**Практическая работа**

Выбор модели жизненного цикла

Выполнил:

Васильев Роман

Гр. 22919/2

**Тема**: Ритуальные услуги

**Работники**:

1. **Fronted – разработчик** – создание дизайна, интерфейса, которые удовлетворяют запросы заказчика
2. **Backend – разработчик** – создание баз данных, настройки их
3. **Инженер по информационной безопасности** – создание защиты конфиденциальной информации
4. **Тестер** – тестирование сайта
5. **Pentester** – тестирует систему безопасности сайта
6. **Бизнес – аналитик** – анализ данных по данной теме, сайты конкурентов, востребованность услуг
7. **Teamlead** – руководит разработкой
8. **User support** – Поддержка и выяснение проблем при использовании сайта пользователями

**Предметная область:**

*Местонахождение*: Россия. Выбор обоснован тем, что это крупнейшая страна по территории и с большим населением, а также значительными различиями в стоимости земли, включая высокие цены на участки для захоронений в ряде регионов. Это означает, что будет большое количество потенциальных клиентов по всей стране.

*Объём нагрузки на сайт:* так как люди интересуются ритуальными услугами только после смерти близких людей, либо же для просмотра цен для себя, то нагрузка на сайт будет незначительной, в среднем до 1000 человек в час.

*Уровни доступа*: на сайте можно просматривать представляемые услуги в режиме «гость», но для оформления заказа понадобиться зарегистрироваться. Также будет уровень «администратор», который сможет редактировать любой товар и каждую страницу сайта.

*Уровень защиты сайта:* Степень защиты должна быть высокой. В неё предполагается комплексная безопасность, включающая защиту от утечки персональной информации — такой как ФИО клиентов, пароли от учётных записей и банковские данные. Также должна быть обеспечена защита от DDoS-атак для стабильной работы ресурса, а все платежные операции должны осуществляться с использованием надёжных протоколов шифрования и интеграции с проверенными платёжными системами.

*Аналог проекта*: Аналогом нашего сайта является «Городская ритуальная служба».

*Заказчик:* Нашим заказчиком выступает компания «Ритуальные услуги в СПб», которая имеет собственный сайт, но старого образца, поэтому хотела бы сделать новый для внедрения новых функций. Благодаря чему у заказчика уже есть готовое ТЗ, в которое в будущем хотели бы внести небольшие правки. У компании, заказывающей у нас услугу разработки, нет чётких рамок на время получения продукта, поэтому рамки были выставлены в 2 месяца с возможностью продления срока разработки. У данной кампании было условие, что окончательная ценна за продукт будет только, когда web-сайт будет готов, поэтому была оговорена минимальная сумма 200.000 руб. (двести тысяч рублей). Так же у нашего заказчика было два пожелания: возможность связаться с нами для внесения корректировок в проект, и возможность получения промежуточных версий web-сайта.

У данной компании были предпочтении и пожелания в реализации проекта:

* Лента меню с их услугами: Главная страница, на которой написана краткая информации о предоставление их услуг; «Услуги», список предоставляемых ими услуг; «Ритуальные товары»; «Справочная информация»; «Контакты», можно получить данные для связи; «Вызвать агента», как вызвать и краткая информация о данной услуге; «Отзывы»; «Акции»; «Вход» и «Регистрация»
* Возможность настройки ритуальных товаров под конкретного человека.
* Настройки для слабовидящих.

**Сильные и слабые стороны предметной области:**

*Сильные:*

1. Все сотрудники с богатым опытом работы в сфере разработки web-сайтов для разных задач.
2. Есть ТЗ, которое предоставил заказчик во время переговоров с ним.
3. Возможность коммуникации с заказчиком в любое время для внесения правок.

