

Лабораторная работа №3. Создание и тестирование статических прототипов пользовательских интерфейсов.

Составил М.А. Бакаев, к.т.н., доц. кафедры АСУ НГТУ

Цель работы: освоить создание **статических** прототипов-«скелетов» пользовательских интерфейсов, опробовать процесс тестирования прототипов и их итерационного улучшения.

1. Теоретическая справка

1.1. Создание прототипов

Прототипирование (создание прототипов) и итеративное улучшение проектов интерфейса признаются одними из наиболее мощных методов в проектировании взаимодействия человека и компьютера. Использование прототипов несёт, в частности, следующие преимущества¹:

- Стимулирование коммуникации между дизайнером и пользователями (заказчиком).
- Возможность уточнить проектные требования и цели проектирования, ещё не затратив много сил и времени на создание реального продукта.
- Возможность проверить идеи, заложенные в решении, на ранних стадиях разработки, избежать дорогостоящих ошибок.

В процессе прототипирования [1] можно выделить несколько стадий:

- **Концептуальный дизайн:** для исследования различных метафор в интерфейсе и подходов к дизайну.
- **Дизайн взаимодействия:** для организации структуры экранов или страниц и переходов между ними.
- **Дизайн экранов:** разработка внешнего вида каждого из конкретных экранов или страниц.
- **Тестирование:** оценка прототипов для их последующего итерационного улучшения (при помощи небольшого тестирования с представителями реальных пользователей или с использованием «эвристического анализа» экспертами).

С точки зрения глубины проработки выделяют следующие виды прототипов пользовательских интерфейсов [2]:

- **«Скелет»**, каркас (wireframe) – прототип «низкой точности» (lo-fi), который отражает основной контент и его структуру, а также расположение элементов интерфейса [3]. Как правило, является чёрно-белым и не содержит элементов визуального оформления. Может создаваться на бумаге (paper prototypes) или в специализированных редакторах (пример реализации в редакторе *Balsamiq* представлен на Рис. 1). Обычно является статическим (не подразумевает взаимодействия пользователя с элементами прототипа), хотя бумажное прототипирование в сочетании с методом Wizard of Oz позволяет осуществлять искусственную динамизацию прототипов. Полезен на начальных этапах проекта, в первую очередь для устранения недопониманий между членами команды разработчиков и пользователями (являются, в некотором смысле, способом документирования в проекте).

