

Дизайн в QlikView для не дизайнеров

Содержание

| | |
|--|----|
| Дизайн в QlikView для не дизайнеров | 1 |
| Введение | 1 |
| Правило пяти П | 3 |
| “Плоский” дизайн (Flat design) | 5 |
| Обзор | 5 |
| Рекомендации по реализации “плоского” дизайна в QlikView | 6 |
| Еще несколько полезных ссылок | 12 |
| Итоги | 12 |
| Бонус для тех, кто дочитал до конца :-) | 12 |
| Пасхалка (Easter egg) от разработчиков Qlik Sense /theme/highvis | 12 |

Введение

От разработчика QlikView могут ожидать навыков дизайнера, которых у него нет. А дополнительно оплачивать работу профессионального дизайнера захотят единицы. Вот и приходится аналитикам, которые раньше работали с базами данных и обычно делали стационарные отчеты, с переходом на QlikView получать новые навыки. В этой статье я постараюсь помочь не дизайнерам освоить принципы и приемы создания красивых и удобных приложений QlikView.

Цепочка разработки состоит из многих шагов: Источники – Модель – Формулы – Диаграммы и таблицы – Сверка показателей – Удобный и красивый Интерфейс. Последний этап обычно опускается для экономии времени и бюджета, либо выполняется в меру навыков разработчиков и настойчивости требований заказчиков. Разработчики с небольшим опытом выведут необходимые диаграммы и таблицы на экран и расположат их так, чтобы поместилось как можно больше (Рис. 1). В принципе это позволит пользователям проводить необходимый анализ данных. Но часто возникают проблемы с запутанной навигацией, переизбытком ярких цветов и выделением важной информации.



Рис. 1 Приложение новичка

Опытные разработчики QlikView учтут базовые требования к дизайну, направление анализа от общего к частному и создадут дашборд на первом листе. Скорее всего, они будут придерживаться корпоративных цветов. Вот как опытный разработчик может переделать приложение новичка (Рис. 2):

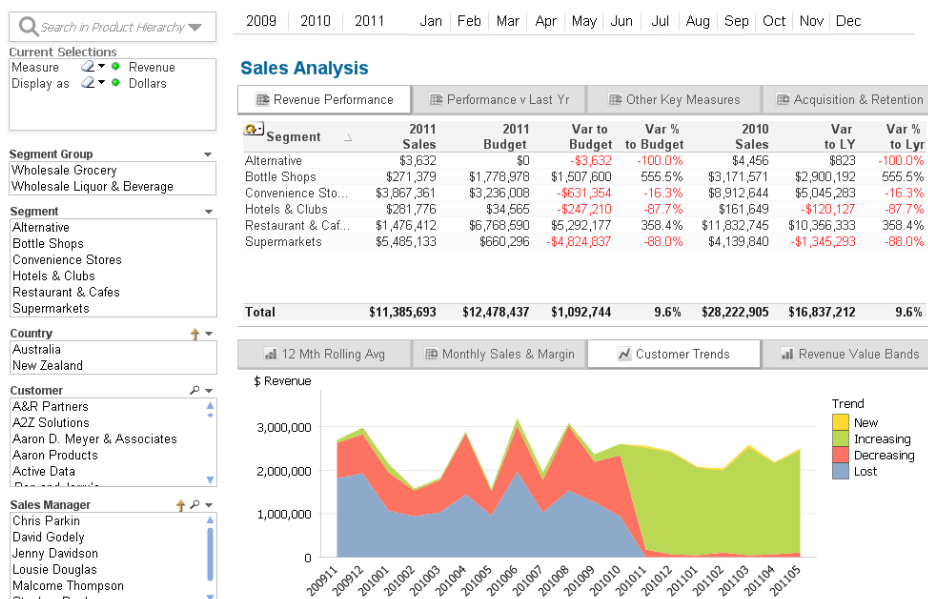


Рис. 2 Доработанное приложение

Но иногда и этого оказывается недостаточно. Например, когда требуется приложение на iPad для руководителей или приложение из российского филиала компании планируют использовать в зарубежных филиалах. И тут важен не только уникальный функционал QlikView, но и внешний вид. Причем требования могут формулироваться от "что-то скучновато" до "хотим, чтобы было Ах! или прямо Вау!", а в качестве примеров приводятся красивые приложения на iOS или веб-сайты, где работали профессиональные дизайнеры.