*Слабые:*

1. Нет чётко поставленных рамок по времени, из-за чего сдача продукта может переноситься на неопределённый срок.
2. Недостаточное количество кадров для разработки, таких как архитектор и др.
3. Многие сотрудники должны будут выполнять не только свои задачи, но и вторичные, по той же причине, которая была описана в п.2.
4. Так как бюджет на стороне заказчика, наша команда должны уложиться в минимальные сроки, чтобы получить деньги за проделанную работу.

**Модели жизненного цикла**

1. Каскадная модель

*Плюсы:*

* Возможность выполнения задачи в сжатые сроки
* Ход работы упорядочен, и так как наша команда уже имеет большой опыт, для неё не составит труда распределить работу между собой.

*Минусы:*

* Невозможность внесения правок во время разработки, так как модель не предоставляет возможности, в нашем случае заказчику требуется связываться с нами для внесения правок.
* При возникновении проблем нужно будет начинать с пункта, где появилась проблема, что не подходит для нашей маленькой команды.
* Результат данного проекта будет только в конце, что не подходит, так как наш заказчик пожелал иметь промежуточные варианты.

1. Инкрементная модель

*Плюсы:*

* Возможность предоставления заказчику промежуточных вариантов, что снижает риск разногласий между нашей командой и заказчиком.
* Возможность выявления проблем на стадии разработки благодаря промежуточным вариантам продукта.
* Нет жестких временных рамок для выполнения заказа.

*Минусы:*

* Недостаточно чёткое определение требований на старте, что критично для нашего проекта.
* Создаётся несколько разных модели сайта, что увеличивает трудозатраты.
* Не всегда можно распланировать содержание следующих промежуточных версий.

1. Спиральная модель

*Плюсы:*

* Раннее выявление и минимизация рисков благодаря цикличности разработки.
* Не четкие рамки выполнения работы, этапы разработки повторяются друг за другом что позволяет воплотить желания заказчика на должном уровне
* Возможность связи с заказчиком неограниченное количество раз, что позволяет вовремя вносить правки и избегать разногласий с заказчиком.
* Возможность регулярной оценки затрат, что приводит к их сокращению. Так как конечную цену выставляет заказчик.

*Минусы:*

* Большой объём документации, что сложно для нашей небольшой команды.
* Существуют трудности контроля времени, а деньги за проект получим только в конце.

**Предпочтительная модель**

После детального анализа всех моделей жизненного цикла наша команда пришла к выводу, что спиральная модель является наиболее подходящей для данного проекта. Выбор обоснован тем, что на этапе проектирования можно выявить и минимизировать риски, что снижает вероятность серьёзных проблем в дальнейшем. Гибкость в работе – возможность вносить правки и корректировки в процессе разработки, что критично важно для нашего заказчика. Немало важный плюс это – поэтапная сдача промежуточных результатов, позволяющая заказчику оценивать прогресс и вносить предложения. Отсутствие жёстких временных рамок, что даёт больше свободы в планировании, но одновременно требует чёткого контроля сроков.

Но также существуют и недостатки: сложность оценки точных сроков завершения, что может затруднить планирование; большой объём документации, который потребует дополнительных ресурсов от нашей небольшой команды.

Однако, по сравнению с каскадной (жёсткость структуры, отсутствие гибкости) и инкрементной (нечёткие требования, дублирование работы) моделями, минусы спирального подхода менее критичны. Главное – он позволяет учесть ключевые требования заказчика: гибкость, промежуточные результаты и возможность доработок.

