

Последовательность разработки веб-ориентированных приложений

Титова Ольга Ивановна
Минск, 2017

Содержание

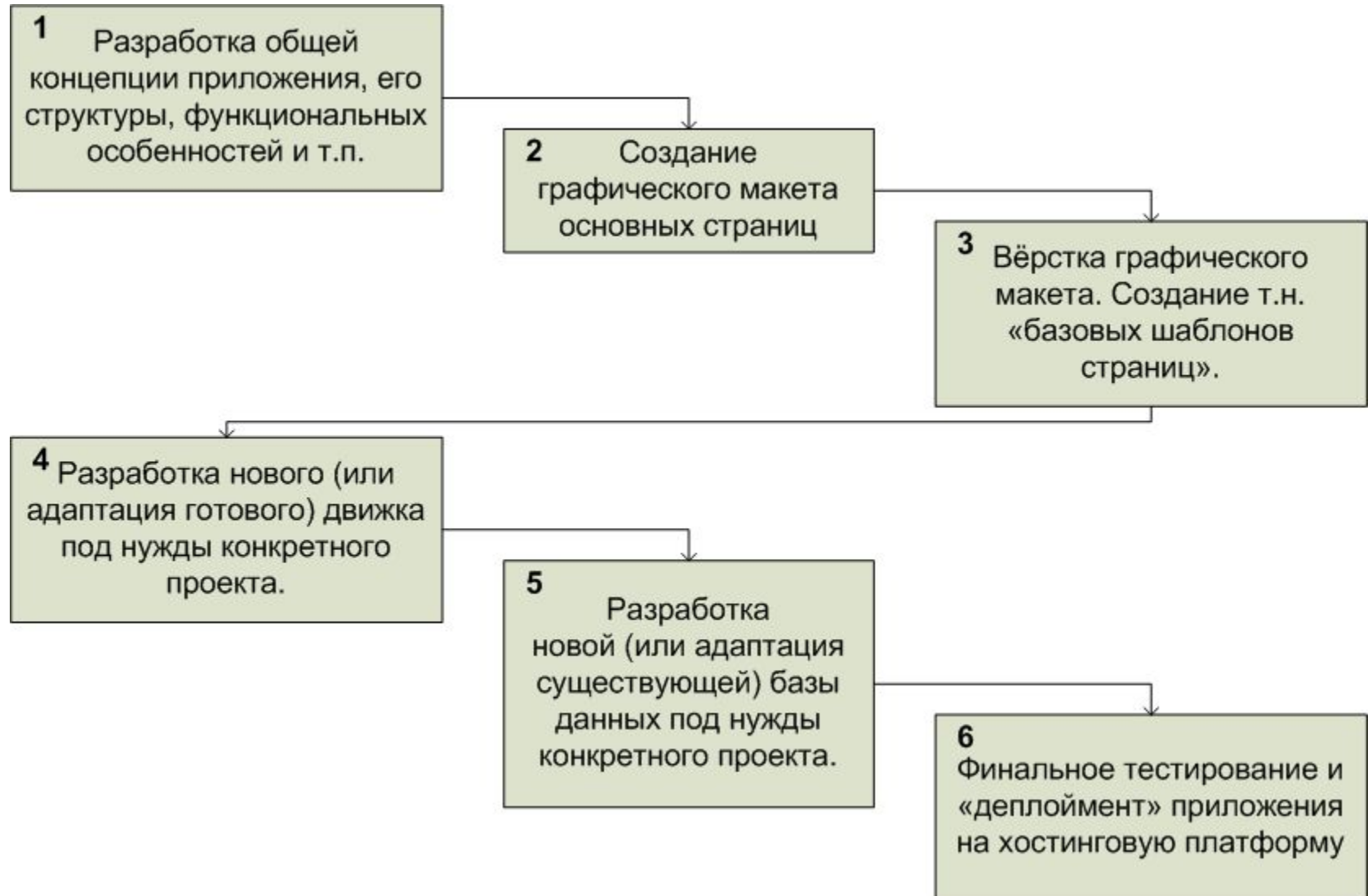
1. Разработка общей концепции приложения.
2. Создание графического макета.
3. Вёрстка.
4. Разработка движка приложения.
5. Разработка базы данных.
6. Тестирование и ввод в эксплуатацию.

Вступление

Прежде, чем мы приступим к рассмотрению конкретных технологий и решению конкретных задач, следует изучить **классический алгоритм разработки веб-ориентированных приложений**.

Давайте рассмотрим рисунок...

Стадии разработки приложения



Концепция

1. Разработка общей концепции приложения, его структуры, функциональных особенностей и т.п.

На первой стадии всей проектной команде следует максимально сплочённо поработать над тем, чтобы выработать единую стройную концепцию веб-приложения.

Графический макет

2. Создание графического макета основных страниц

В крупных проектах «макетироваться» могут все страницы, причём довольно подробно.

В небольших проектах **обычно ограничиваются макетами главной страницы, «второстепенной страницы»** (единой по внешнему виду для всех подразделов сайта), а также **страниц, принципиально отличающихся от главной и «второстепенной»** (карты сайта, страницы поиска, страниц каталога, форм регистрации и заказа товара, корзины и т.п.)

На данной стадии работы над проектом **основную задачу выполняют художник и дизайнер** (если они есть в виде «отдельных людей») или же те, кто выполняет их функции.

3. Вёрстка графического макета. Создание т.н. «базовых шаблонов страниц»

На данной стадии работы над проектом **основную задачу выполняет верстальщик**. На основе «картинок из фотошопа» (полученных на стадии 2) он **создаёт статические шаблоны страниц** (перечень страниц – см. в предыдущем пункте).

На выходе его работы **получаются страницы сайта в таком виде, в каком они могли бы быть сгенерированы движком при некоторых условиях**.