Но прежде, чем пытаться угнаться за профессионалами в дизайне, давайте для начала повторим базовые требования к дизайну, чтобы больше не совершать ошибок новичков. В этой статье у меня получилось Правило пяти П.

Правило пяти П

Рекомендаций для создания хорошего дизайна много, но я хотела бы обратить ваше внимание на 5 принципов, которые не используют или часто нарушают новички:

- *Постоянство*
- *Пространство*
- *Правильные диаграммы*
- *Простота и пояснения*
- *Привлечение внимания*

Постоянство

Элементы управления и общие объекты должны на всех листах располагаться одинаково с точностью до пикселя. Чтобы кнопки не скакали при переключении между листами. И один раз разобравшись, где расположены фильтры, кнопки и меню, пользователь не будет тратить время на их поиск на остальных листах.

Сюда же отнесем требование единообразия в цвете, шрифтах, стилях диаграмм, иконок и кнопок.

Если есть важные измерения с небольшим количеством значений, то лучше чтобы они во всем приложении имели свои постоянные цвета. Запомнив одну легенду, пользователь сразу сможет читать много диаграмм. Например, в главе о плоском дизайне на [Рис. 17](#) список *Статус требования* содержит легенду на экране слева и в диаграммах используются цвета этой легенды; для анализа отклоненных требований, достаточно будет найти круги и сегменты фиолетового цвета.

То же касается мер. Часто в приложениях вижу, как разработчик выбрав для динамики плана оранжевый, а для факта - зеленый цвет, на соседнем графике сделал их другого цвета, просто потому что там не динамика, а план-факт по менеджерам. И на следующем листе опять берется новый цвет. Непостоянство в выборе цветов не только замедляет работу пользователя, но и может привести к ошибочным выводам.

Пространство

Не захламляйте экран, пытайтесь потратить с пользой каждый пиксель. Как бы ни хотелось добавить еще вот ту супер диаграмму или еще пару фильтров, оставьте между объектами больше “воздуха”. Приложение выглядит очень неаккуратно, если объекты разбросаны как попало. Обычно дизайнеры используют модульную сетку для расположения объектов. Нам вполне хватит создания нескольких линий. Если располагать объекты по ним, то интерфейс будет выглядеть намного аккуратнее.

Правильные диаграммы

Выбирайте правильные диаграммы, исходя из ваших данных и ваших задач.

Нужно выбрать такое представление данных, которое позволит пользователю максимально просто и безошибочно понять изучаемые данные и возможно открыть для себя что-то новое, какие-либо паттерны или связи между данными.

Примеры можно посмотреть здесь <https://community.qlik.com/docs/DOC-7816>

Также рекомендую посмотреть полезное приложение *Data Visualization.qvw*, которое после установки QlikView можно найти в папке *Program Files\QlikView\Examples\Documents*.

Простота и пояснения

Стремитесь максимально упростить пользователю работу с вашим приложением, что все было интуитивно понятно.

Как донести до пользователя, что для чего ему ваше приложение и что делает каждый объект?

Варианты на выбор

- страница **Помощь** с описанием основных фишек QlikView и элементов управления;
- описание задач, решаемых вашим приложением, например, первый лист **Введение** и последний лист **О приложении**;
- добавьте в свой арсенал иконки, подписи к объектам, всплывающие подсказки, скрытые текстовые объекты с подробным описанием, появляющиеся по нажатию кнопки.

Пользователь должен понимать, где он в данный момент находится и как быстро перейти в любое другое место приложения, куда направит его ассоциативное мышление.

Для этого надо продумать удобную навигацию по приложению. Примеры приведу в разделе *Навигация*.

Привлечение внимания

Использование цвета

Что первым привлекло ваше внимание на этих рисунках?



Рис. 3



Рис. 4



Рис. 5

Не забывайте, что кроме использования ярких цветов, у вас есть возможность привлечь внимание к нужным значениям с помощью интенсивности цвета или контраста.

Слишком большое количество цветов усложняет восприятие информации. Отсюда вывод: ограничиваем используемую палитру и добавляем нейтральные цвета.

Размер имеет значение

Отведите больше места для самых важных показателей.