**Модель проекта**



*1 этап. Анализ требований и рисков:*

* 6.3.1.3.1.1 Менеджер должен определять требования инициируемого проекта.
* 6.3.1.3.1.3 По мере необходимости и при согласии всех заинтересованных сторон требования проекта могут быть изменены на этом этапе для достижения критериев завершения.
* 6.3.3.3.2.1 При реализации проекта необходимо выбирать и объявлять стратегию принятия решения для каждой ситуации, в которой необходимо принять решение. Проект должен определять желаемые результаты и измеримые критерии успешного принятия решений.
* 6.3.4.3.1.1 Должны быть определены политики менеджмента рисков, описывающие руководящие указания, регламентирующие выполнение менеджмента рисков.
* 6.3.4.3.2.2 Должны быть документированы пороговые значения риска, определяющие условия, при которых уровень риска может быть принят.
* 6.3.4.3.3.1 Риски должны быть идентифицированы в категориях, описанных в контексте менеджмента рисков.
* 6.3.4.3.3.2 Должна быть оценена вероятность возникновения и последствия каждого идентифицированного риска.
* 6.3.4.3.4.1 Правообладатели должны предоставлять рекомендованные альтернативы обработки риска в требованиях на действия по отношению к риску.
* 6.3.4.3.5.1 Все риски и содержание менеджмента рисков должны подвергаться мониторингу для выявления изменений. В случае изменения состояния риска должна быть выполнена его оценка.
* 6.4.1.3.2.1 Должны быть выявлены требования правообладателей проекта.
* 6.4.1.3.2.5 В проекте необходимо установить требования к здоровью, безопасности, защищенности, окружающим условиям и другие требования и функции правообладателей, имеющие отношение к критическим свойствам. При этом необходимо учесть возможные неблагоприятные воздействия использования системы на здоровье и безопасность человека.
* 6.4.1.3.3.1 В проекте необходимо анализировать полную совокупность выявленных требований.
* 6.4.1.3.4.2 В проекте должна предусматриваться обратная связь от проанализированных требований к соответствующим правообладателям для гарантии того, что их потребности и ожидания были правильно зафиксированы и выражены.
* 6.4.2.3.1.1 Должны быть проанализированы особенности планируемого применения разрабатываемой системы для задания системных требований
* 6.4.9.3.5.1 Оператор должен направлять возникшие проблемы в процесс решения проблем в программных средствах для их устранения.
* 6.4.10.3.1.1 Сопровождающая сторона должна разрабатывать, документировать и выполнять планы и процедуры проведения действий и решения задач в рамках процесса сопровождения программных средств.
* 6.4.10.3.2.1 Сопровождающая сторона должна анализировать отчеты о проблемах или заявки на модификацию для определения воздействий на организацию, существующую систему и связанные с ней системы.
* 6.4.10.3.3.1 Сопровождающая сторона должна провести анализ и определить, какая документация, программные модули и какая из версий нуждаются в модификации.
* 6.4.11.3.1.1 Определяется и документируется стратегия прекращения применения программных средств.
* 7.1.2.3.1.2 Исполнитель должен оценить требования к программным средствам.

*2 этап. Проектирование:*

* 6.3.3.3.1.1 Проект должен определять стратегию принятия решений.
* 6.3.7.3.1.1 В проекте необходимо описать характеристики организации, проводящей измерения.
* 6.3.7.3.1.2 В проекте необходимо идентифицировать и распределить по приоритетам потребности в информации.
* 6.3.7.3.1.3 При реализации проекта должны быть выбраны и документированы единицы измерения, удовлетворяющие информационным потребностям.
* 6.3.7.3.1.4 В проекте необходимо определять процедуры сбора данных, анализа и представления отчетов.
* 6.4.3.3.1.1 Должен быть определен верхний уровень архитектуры системы. Архитектура должна идентифицировать составные части технических средств, программных средств и ручных операций. Должно гарантироваться, что все системные требования распределяются между этими составными частями.
* 6.4.10.3.1.1 Сопровождающая сторона должна разрабатывать, документировать и выполнять планы и процедуры проведения действий и решения задач в рамках процесса сопровождения программных средств.
* 6.4.10.3.1.2 Сопровождающая сторона должна определять процедуры получения, регистрации и прослеживания отчетов о проблемах, заявок на модификацию от пользователей и обеспечения обратной связи с пользователями.
* 6.4.10.3.2.3 Основываясь на анализе, сопровождающая сторона должна разработать варианты осуществления модификации.
* 6.4.10.3.3.2 Для осуществления модификации сопровождающая сторона должна принять участие в технических процессах.
* 6.4.11.3.1.1 Определяется и документируется стратегия прекращения применения программных средств.
* 7.1.1.3.1.1 Если не оговорено в контракте, разработчик должен определить или выбрать модель жизненного цикла, соответствующую области применения, размерам и сложности проекта.
* 7.1.1.3.1.4 Исполнитель должен разрабатывать планы проведения действий процесса реализации программных средств.
* 7.1.3.3.1.1 Исполнитель должен преобразовать требования к программным составным частям в архитектуру, которая описывает верхний уровень его структуры и идентифицирует программные компоненты.
* 7.1.3.3.1.2 Исполнитель должен разработать и документально оформить проект верхнего уровня для внешних интерфейсов программной составной части и интерфейсов между ней и программными компонентами.
* 7.1.3.3.1.3 Исполнитель должен разработать и документально оформить проект верхнего уровня для базы данных.
* 7.1.4.3.1.1 Исполнитель должен разработать детальный проект для каждого программного компонента программной составной части.

