1. Тема веб-сайта: сервис покупки компьютерных игр.
2. Предметная область: веб-сайт предоставляет возможность приобретения компьютерных игр. На сайте будут публиковаться статьи о новинках игровой индустрии, обзоры игр, а также сами игры для покупки и загрузки. Кроме того, пользователи смогут оставлять комментарии и оценки к играм. Заказчик получит широкий спектр возможностей для использования данного ресурса, который ориентирован на англоязычную аудиторию, так как сайт будет иметь сложный функционал для администратора, но простой и удобный интерфейс для пользователей с ограничением ≤ 100 000. Особенностью заказчика является его богатый опыт в создании и продвижении веб-сайтов, а также то, что он является гражданином США. Надежность сайта подтверждается двухфакторной аутентификацией для всех пользователей, а также возможностью сохранения данных при сбое в работе серверов. Срок разработки данного сервиса — от года до полутора лет.
3. Роли для группы разработки:
   1. Проектный менеджер
   2. Аналитик
   3. Web дизайнер
   4. Верстальщик
   5. Frontend-разработчик
   6. Backend-разработчик
   7. Тестировщик
   8. DevOps-инженер
   9. Контент-менеджер
   10. Маркетолог
   11. Технический писатель
4. Особенности клиента: так как клиент является гражданином США и имеет богатый опыт в создании и продвижении веб-сайтов, то это говорит о том, что сайт будет рассчитан на англоязычную аудиторию с доступным функционалом для пользователей и понятным внутренним функционалом для заказчика.

Количество и качество ролей: Команда разработки состоит из 11 человек. Качество ролей зависит от опыта, навыков и компетенций каждого участника. Основными и необходимыми для успешной разработки проекта являются роли проектного менеджера, аналитика, верстальщика и разработчиков. Дополнительные и специализированные роли, такие как тестировщик, DevOps-инженер, контент-менеджер, маркетолог и технический писатель, могут варьироваться в зависимости от требований проекта. Качество ролей определяется знаниями, навыками, опытом и эффективностью каждого члена команды.

Сложность функционала ПО и требования к надежности: функционал ПО должен быть достаточно простым, чтобы обеспечить удобство использования для пользователей. Требования к надежности высокие, так как сайт должен быть доступен круглосуточно.

Сильные и слабые стороны предметной области:

«+» Высокий интерес к компьютерным играм: это означает, что у сервиса будет готовая аудитория, активно использующая функционал сайта. Высокий спрос на продукт поможет в его успешной разработке и продвижении.

«+» Доступность и простота функционала: это позволит пользователям легко приобретать, оценивать и комментировать игры, что может стимулировать их активность и повысить популярность платформы.

«-» Сильная конкуренция на рынке: существует множество уже известных сервисов по продаже игр с устоявшейся пользовательской базой. Новому сервису может быть сложно привлечь пользователей и конкурировать с ними.

«-» Сложности с монетизацией: если пользователи привыкли к использованию бесплатных или дешевых сервисов, они могут оказаться не готовы платить за аналогичный функционал на новом сайте. Это может затруднить монетизацию сервиса и обеспечение его финансовой устойчивости.

1. Три модели жизненного цикла для данного проекта:
   1. Каскадная модель
   2. Итерационная модель
   3. Agile-модель
2. Плюсы и минусы выбранных моделей:

«+» Каскадной модели:

1. Структурированность: каскадная модель предоставляет четкую и линейную последовательность этапов разработки.

2. Хорошая документация: каждый этап в каскадной модели должен быть документирован, что делает проект более понятным и удобным для последующего сопровождения.

3. Предсказуемость: каскадная модель позволяет легче планировать и прогнозировать время и ресурсы, необходимые для завершения проекта.

«-» Каскадной модели:

1. Жесткость: каскадная модель не предусматривает возможности внесения изменений после завершения каждого этапа, что может вызвать проблемы, если требования меняются или нужно внести коррективы.

2. Ограниченная коммуникация: в каскадной модели коммуникация между разработчиками и заказчиками происходит в основном на начальных этапах, что может привести к непониманию и неправильному истолкованию требований.