¹ <https://www.webmascon.com/topics/testing/13a.asp>

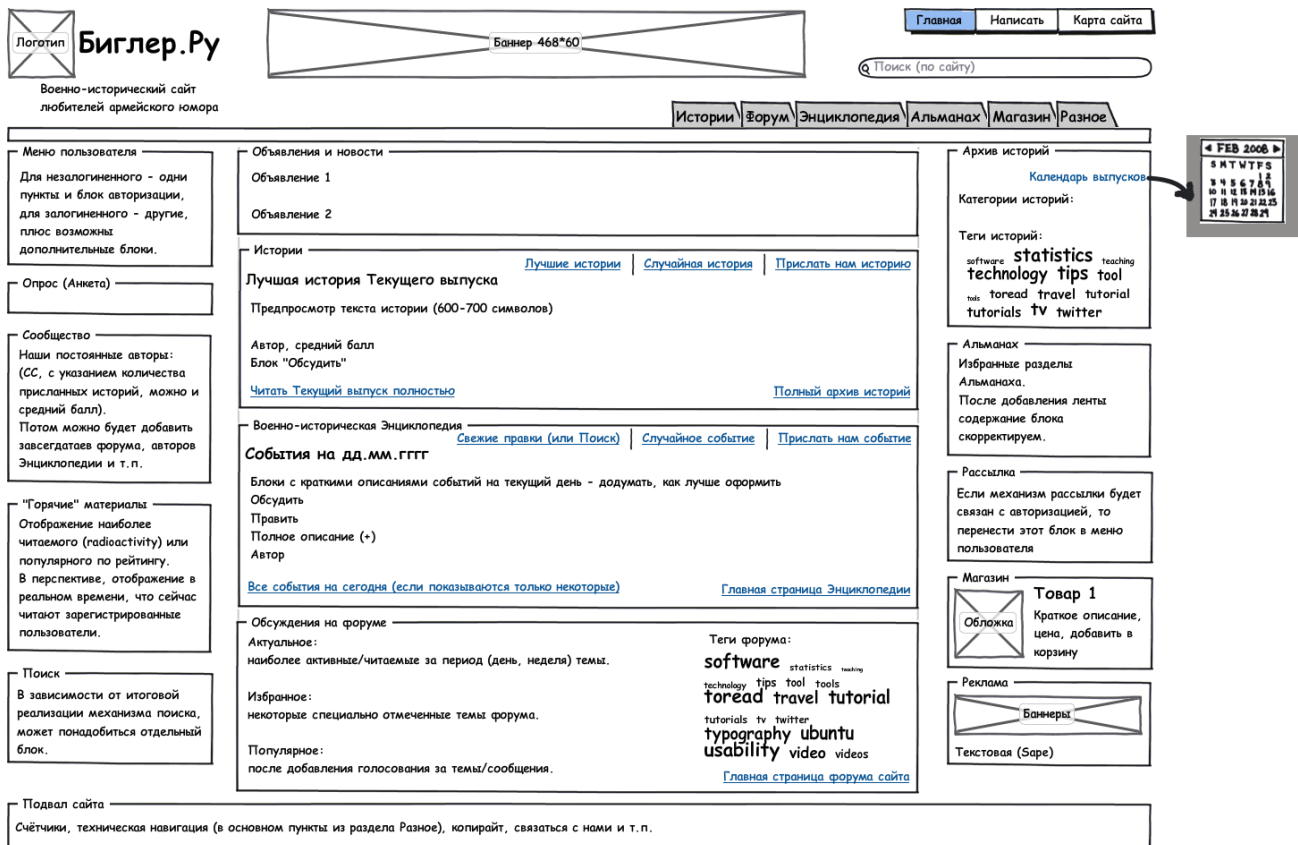


Рис. 1. Пример статичного прототипа-«скелета», созданного в программе Balsamiq.

- **Макет (mockup)** - статический прототип, основной акцент в котором сделан на визуальное оформление (графический дизайн интерфейса). Обычно создаётся в графических редакторах (при разработке новых, уникальных визуальных решений) или в HTML/CSS-редакторах (при адаптации существующих). Полезен при согласовании с заказчиком - как показывает практика, основное внимание он обычно уделяет внешнему виду, а не процессу взаимодействия.
- **Динамические (интерактивные) прототипы** - поддерживают взаимодействие с пользователями (переходы по ссылкам, заполнение полей и т.п.), иногда часть функционала или бизнес-логики конечного продукта. Могут создаваться в специализированных редакторах (таких, как *Axure* или *Adobe XD*) или на основе систем управления содержимым сайтов (CMS).

1.2. Тестирование прототипов

Тестирование прототипов пользовательских интерфейсов [4] может осуществляться на разных стадиях проработки продукта с использованием соответствующих методов (см. список в [5]), например:

1. **Тестирование скелетов или бумажных прототипов.** Типичными целями являются: проверка общей полезности идеи продукта, выявление пропущенных и дополнительных требований к продукту, проверка информационной архитектуры, понятности текстового контента. Для статических прототипов может использоваться метод опроса пользователей или экспертов (см. пример в Приложении А).
2. **Тестирование динамических прототипов.** Типичными целями являются: проверка функционала и уровня юзабилити продукта, выявление проблемных моментов в процессе взаимодействия пользователя с продуктом при выполнении своих задач, рабочего процесса взаимодействия с продуктом (в т.ч. навигации), сравнение нескольких вариантов реализации интерфейса. Распространённые методы: юзабилити-тестирование, эвристическое обследование.
3. **Тестирование конечного продукта.** Возможные цели и методы в основном совпадают с теми, что используются при тестировании динамических прототипов. Некоторые отличия тестирования конечного

продукта: используются реальные данные (например, перечень товаров в эл. магазине), могут оцениваться субъективные аспекты взаимодействия (например, эстетическое впечатление пользователя).