*3 этап. Реализация:*

* 6.3.3.3.3.1 При реализации проекта необходимо регистрировать, отслеживать, оценивать и сообщать о результатах принятия решений для подтверждения эффективности решения проблем, устранения отрицательных тенденций и получения возможных преимуществ.
* 6.3.3.3.3.2 При реализации проекта необходимо поддерживать записи о проблемах и возможностях их решения, а также размещать записи в соответствии с соглашениями или организационными процедурами таким образом, который позволяет проводить аудит и изучать полученный опыт.
* 6.3.4.3.5.2 Для оценки результативности обработки риска должны разрабатываться и контролироваться соответствующие измеримые показатели.
* 6.3.6.3.2.2 При реализации проекта необходимо сопровождать блоки информации и хранящиеся записи этих блоков в соответствии с требованиями к целостности, защищенности и секретности.
* 6.3.7.3.2.2 Данные при реализации проекта должны накапливаться, сохраняться и проверяться
* 6.4.10.3.1.1 Сопровождающая сторона должна разрабатывать, документировать и выполнять планы и процедуры проведения действий и решения задач в рамках процесса сопровождения программных средств.
* 6.4.10.3.3.2 Для осуществления модификации сопровождающая сторона должна принять участие в технических процессах.
* 7.1.1.3.1.5 При разработке или сопровождении программных продуктов могут применяться непостовляемые элементы.
* 7.1.3.3.1.1 Исполнитель должен преобразовать требования к программным составным частям в архитектуру, которая описывает верхний уровень его структуры и идентифицирует программные компоненты.
* 7.2.1.3.3.1 Документы должны изготавливаться и поставляться в соответствии с планом.
* 7.2.1.3.3.2 В соответствии с процессом менеджмента конфигурации программных средств должны быть установлены необходимые средства управления.
* 7.2.1.3.4.1 Должны выполняться задачи процесса сопровождения программных средств, которые необходимы при изменениях в документации.
* 7.2.3.3.1.4 Планируемые и осуществляемые виды деятельности и задачи обеспечения гарантии качества должны быть выполнены.