Иконки, размер шрифта, фон и границы

Все они могут помочь вам в привлечении внимания пользователя, но не злоупотребляйте ими!

“Плоский” дизайн (Flat design)

Обзор

Давайте перейдем к рассмотрению одного из направлений современного дизайна, подходящего для нашей задачи.

В QlikView хорошо смотрится и достаточно просто реализуется [flat design](#). Это легкий, чистый интерфейс с простыми шрифтами и приличным расстоянием между элементами без отвлекающего объема, с минимумом теней и прочих изысков, чтобы ничто не отвлекало пользователя от его главной задачи. По сути мы стремимся свести интерфейс к минимуму, оставив только контент, а именно данные для анализа.

Вот примеры веб-дизайна, которые нам пригодятся для вдохновения:

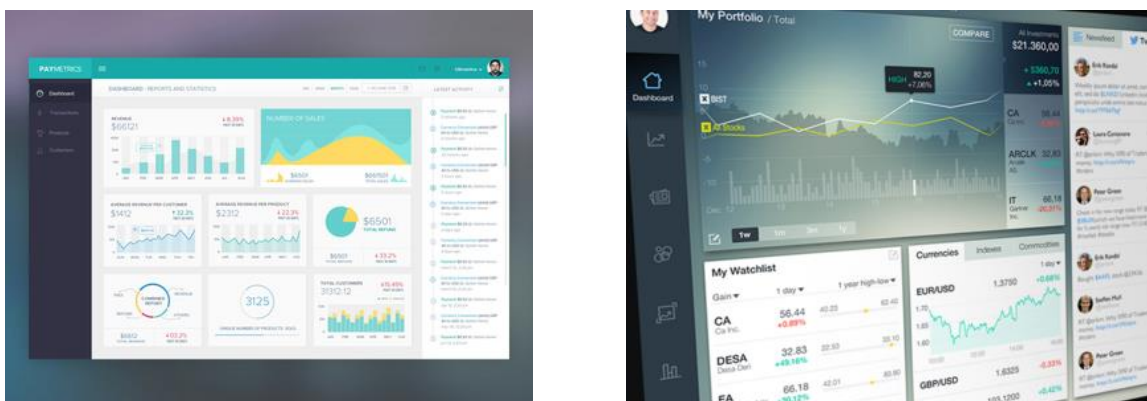


Рис. 6

Обратите внимание на блоки информации и расстояние между ними. Разработку дизайна начинают именно с создания каркаса приложения, состоящего из блоков. Чтобы не тратить время на изучение инструментов прототипирования для создания макета приложения, можно воспользоваться листом бумаги и карандашом. Набросайте расположение основных блоков дашборда и остальных листов и продумайте удобную навигацию по приложению.

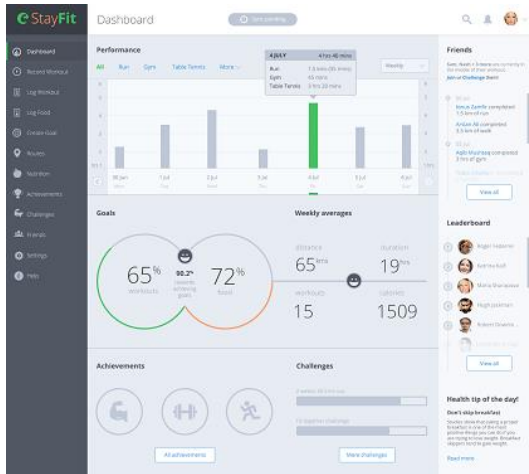


Рис. 7

Рекомендации по реализации “плоского” дизайна в QlikView

Ссылки на примеры QlikView приложений:



Рис. 8 QlikView for Salesforce



Рис. 9 Insurance Demo



Рис. 10 My life in data



Рис. 11 OEE Analysis

Приведение стиля объектов к плоскому дизайну

- Кнопки и значения KPI создаем на основе текстовых объектов. Не используйте LED стиль в датчике. Избегайте объемных кнопок и сложных картинок.
- Диаграммы делаем без границ, с прозрачным фоном, с плоским стилем (полоски, а не столбики или цилиндры). Также можно скрывать заголовки.
- Иконки максимально простые, схематичные.
- Обязательно оставляем свободное место между блоками информации, как бы ни хотелось заполнить лист по максимуму.

