Выполнила: Гракович Ю.Д. группа 22919/1

Название программы АндБрейн/ AndBrain (состоит из 2-ч слов android и brain).

Заказчик проекта – это владелец крупной сети офлайн супермаркетов, специализирующихся на продаже, ремонте и обслуживании компьютерной техники. Он хочет удобную и безопасную для продаж программу (веб-сайт).

Соответственно задачей является создание интернет-магазина компьютерной техники. В нём можно будет купить нужные комплектующие для компьютера или вызвать мастера на дом.

**Предметная область.**

Планируется примерно 2 уровня доступа: для будущих покупателей (совершение покупок), для администрации (редактирование страниц сайта, например, смена цен, смена продукции, показ резюме компьютерных мастеров и т.д.).

Большой объём базы данных. Предполагаемый объём нагрузки: 100000/200000 пользователей могут заходить одновременно. Примерно 400000 личных кабинетов. Более 100000 различных товаров.

Программу необходимо обеспечить защитой. Защитить нужно: сам сайт от DDos атак, личные данные пользователей (телефоны, любые банковские карты) от утечки информации, защита уровня доступа администрации от взлома (например, чтобы мошенники не смогли изменить цены на товар или написать от лица компьютерного мастера покупателю).

Планируемые системы: регистрация пользователей, система поиска товаров, различные каталоги сортировки (например, по ценам, по фирмам), система «добавить в избранное», система заказа, система оплаты, система отзывов, система вызова мастера…

Существующий аналог: Ozon (как аналог интернет-магазина), DNS.

**Группа разработки состоит из 14 человек**:

Менеджер проекта; 2 архитектора: системный архитектор, технический архитектор; 4 разработчика: Back-end разработчик, Front-end разработчик, инженер баз данных, DevOps программист; тестеровщик; технический писатель; бизнес-аналитик; релиз-менеджер; дизайнер; специалист по информационной безопасности; аналитик по угрозам.

Количество ролей довольно большое. В команде разработки работают люди разного уровня опыта и квалификации. Среди наиболее опытных: Back-end разработчик, Front-end разработчик, тестеровщик, технический писатель, менеджер проекта, бизнес-аналитик, релиз-менеджер. Среди наименее опытных: системный архитектор, технический архитектор, инженер баз данных, DevOps программист, дизайнер, специалист по информационной безопасности, аналитик по угрозам. Заказчик ответственный, он хорошо владеет знаниями в своей сфере, но большинство его требований размыты и неясны. Клиент может выходить на связь 1-2 раза в неделю. Сроки довольно мягкие. Функционирование предметной области не простое и предполагает высокую надёжность, так как программа будет связана с некоторыми личными данными пользователей.

Сильные стороны ПО: опытные Back-end разработчик, Front-end разработчик, тестеровщик, технический писатель, менеджер проекта, дизайнер, релиз-менеджер; заказчик может выходить на связь. Мягкие сроки; есть известные аналоги.

Слабые стороны ПО: неопытные архитекторы, инженер баз данных, DevOps программист, бизнес-аналитик, специалист по информационной безопасности, аналитик по угрозам. Неполнота и неясность требований заказчика из-за чего невозможно составить чёткий план и график. Некоторый функционал сайта труднореализуем, например, возможность связаться с мастером без использования интернета. Так же для большой базы данных необходимы большие сервера.

В связи с особенностями условий проекта для его выполнения подходят 3 жизненных цикла: спиральная, инкрементная, прототипирование.

**Спиральная**

|  |  |
| --- | --- |
| Плюсы | Минусы |
| Отражает эволюционный характер проектирования. | Высокие требования к разработчику и заказчику. |
| Позволяет учитывать риски на каждом витке. | Трудность контроля времени. |
| Используется результат моделирования. |

Обоснование

Спиральная модель жизненного цикла не подходит из-за высоких требований, как к заказчику, так и к разработчикам. В данном случае заказчик не обладает высокими знаниями в онлайн сфере. В команде разработки есть неопытные специалисты, в особенности системный архитектор, технический архитектор, инженер баз данных, DevOps программист, специалист по информационной безопасности, аналитик по угрозам. Неопытность этих специалистов может значительно затруднить создание программы по данному довольно сложному жизненному циклу.

**Инкрементная**

|  |  |
| --- | --- |
| Плюсы | Минусы |
| Есть план и график по всем этапам. | Часто всех требований в реальном проекте нет на начальном этапе, а менять их модели нельзя. |
| Промежуточные версии доступны заказчику. | Не всегда можно спланировать содержание версии. |
| Необходимость подробного описания требований на начальном этапе (требования берутся 1 раз) |
| Тенденция оттягивания решения сложных проблем на последние итерации. |

Обоснование

Реализация данной модели предполагает изначально обозначить все требования. Заказчик ещё до конца не определился со всеми своими требованиями. Тем более он может их кардинально изменить во время создания проекта, а изменять модели требований в этом жизненном цикле нельзя. Соответственно инкрементная модель жизненного цикла не подходит из-за непредсказуемости требований заказчика.

**Прототипирование**

|  |  |
| --- | --- |
| Плюсы | Минусы |
| Обеспечивает определение полных требований к ПО. | Не является полноценным жизненным циклом. |
| Повышение качества требований. | Возможность предпочтения заказчиком прототипа вместо конечной версии. |
| Увеличение времени на написание документации. |

Обоснование

Данная модель жизненного цикла больше всего подходит для реализации проекта. Из особенностей условий мы знаем, что заказчик готов выходить на связь 1-2 раза в неделю, а данная модель жизненного цикла как раз предполагает общение с клиентом. После создания нескольких прототипов можно окончательно выяснить все требования заказчика. Вследствие чего можно составить чёткий план и график дальнейшей работы. А что бы сделать окончательную версию проекта можно объединить прототипирование с другим лёгким жизненным циклом, например с классической (каскадной) моделью. Главное убедить заказчика, что прототип не является конечным продуктом.

**Прототипирование**

Сбор и анализ требований.

Виды деятельности: идентификация требований, оценка требований, регистрация требований спецификация требований, оценивание требований, анализ требований к программным средствам.

Быстрое проектирование.

Виды деятельности: создание архитектуры, оценивание архитектуры, стратегия реализации программных средств, проектирование архитектуры программных средств.

Построение макета.

Виды деятельности: кодирование, квалификационное тестирование

Проверка заказчиком.

Вид деятельности: тестирование

Уточнение требований.

Виды деятельности: идентификация требований, оценка требований, регистрация требований спецификация требований, оценивание требований, анализ требований к программным средствам.

Конструирование.

Вид деятельности: конструирование программных средств.

Конструирование

Проверка заказчиком

Построение макета

Уточнение требований

Быстрое проектирование

Сбор и анализ требований