Наиболее распространённые проблемы и ошибки, связанные с прототипированием, включают в себя [6]:

- **Не иметь цели:** прототип должен создаваться для решения конкретной проблемы (например, апробация ключевого функционала будущего программного продукта, проверка понятности информационной архитектуры веб-сайта и т.д.).
- **Избегать проблем:** прототип, который вызвал затруднения у пользователя и приращение понимания у разработчиков - лучше, чем тот, тестирование которого прошло без замечаний.
- **Любить прототип:** не следует вкладывать слишком много усилий в доведение более-менее удачного прототипа «до совершенства», лучше попробовать несколько различных идей и подходов.
- **Неверно выбирать детализацию:** глубина проработки прототипа должна быть минимально достаточной для выполнения поставленной цели. Например, слишком качественное графическое оформление может мешать (отвлекать заказчика), если целью является тестирование базового функционала.

1.3. Программные инструменты

В настоящее время существует значительное количество программных инструментов, предназначенных для прототипирования пользовательских интерфейсов². Пожалуй, наиболее обширный функционал представлен в **Axure**³, который имеет широкое распространение среди профессиональных разработчиков. Для **Axure** также разработано большое количество обучающих материалов, в том числе на русском языке⁴, **есть возможность использования бесплатной ознакомительной версии**, работающей в облаке или устанавливаемой на компьютер. Вместе с тем, освоение **Axure** является относительно трудоёмким процессом, а предоставляемый этим продуктом функционал несколько избыточен, если стоит задача разработки только лишь статичных прототипов-скелетов⁵.

В качестве альтернатив или дополняющих **Axure** инструментов можно отметить:

- **Balsamiq**⁶: редактор, специализирующийся именно на создании прототипов-скелетов. Доступна бесплатная ознакомительная версия (в течение 30 дней), работающая в облаке через браузер или же устанавливаемая на компьютер.
- **Adobe XD**⁷: позволяет как создание скелетов, так и интерактивных прототипов. Стартовый тарифный план является бесплатным для одного проекта.

Широко распространённым инструментом (правда, более любительского уровня) является также Sketch, делающий особый акцент на прототипировании интерфейсов мобильных приложений, однако данный продукт работает только на платформе macOS. Существует также значительное количество бесплатных инструментов, доступность которых не ограничена ознакомительным периодом.

² См., например, обзоры <https://dzone.com/articles/27-open-source-web-ui-mockup-tools> и <https://www.creativeblog.com/how-to/20-best-ui-design-tools>

³ <https://www.axure.com/>

⁴ См. вводный курс в современную версию Axure <https://www.uplab.ru/blog/kak-my-delaem-prototypy-saytov/> или набор видео для более старой версии <https://habr.com/ru/post/277395/>

⁵ Примечательно, что в Axure есть возможность переключать степень проработки прототипа: <https://www.axure.com/blog/hi-fi-to-lo-fi-with-the-flip-of-a-switch>

⁶ <https://balsamiq.com>

⁷ <https://www.adobe.com/products/xd.html>

2. Задание на лабораторную работу

1. Изучить теоретическую справку и **все пункты** списка литературы.
2. Определить основное функциональное наполнение разрабатываемого вами программного продукта, исходя из наработок по анализу задач из предыдущей лабораторной работы и темы магистерской диссертации.
3. Выбрать программный инструмент для реализации статического прототипа в данной лабораторной работе.
 - рекомендуемый преподавателем вариант – *Balsamiq* (как более простой в освоении), однако вы можете выбрать иной (удобный или знакомый для вас) инструмент;
 - необходимо иметь в виду, что в следующей лабораторной работе вам будет необходимо осуществлять дальнейшую проработку своего программного продукта, создавая динамический прототип, причём рекомендуемый преподавателем программный инструмент – *Axure*;
 - обратите внимание, что в *Axure* реализована возможность импорта проектов из *Adobe XD*⁸ и *Sketch*⁹, но не из *Balsamiq* (оттуда может быть импортирован только графический файл).
4. Разработать статические скелеты-прототипы пользовательского интерфейса вашего программного продукта (на веб-, мобильной или десктоп-платформе). Разрабатываемые экраны должны **соответствовать основному функционалу вашего продукта** (экранов должно быть создано не менее 2-х, для мобильной платформы – не менее 4-х). **Согласовать результаты вашей работы с преподавателем.**
5. Протестировать разработанные вами прототипы как минимум с одним пользователем или экспертом.
 - для тестирования вы можете использовать список вопросов из Приложения А (полностью или частично, т.е. удалив нерелевантные для вашего продукта вопросы);
 - обратите внимание, что в случае использования опросника из Приложения А пользователь или эксперт **не должны заранее подробно знать** о назначении и функционале вашего программного продукта;
 - вы также можете предложить свой вариант опросника (не забыв указать ссылку на источники) или вообще иной метод тестирования.
6. Предложить и реализовать улучшения по результатам тестирования прототипов (т.е. создав новые версии прототипов на следующей дизайн-итерации).
7. Составить отчёт по лабораторной работе.

