

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ЭССЕ

Системная инженерия на примере системы «Блог»

Студент: А.А. Валиков

Воронеж 2018

Содержание

1	Введение	3
1.1	Функциональные требования стейкхолдеров	3
1.2	Конструкционные ограничения архитектуры	3
1.3	Описываемая система как холон в иерархии	4
1.4	Системы: целевая, обеспечивающая, в эксплуатационной среде	4
2	Жизненный цикл системы согласно V-диаграмме	5
2.1	Определение требований	5
2.2	Архитектурное проектирование	6
2.3	Рабочее проектирование	6
2.4	Изготовление	6
2.5	Интеграция	6
2.6	Приём в эксплуатацию и эксплуатация	6
3	Практики системной инженерии	6
3.1	Сбор требований	6
3.2	Анализ требований	6
3.3	Архитектурный дизайн	7
3.4	Изготовление	7
3.5	Интеграция	7
3.6	Проверка всей системы	7
3.7	Переход к эксплуатации	7
3.8	Валидация	7
3.9	Эксплуатация	7
3.10	Обслуживание	7
3.11	Вывод из эксплуатации	8

1 Введение

1.1 Функциональные требования стейкхолдеров

Блог представляет из себя сайт, посвящённый одному человеку или компании, называемыми заказчиком. Для заказчика нужна возможность писать посты, добавлять разделы. Сайт состоит из «админки», – страницы, на которых заказчик сможет редактировать содержимое сайта и публичной части.

Заказчик

- Удобство администрирования, то есть структура страниц админки должна быть максимально проста и понятна
- Какие части сайта могут быть изменены. После того как сайт сделан, заказчик остаётся один на один с ним, и в случае возникновения новых требований к функционалу, админка может позволить (или не дать) добавить новые возможности.

Читатели блога

- Современный дизайн
- Интуитивно понятные элементы управления (куда тут что нажимать)
- Скорость загрузки и работы
- Доступность: сайт должен быть доступен максимальное количество времени для максимального количества читателей.

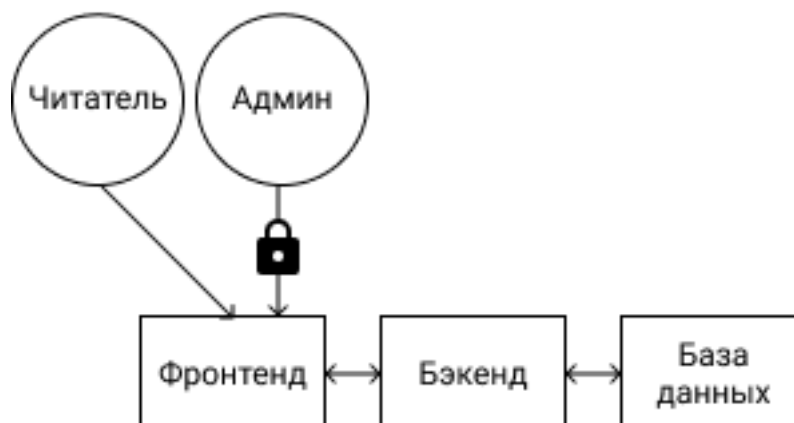
Разработчик

- Возможности разработчик отождествляются с возможностями фреймворка, который он использует, поэтому среди технических требований не должно быть таких, которые невозможно реализовать в этих условиях.

Основная функция блога – доставлять информацию от человека, ведущего блог к его читателям

1.2 Конструкционные ограничения архитектуры

Опишем какие основные части составляют блог.