*4 этап. Тестирование:*

* 6.1.1.3.6.2 Приобретающая сторона должна провести приемочный осмотр и приемочное тестирование поставляемого программного продукта или услуги и должна принять их от поставщика, если все условия приемки удовлетворены.
* 6.4.5.3.2.1 Для каждого квалификационного требования системы должны быть разработаны и документированы: набор тестов, тестовые примеры (входы, выходы, критерии тестирования) и процедуры тестирования.
* 6.4.6.3.1.1 Квалификационное тестирование системы должно проводиться в соответствии с квалификационными требованиями, установленными для системы
* 6.4.6.3.1.2 Система должна быть оценена с учетом перечисленных ниже критериев:
  1. тестовое покрытие системных требований;
  2. соответствие ожидаемым результатам;
* 6.4.6.3.1.3 Разработчик должен поддерживать проведение аудитов.
* 6.4.6.3.1.4 После успешного окончания аудита (если он проводился) разработчик должен доработать и подготовить поставляемый программный продукт к инсталляции и поддержке его приемки.
* 6.4.8.3.1.1 Разработчик должен поддерживать ревизии и тестирование программного продукта, проводимые
* приобретающей стороной в процессе приемки.
* 6.4.9.3.2.1 Для каждого выпуска программного продукта оператор должен выполнить тестирование на соответствие функциональным требованиям и при условии удовлетворения заданных критериев выпустить программный продукт для применения по назначению.
* 7.1.4.3.1.5 Исполнитель должен определять и документировать требования к тестированию и графики работ по тестированию программных блоков.
* 7.1.4.3.1.6 Исполнитель должен обновлять требования к тестированию и графики работ по комплексированию программных средств.
* 7.1.5.3.1.2 Исполнитель должен тестировать каждый программный блок и базу данных, гарантируя, что они удовлетворяют требованиям. Результаты тестирования должны быть документально оформлены.
* 7.2.7.3.1.1 Аудиторские проверки должны проводиться в предварительно установленные контрольные сроки, указанные в плане (планах) проекта.
* 7.2.5.3.1.2 Если проект предусматривает работы по валидации, то должен быть установлен процесс валидации для подтверждающей проверки системного или программного продукта.
* 7.2.5.3.1.3 Если проект предусматривает независимые работы по валидации, то должна быть выбрана квалифицированная организация, ответственная за проведение работ.
* 7.2.5.3.1.5 План валидации должен быть выполнен.

*5 этап. Внедрение очередной версии системы. Релиз:*

* 6.3.2.3.4.1 Когда все программные продукты, действия и задачи завершаются, менеджер должен определить, закончен ли проект, принимая во внимание критерии, указанные в контракте или установленные как часть процедуры организации.
* 6.3.2.3.4.2 Эти результаты и отчеты должны быть архивированы в соответствующей среде, как определено в контракте.
* 6.4.7.3.1.2 Разработчик должен инсталлировать программный продукт в соответствии с планом инсталляции
* 6.4.8.3.1.1 Разработчик должен поддерживать ревизии и тестирование программного
* продукта, проводимые приобретающей стороной в процессе приемки.
* 6.4.9.3.1.3 Оператор должен устанавливать процедуры тестирования программного продукта в среде его эксплуатации для включения отчетов по проблемам, заявок на модификацию процесса сопровождения программных средств.
* 6.4.9.3.2.1 Для каждого выпуска программного продукта оператор должен выполнить тестирование на соответствие функциональным требованиям и при условии удовлетворения заданных критериев выпустить программный продукт для применения по назначению.
* 6.4.9.3.2.2 Оператор должен гарантировать, что программный код и база данных инициируются, реализуются и заканчивают свое действие, как указано в плане.
* 6.3.2.3.1 Мониторинг проекта
* 6.4.9.3.3.1 Система должна функционировать в предназначенной для нее среде согласно пользовательской документации.
* 6.4.9.3.4.1 Оператор должен обеспечивать содействие и консультации пользователей по их просьбе.
* 6.4.10.3.4.2 Сопровождающая сторона должна получить одобрение удовлетворительного завершения модификации, как определено в контракте.
* 6.4.11.3.2.1 Должен исполняться план прекращения применения программных средств.
* 7.1.2.3.1.3 Исполнитель должен проводить ревизии в соответствии с 7.2.6.
* 7.1.3.3.1.7 Исполнитель должен проводить ревизии в соответствии с 7.2.6.
* 7.1.4.3.1.8 Исполнитель должен проводить ревизии в соответствии с 7.2.6.