3. Долгий цикл: каскадная модель может иметь долгий цикл разработки, особенно если в ходе выполнения обнаруживаются проблемы или необходимы корректировки.

«+» Итерационной модели:

1. Гибкость: итерационная модель позволяет вносить изменения и корректировки, основываясь на обратной связи и опыте каждой итерации.

2. Улучшенная коммуникация: итерационная модель предусматривает регулярные обновления и демонстрации промежуточных результатов, что способствует более активной коммуникации между разработчиками и заказчиками.

3. Быстрое обнаружение проблем: итерационная модель позволяет быстрее выявлять и исправлять ошибки и проблемы, так как каждая итерация включает проверку и тестирование результатов.

«-» Итерационной модели:

1. Высокая сложность планирования: итерационная модель требует более сложного планирования проекта из-за возможности внесения изменений и необходимости учета фидбэка.

2. Высокие затраты на коммуникацию: регулярные совещания и обновления в итерационной модели могут стоить времени и ресурсов компании.

3. Возможность нежелательных изменений: если необходимые корректировки и изменения не отслеживаются и не контролируются должным образом, это может привести к нежелательным изменениям в проекте.

«+» Agile-модели:

1. Гибкость: Agile-модель позволяет быстро адаптироваться к изменениям требований и рыночным условиям.

2. Высокая коммуникация: Agile-модель ставит коммуникацию в центр разработки, что способствует более эффективному взаимодействию между всеми участниками проекта.

3. Постоянная обратная связь: Agile-модель предусматривает регулярные проверки и демонстрации работающего продукта, что позволяет быстро получать обратную связь.

«-» Agile-модели:

1. Недостаток документации: Agile-модель развивается на основе итераций и упорства на достижение поставленных целей. В результате часто недостаточное внимание уделяется документации, что может затруднять последующую поддержку и обслуживание проекта.

2. Потеря времени из-за неопределенности требований: Agile основана на изменчивых требованиях, которые могут меняться в процессе разработки. В связи с этим, возникает риск потери времени на перепланирование и изменение направления проекта, особенно если требования постоянно меняются.

3. Недостаток контроля над проектом: Agile-модель предлагает большую степень автономии команде разработчиков, что может привести к потере контроля над проектом со стороны управленческой команды. Это может стать проблемой, если внутренняя команда пренебрегает установленными сроками или плохо управляет ресурсами.

Нами была выбрана Agile-модель в пользу её гибкости и адаптивности, а также постоянного взаимодействия между участниками проекта. Она больше всех подходит для данного проекта, в связи с постоянными изменениями требований в процессе разработки.

1. Виды деятельности:
   1. Установление консультативного органа до начала основных работ
   2. Определить процедуру управления изменениями в контракте
   3. Документировать и объяснить спецификации, причины и исходные данные при запросе на изменение контракта
   4. Исследовать влияние изменений на проектные планы, затраты, выгоду, качество и график работ
   5. Переговоры и достижение наиболее приемлемых решений, рассмотрев содержание изменений и их воздействие на проект
   6. При необходимости согласовать проблему на высшем уровне руководства
   7. Документировать взаимное соглашение и модифицировать контракт
   8. Зафиксировать влияние модификации на конфигурацию системы
   9. Отразить результат модификации в проектных планах и сообщить об этом всем заинтересованным сторонам.
2. Этапы:

Этап 1: Планирование

* Установление консультативного органа до начала основных работ
* Определить процедуру управления изменениями в контракте

Этап 2: Проектирование

* Документировать и объяснить спецификации, причины и исходные данные при запросе на изменение контракта

Этап 3: Создание прототипа

* Исследовать влияние изменений на проектные планы, затраты, выгоду, качество и график работ

Этап 4: Тестирование

* Переговорить и прийти к наиболее приемлемым заключениям, рассмотрев содержание изменений и их воздействие на проект

Этап 5: Обратная связь

* При необходимости согласовать проблему на высшем уровне руководства

Этап 6: Запуск

* Документировать взаимное соглашение и модифицировать контракт
* Зафиксировать влияние модификации на конфигурацию системы
* Отразить результат модификации в проектных планах и сообщить об этом всем заинтересованным сторонам.