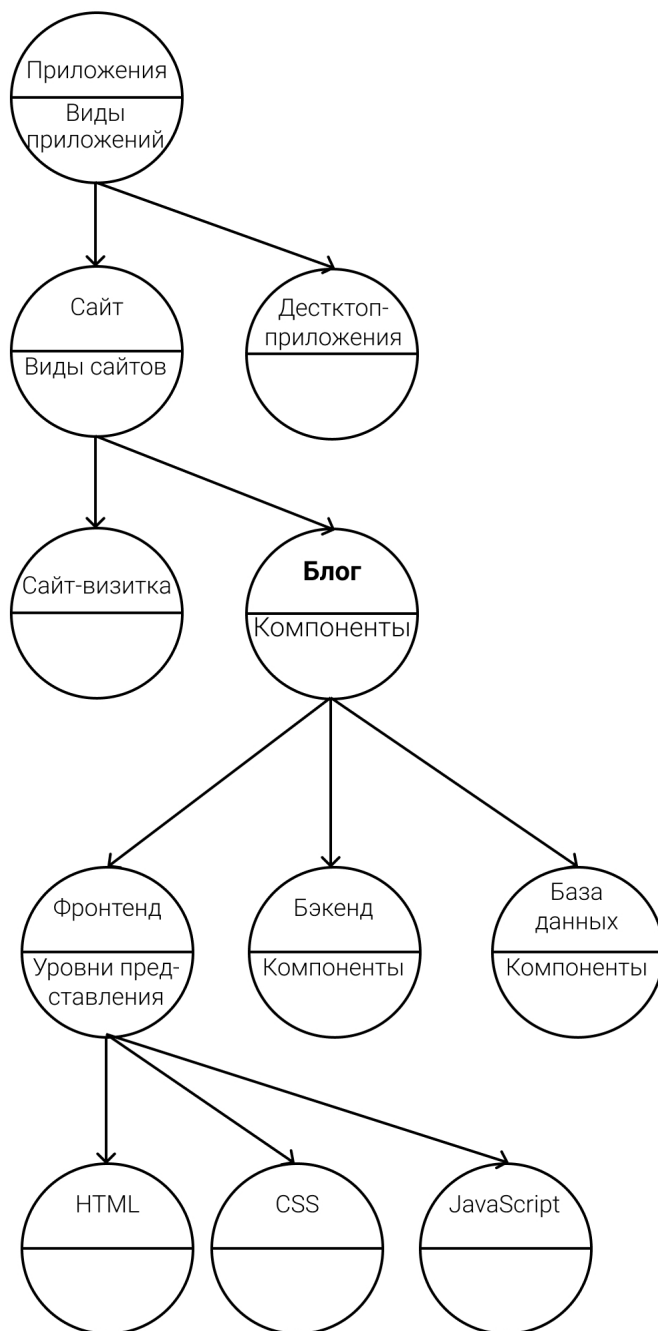


- **Фронтенд** – клиентская часть, открываемая как сайт в браузере пользователя. Фронтенд содержит в себе визуальное отображение страниц, стили, картинка (не все), страницы без контента.

- **Бэкенд** – серверная часть, обрабатывающая запросы фронтенда, взаимодействующая для этого с базой данных и возвращающая клиенту необходимую для работы информацию.
- **База данных** – более глубокая серверная часть, содержащая динамическую информацию сайта: учётные записи, посты, лайки, комментарии.

1.3 Описываемая система как холон в иерархии

Рассмотрим какое место может занять рассматриваемая система как холон в иерархии других холонов – холархии.



1.4 Системы: целевая, обеспечивающая, в эксплуатационной среде

Целевая система Целевой системой будет являться сам блог.

Обеспечивающая система

- Интернет, вероятнее всего с помощью которого будут общаться заказчик и разработчик.
- Компьютеры заказчика и разработчика
- Разработчик
- Фреймворк (то есть программный комплекс, предоставляющий готовые модули для сборки приложения)
- Другие необходимые для работы приложения библиотеки

Система в эксплуатационной среде

- Серверы, то есть компьютеры, на которых размещён сам сайт либо целиком (фронтенд, бэкенд, база данных), либо частично, тогда используется несколько серверов.
- Опять-таки Интернет, но теперь с помощью него содержимое сайта будет доставляться к читателям блога и клиентам. А если фронтенд и бэкенд будут на разных серверах то и для связи между ними.
- DNS-сервер, то есть специальная служба позволяющая сопоставить доменное имя (напр. google.com) с IP-адресом фронтенд сервера (именно фронтенд, потому что фронтенд являться точкой входа для клиента)

Опишем также людей, которым предстоит взаимодействовать с системой, они также будут являться частью системы в эксплуатационной среде:

- Клиенты сайта: постоянные читатели блога или люди зашедшие первый раз).
- Администратор или модератор (человек, который отвечает только за контент сайта). Их работа, как уже сказано будет производиться в админской части сайта.

2 Жизненный цикл системы согласно V-диаграмме

2.1 Определение требований

На данной стадии происходит сбор и анализ требований к блогу, которые формулируются стейкхолдерами. Требования могут касаться:

- Какие разделы желает заказчик видеть доступными для пользователя и насколько сайт может быть изменям. Будет ли это конструктор из отдельных виджетов или структура будет жёстко закреплена и потребуются больше времени и навыков для внесения изменений.
- Нагрузки на сайт. На какое среднее количество посетителей должна быть рассчитана мощность серверов. Здесь же можно отметить отказоустойчивость (сайт должен быть доступен для каждого кто на него зайдёт)
- Предполагаемой целевой аудитории. Если блог будет посвящён бизнес-тематике, то вероятно правильнее будет выбрать спокойный, однотонный дизайн без резких акцентов. И, наоборот, если, например, это блог художника-граффитиста, то логично использовать яркие цвета в молодёжном стиле.
- Потенциал роста. Если предполагается, что сайт будет расти в дальнейшем и потребуется добавить новый функционал, напριме возможность выкладывать фотографии для фотогалереи, разработчику следует учесть возможное усложнение архитектуры в будущем.

2.2 Архитектурное проектирование

Решается какие разделы будут на сайте, опеределаются сущности блога (напр. посты, комментарии, фотографии, лайки) и взаимоотношения между ними (посты имеют комментарии, а фотографии лайки).