- Часть настроек можно сделать один раз в начале работы над приложением в Свойствах документа (меню Параметры-> Свойствах документа) на вкладках Шрифт, Макет и Заголовок. Тогда новые объекты будут создаваться в выбранном вами стиле, экономя вам время.
- Не рекомендую использовать стандартные датчики (спидометры). Они занимают много места на экране и сильно отвлекают внимание пользователя.
 - Можно оставить только текст со значением показателя и раскрасить цифры в нужный цвет.
 - Либо можно воспользоваться [примером создания плоского датчика](#).

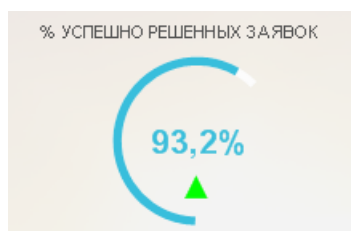


Рис. 12

- Достаточно часто требуется поместить в один датчик много показателей для сравнения, например: план, факт, прогноз и цветовые сегменты “хорошо-плохо”. В этом случае обратите внимание на диаграмму Буллет ([bullet chart](#)).

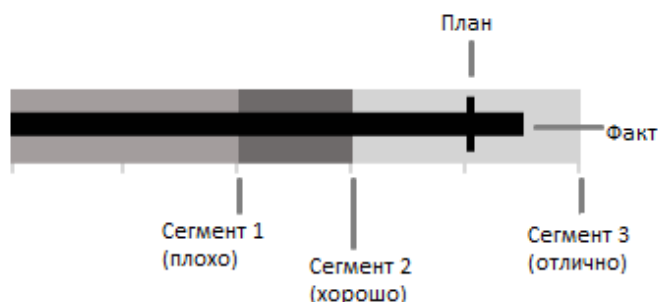


Рис. 13 Описание диаграммы bullet

Ее можно сделать с помощью расширения Bullet (extension Bullet). После установки QlikView архив расширений можно найти в папке *Program Files\QlikView\Examples\Extensions*. Установите расширения, запустив файл **Extensions.qar**. Пример работы с расширениями можно найти в **Extension Examples.qvw** (*Program Files\QlikView\Examples\Documents*). Не забудьте включить WebView в меню Вид.

- Используя богатые возможности QlikView, вы можете создать эту диаграмму и без расширения. На [Qlik Community](#) можно найти несколько способов реализации такой задачи.



Рис. 14 На основе комбо-диаграммы

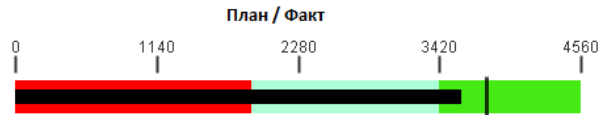


Рис. 15 [На столбчатую диаграмму наложен датчик](#)

Палитра

Большинство цветовых палитр плоского дизайна основаны на 2-3 основных цветах, причем выбираются чистые яркие цвета, без лишних переходов и градиентов.

1. Очень удобно создать переменные для этих цветов

```
vColor_Blue_1 = RGB(0,128,255)
```

и во всех объектах указывать цвет с помощью переменных

```
$(vColor_Blue_1)
```

Если результат не понравится, можно будет легко изменить палитру готового приложения.

2. Либо создаем таблицу для изменения основного цвета на лету:

В начале скрипта добавляем код:

```
SET HidePrefix = '_'; //чтобы в текущих выборках не было видно полей,
начинающихся на "_", в частности поля _Цвет
```

```
SET vColorBright = '=if(count(Код_цвета)=1,Код_цвета,RGB(56,190,220))';
```

Цвета:

```
LOAD * INLINE [
    _Цвет, Код_цвета,
    Синий, "RGB(0,102,179)"
    Светло-голубой, "RGB(102,226,238)"
    Голубой, "RGB(56,190,220)"
    Зеленый, "RGB(1,205,57)"
    Желто-оранжевый, "RGB(243,162,10)"
];
```

Выводим на лист список (listbox) для поля `_Цвет` со свойством "Всегда одно значение".

А в свойствах диаграмм и текстовых объектов указываем вычисляемый цвет `$(vColorBright)`.

