paragraphs handling different subjects.- Figures/Charts/Images- Code snippets	<ul style="list-style-type: none">- Introduction and paragraphs are quite similar and should appear with a header. The width of the text should have a minimum and maximum width.- The table of content should appear as a list of underlined links, visualizing the hierarchal structure of the page.- Figures and charts should follow a set of rules regarding use of shapes, colors and elements.- Images should appear with no border, centered and with image text. Their size should be restrained to fit the text width. If needed they could be clickable to see the image in a larger version.- Code snippets should appear on a gray background and have a slightly different font and font style.
Reference pages (i.e. Class reference)	<ul style="list-style-type: none">- Short introduction- Lists of properties, functions etc.- Full description- Figures/Charts/Images- Code snippets- Property, function details.	<ul style="list-style-type: none">- The short introduction should contain the first line of the full description. The width of the text should have a minimum and maximum width.- Figures and charts should follow a set of rules regarding use of shapes, colors and elements.- Images should appear with no border, centered and with image text. Their size should be restrained to fit the text width. If needed they could be clickable to see the image in a larger version.- Code snippets should appear on a gray background and have a slightly different font and font style.- Property, function details etc. should be short and to the point. It should also include a link to the top of the page, in addition to other relevant documentation.