2.3 Рабочее проектирование

Теперь, когда определены разделы и сущности, настает времени детальной проработки компонентов. Для фронтенда описывается роутинг, производится разбиение на компоненты. Для бэкенда определяются запросы API. А для базы данных описываются таблицы и взаимоотношения между ними.

2.4 Изготовление

Здесь специалисты в своих областях воплощают части блога согласно проекту. Во фронтенде делается верстка на HTML, пишутся стили на CSS и описывается логика на JavaScript. Бэкенд-разработчики создают работающее API, таким образом связывая фронтенд и базу данных и они же создают таблицы в базе данных и с помощью внешних ключей описываются взаимоотношения между таблицами. По окончании создания компоненты тестируются специалистами.

2.5 Интеграция

Здесь стоит отметить интересную особенность разработки программного обеспечения. Благодаря лёгкости создания, внесения изменений в продукт интеграция может осуществляться в течение всей разработки, с самого её начала (что невозможно в реальном промышленном производстве, где компонент должен быть полностью закончен для того чтобы его можно было интегрировать в систему). То есть проще говоря, например, фронтенд-разработчик может сделать одну страницу с одной динамической сущностью, бэкенд описать API на одну сущность и сделать одну таблицу. И уже вот этот очень маленький комплекс может считаться сайтом. То есть как уже было сказано, интеграция производится часто, по мере расширения функционала в частях.

2.6 Приём в эксплуатацию и эксплуатация

После окончания внедрения всех функций и исправления ошибок, система проверяется и отдаётся заказчику.

3 Практики системной инженерии

В стандарте ISO 15288:2008 указаны 25 обязательных практик системной инженерии. Опишем технические практики, при этом отметим некоторое пересечение с этапами V-диаграммы

3.1 Сбор требований

Здесь полностью повторяется пункт «Определение требований» V-диаграммы, то есть в первую очередь заказчик сообщает исполнителю о желаемом функционале.

3.2 Анализ требований

При этом исполнитель может посоветовать изменить часть требований по разным причинам (из-за несовместимости, невозможности реализации)

3.3 Архитектурный дизайн

Эта практика объединяет в себе этапы «Архитектурное проектирование» и «Рабочее проектирование» V-диаграммы. Можно только добавить, что из-за относительной лёгкости внесения изменений в проект, проектирование не занимает слишком большого времени (конечно, всё зависит от сложности системы), но чаще всего ограничиваются функционалом, описанием страниц, сущностей, всё это составляет проект и переходят к уже непосредственному воплощению в коде.

3.4 Изготовление

Здесь полностью повторяется этап «Изготовление». Добавим, что в разработке ПО как правило используются спринты – то есть отведённые участки времени (например 2 недели) на которые ставится ряд задач. По окончании спринта проверяется завершённость заданий, если что-то не сделано, оно переходит на следующий спринт. Для каждого разработчика при этом важно проводить своё, так называемое юнит-тестирование, то есть проверка маленьких частей приложения.

3.5 Интеграция

То же, что и «Интеграция» в V-диаграмме, – интеграция происходит с самых ранних этапов разработки, так что не будет пикового момента, когда компоненты долго разрабатывались по отдельности и в короткое время собираются в одну систему.

3.6 Проверка всей системы

В разработке ПО этот процесс называется интеграционным тестированием. По мере готовности какого-то логически связанного блока функций, проводится интеграционное тестирование. То есть как и для самой интеграции на весь проект интеграционных тестов может быть достаточно много. Если интеграционный тест выявляет несовпадение с проектом, создаётся задача по исправлению ошибки.

3.7 Переход к эксплуатации

По мере изготовления сайт показывается заказчику, после последнего интеграционного теста, может быть объявлено закрытое тестирование среди ограниченного круга пользователей.

3.8 Валидация

Выявленные на предыдущем этапе ошибки отправляются на доработку, но обычно на этом этапе не выявляются серьёзных ошибок (конечно при условии хорошего разрабатки ранее)

3.9 Эксплуатация

Система отдаётся заказчику, он получает пароли от сервера, базы данных, собственно админки. Пользователи могут заходить на сайт и читать блог.

3.10 Обслуживание

Со стороны производителя В течение эксплуатации у заказчика могут возникнуть вопросы по системе. Или на этом этапе может быть использован SEO, то есть оптимизация сайта в поисковой выдаче (в идеале если написать "блог" в google то наш сайт должен быть как можно выше)

Со стороны администратора или модератора Ведущему блог появляется о чём рассказать, добавляются посты, фотографии. Пользователи делятся и комментируют

3.11 Вывод из эксплуатации

Любой сайт живёт пока его смотрят и читают. Если по ряду обстоятельств сайт не набрал или потерял популярность, владелец может решить забросить его. Это будет выражаться в отсутствии обновлений, а если владелец перестанет платить за хостинг и домен, то сервер будет очищен и передан другому пользователю как и доменное имя.