Пример такой смены основного цвета можно видеть на следующих рисунках (Рис. 16 и Рис.

17).

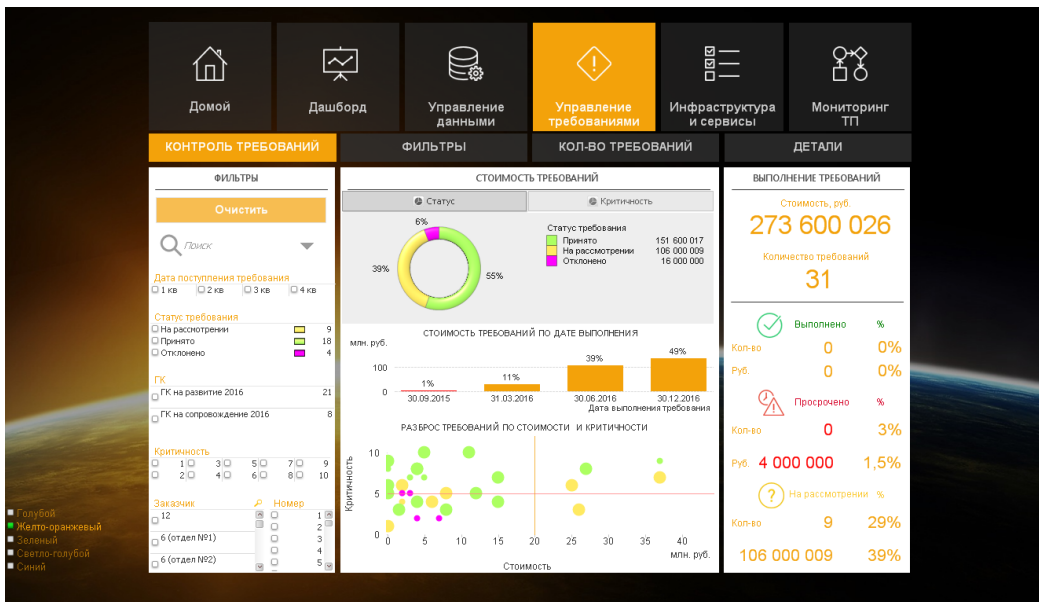


Рис. 16

Листбок для выбора цвета я вывела слева для наглядности.

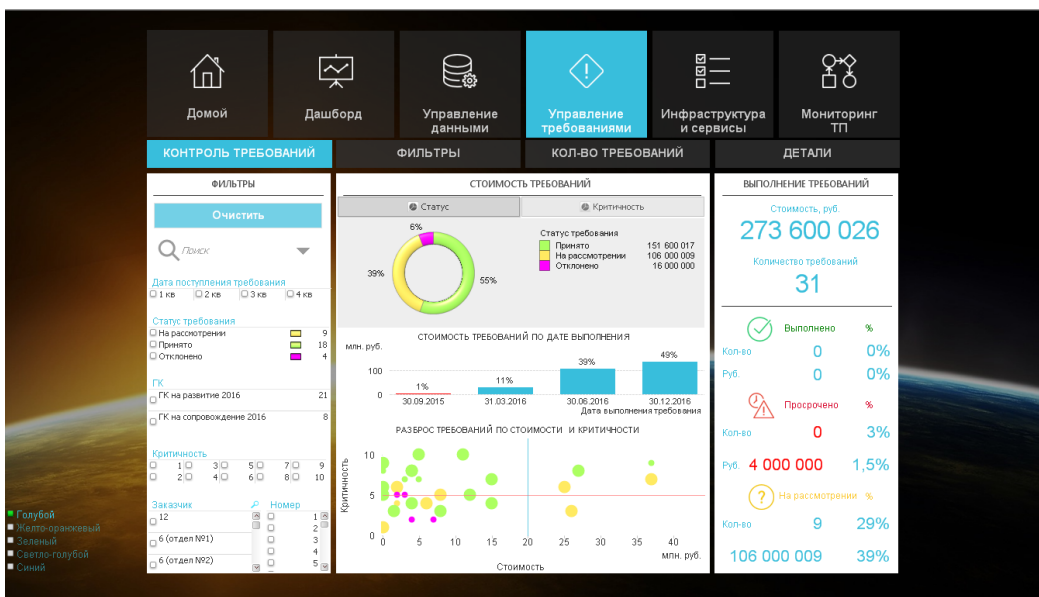


Рис. 17

Обычно я ищу хорошо сочетающиеся цвета здесь:

<http://www.colourlovers.com/> - в поиске можно задать названия цветов или просто поискать симпатичные вам цвета среди самых популярных палитр;

<http://www.colorpicker.com/> - быстро и просто выбираете основной цвет на палитре или вводите координаты и получаете несколько вариантов палитр на выбор.