Желательно заполнять такие шаблоны реальной информацией, чтобы видеть, где что-то может «поехать», что-то не отобразиться и т.д.

ОЧЕНЬ ЖЕЛАТЕЛЬНО, чтобы верстальщик (равно как художник и дизайнер) имел представление о том, «что можно запрограммировать, а что – нет», т.к. в противном случае при дальнейшей работе с проектом веб-программистам придётся долго ломать голову, как в коде реализовать то, что «на картинке» нарисовалось так просто.

4. Разработка нового (или адаптация готового) движка под нужды конкретного проекта

На данной стадии работы над проектом **основную задачу выполняет команда веб-программистов**.

Они режут шаблоны, полученные на стадии 3, на «куски» и **организуют логику работы движка** таким образом, чтобы из этих «кусочных шаблонов» в итоге получился красивый и функциональный сайт.

База данных

5. Разработка новой (или адаптация существующей) базы данных под нужды конкретного проекта

На данной стадии работы над проектом **основную задачу выполняет специалист по базам данных** (если он есть в проектной команде) или же – всё те же веб-программисты.

Тестирование

6. Финальное тестирование и «деплоймент» приложения на хостинговую платформу

Тестирование должно быть на КАЖДОЙ стадии работы с проектом. Чем раньше будет обнаружена ошибка, тем быстрее, проще и дешевле будет её исправление.

Однако в конце проекта тестированию следует уделить особое внимание. Следует проверить работу приложения как во всех «стандартных», так и в хотя бы наиболее вероятных «нестандартных» ситуациях.

Когда всё готово, сайт закатывается на хостинговую платформу, где снова тестируется.

Среды дизайна и вёрстки

Основными программными средствами, помогающими являются:

- 1) Adobe Photoshop
- 2) Редакторы HTML/XML/CSS

Adobe Photoshop в рекомендациях не нуждается. Графику удобнее всего рисовать именно в нём. Также в нём удобно создавать исходные макеты страниц, показывать их заказчику, вносить правки и так по кругу до того момента, пока макет не будет готов к вёрстке.

Среды дизайна и вёрстки

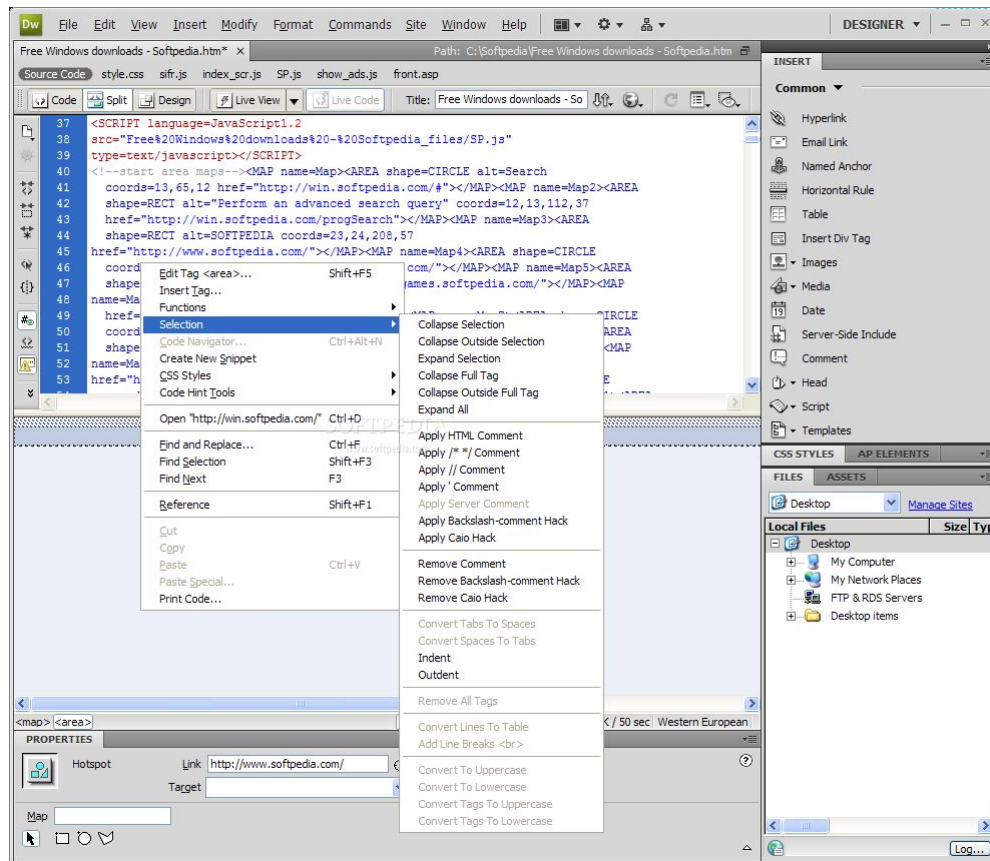
Вёрстку (т.е. преобразование «картинки» веб-страницы в реальный HTML/XML/CSS-код) **удобно выполнять редакторами, в которых реализованы следующие функции:**

- **подсветка** синтаксиса;
- **валидация** кода;
- встроенный **просмотрщик результата**;
- **возможность просмотра результата в браузерах одним нажатием клавиши** (желательно, чтобы можно было настроить несколько браузеров на разные клавиши или их комбинации);
- возможность «**визуального редактирования**».

Этим критериям удовлетворяют многие редакторы, но есть более универсальное средство...

Среды дизайна и вёрстки

Вышеперечисленным требованиям в наибольшей мере отвечает **Adobe DreamWeaver** (ранее — Macromedia DreamWeaver).



Спасибо за внимание!

Продолжение следует 😊