Содержание отчёта по лабораторной работе

1. Краткое описание разрабатываемого вами программного продукта и функционала, который будет отражён в создаваемых в рамках лабораторной работы прототипах.
2. Статические прототипы-скелеты пользовательского интерфейса вашего продукта, разработанные на первой итерации: в графическом формате и файлы-исходники (с указанием выбранного программного инструмента).
3. Результаты тестирования: информация о пользователе или эксперте, итог проведённого опроса или применения другого метода (в случае использования собственного опросника или метода – описать их).
4. Ваши выводы по тестированию, их соотнесение с конкретными элементами прототипов, разработанных на первой итерации, описание направлений улучшений.
5. Улучшенные прототипы, разработанные на второй итерации: в графическом формате и файлы-исходники.

⁸ <https://www.axure.com/blog/import-adobe-xd-designs-into-axure>

⁹ <https://www.axure.com/blog/import-sketch-designs-into-axure-rp>

Литература

- 1) С. Тихонов. Этапы разработки интерфейса. 2017. Доступ по адресу: <https://designpub.ru/этапы-разработки-интерфейса-baf666d6ad8f>
- 2) М. Тредер. Вайрфреймы, прототипы и мокапы – в чем разница? Пер. с англ. О. Шевцова, К. Олейниченко. 2014. Доступ по адресу <https://awdee.ru/wireframes-prototypes-and-mockups/>
- 3) IDBI. Что такое вайрфреймы и как их использовать в дизайне интерфейсов. 2020. Доступ по адресу <https://idbi.ru/blogs/blog/wireframe-v-veb-dizayne>
- 4) Д. Мелентьев. Тестирование прототипов при разработке программного продукта. 2017. Доступ по адресу <https://netology.ru/blog/testing-prototypes>
- 5) All about UX. Methods for early prototypes. Доступ по адресу <https://www.allaboutux.org/early-prototype-methods>
- 6) Н. Бабич. 6 ловушек прототипирования. Пер. с англ. 2018. Доступ по адресу https://contented.ru/media/prototyping_pitfalls

Приложение А

Пример вопросов для тестирования статичного прототипа

Общая идея продукта:

1. На ваш взгляд: для чего предназначен продукт, какую проблему и для какого пользователя он решает?
2. Если вам доводилось пользоваться продуктами-конкурентами, то в чём вы видите сходство и в чём отличие?

Функциональные требования:

3. Какие действия нужно будет предпринять пользователю, чтобы выполнить свою задачу?
4. Какого функционала, на ваш взгляд, не хватает? Возможно, что-то наоборот лишнее?

Информационная архитектура:

5. Понятны ли названия навигации и ссылок?
6. Есть ли среди ссылок такие, которые заставили вас сомневаться, куда они ведут?

Контент (содержание):

7. Присутствует ли какая-либо информация, которая вам неясна?
8. Достаточное ли количество текстовой информации представлено? Возможно, что-то наоборот излишнее?
9. Видите ли вы какие-либо ошибки в контенте?

Общая оценка:

10. Легко ли было понять функционал продукта? Насколько эффективным вам кажется продукт в выполнении своей задачи?
11. Ваше впечатление от расположения элементов интерфейса? Насколько оно привычно или необычно?
12. Ваше общее впечатление от интерфейса продукта? Что понравилось и не понравилось?