<http://colorscheme.ru/> - мощный инструмент по выбору палитры, есть отличная возможность увидеть варианты дизайна веб-страницы с выбранной палитрой.

Цвет может быть задан по-разному, например: RGB(), HSL(), HEX код, CMYK.

Для перевода координат в удобный формат можно воспользоваться онлайн-помощниками:

<http://www.spycolor.com/>

<http://www.rgbtohex.net/hextorgb/>

Навигация

На рисунках с примерами веб-дизайна (Рис. 6-9) мы видели две панели, позволяющие пользователю мгновенно оценить, где именно он находится и как перейти к интересующей информации:

- информационная панель сверху с названием вкладки и элементами управления
- панель навигации слева (может также содержать несколько важных показателей, которые видно всегда). Она может быть выпадающей или стационарной.

Такой подход будет хорош и для QlikView приложений. Но конечно, вы сами должны выбрать вариант, наиболее подходящий к вашей задаче.

Если листов немного, можно обойтись горизонтальным меню, в QlikView оно может быть таким:



Рис. 18

Для того, чтобы убрать вкладки листов, в свойствах документа на вкладке **Общие** поставьте галочку *Скрыть ряд вкладок*. Теперь вся навигация будет происходить через созданные вами текстовые объекты, в свойствах которых будет указано действие *Активировать лист* и ID листа, на который нужно перейти после нажатия (Рис. 19).

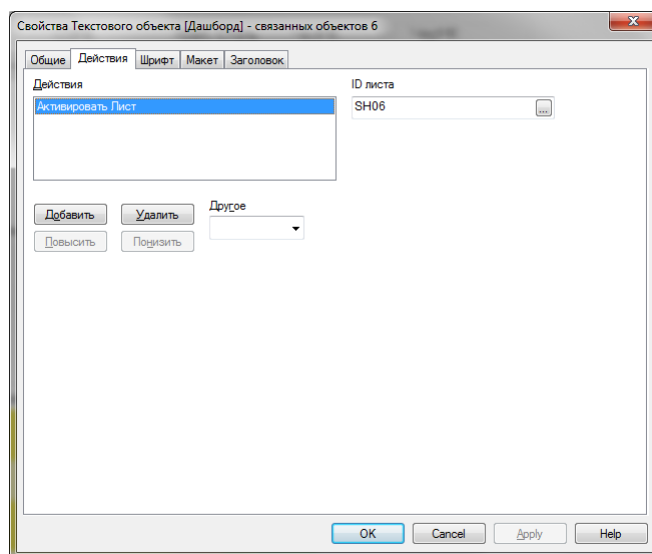


Рис. 19

ID листа легко найти в свойствах листа на вкладке **Общие** в правом верхнем углу.

Примеры выпадающего и статичного меню в QlikView можно увидеть на рисунках ниже (Рис. 20).

Выпадающее меню делается на основе текстовых объектов с условием на показ:

- создаем переменную **vPictShow** (Параметры-> Обзорщик переменных)
- в текстовом объекте или кнопке, отвечающей за открытие и закрытие нашего меню, добавляем действие *Установить переменную* и для переменной **vPictShow** указываем значение **=if(vPictShow=1,0,1)**
- для всех объектов меню добавляем условие на показ: Свойства текстового объекта->Макет->Показать->По условию-> **vPictShow=1**.

Подобные меню создаются на отдельном листе, а затем всё сразу копируется на все листы через контекстное меню (вызываемое правой кнопкой мыши) *Вставить объект листа как ссылку*. Тогда неизбежные правки, сделанные в одном месте, будут применены для всех связанных объектов. Также через контекстное меню можно будет откорректировать позицию всех связанных объектов сразу. Останется только продумать, что делать с активными строками, которые надо выделять другим цветом. Можно либо создать вычисляемый цвет, либо разорвать связь с остальными объектами и сменить цвет вручную.

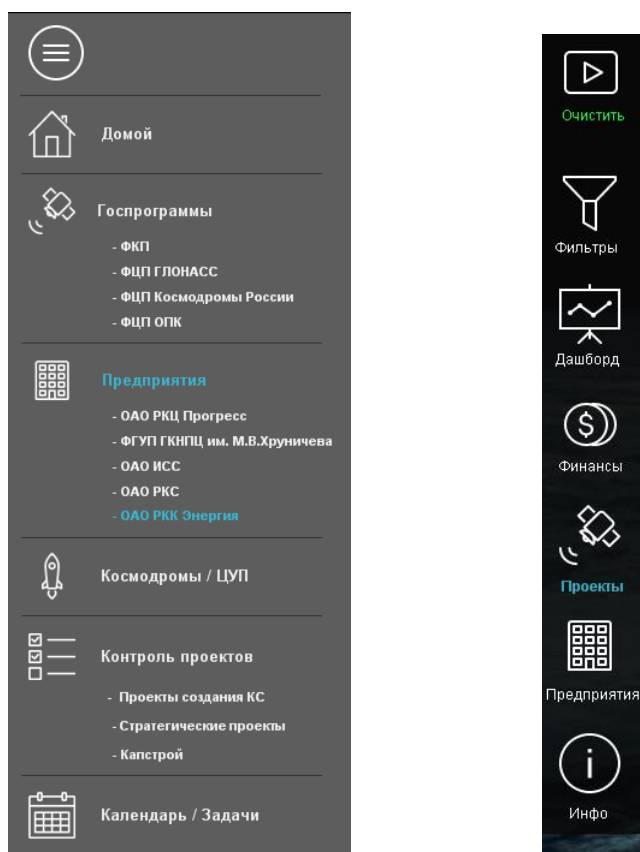


Рис. 20 Выпадающее и статичное меню

Те, кто не хочет возиться с большим количеством скрытых объектов, могут использовать в качестве меню сводную таблицу с прозрачным фоном, триггеры события поля и файл со структурой в Excel.

Еще несколько полезных ссылок

Примеры дашбордов для вдохновения:

https://ru.pinterest.com/search/pins/?rs=ac&len=2&q=ui+dashboard+design&term_meta%5B%5D=ui%7Cautocomplete%7C1&term_meta%5B%5D=dashboard%7Cautocomplete%7C1&term_meta%5B%5D=design%7Cautocomplete%7C1

<https://econsultancy.com/blog/62844-24-beautifully-designed-web-dashboards-that-data-geeks-will-love>

Бесплатные наборы элементов дизайна и иконок в стиле flat design:

<http://www.chethstudios.net/2013/07/20-free-flat-ui-icon-packs-design-elements.html>

<http://www.flaticon.com/>

Итоги

Для разработчиков приложений QlikView, которым было не совсем понятно, что делать, когда просят улучшить дизайн приложения, я предложила направление Flat design и дала рекомендации, как выполнить его условия в QlikView.

Чтобы не изобретать колесо, а именно с нуля не придумывать каркас интерфейса, дизайн и навигацию по нему, рекомендую использовать понравившиеся дашборды веб-сайтов для переноса в QlikView. Совет для начинающих или для большей экономии времени: выберите понравившееся QlikView приложение и на его основе сделайте свое приложение с учетом рекомендаций в статье.

Бонус для тех, кто дочитал до конца :-)

Пасхалка (Easter egg) от разработчиков Qlik Sense /theme/highvis

Возможность создавать темы и применять их к приложениям Qlik Sense Enterprise в принципе есть, но пока глубоко зарыта и недоступна простым пользователям.

Тем не менее, тем, кто хочет попробовать легко сменить вид приложения, предлагаю открыть Qlik Sense Desktop, а затем скопировать в браузер ссылку

<http://localhost:4848/sense/app/Executive%20Dashboard/sheet/FaQeFa/state/analysis/theme/highvis>

или подставьте имя своего приложения в строку

<http://localhost:4848/sense/app/Executive%20Dashboard/theme/highvis>

Результат можно увидеть на рисунке ниже:



Рис. 21 Тема highvis