**Структура 3 главы ВКР направления Программная инженерия**

На ИТ направлениях подготовки бакалавриата ВВГУ в основном принята (рекомендована, но не является строго обязательной) типовая структура ВКР, состоящая из трёх глав. В первой главе даётся обзор предметной области и постановка задачи, мотивация и обзор существующих решений. Во второй главе даются модельные представления, методы и алгоритмы решения задачи, архитектура создаваемого программного средства. В третьей главе описывается процесс создания программы – требования, этапы разработки, тесты и результаты тестирования, примеры использования программы (в том числе, например, результаты исследования, если программа использовалась в качестве инструмента для исследований).

Направление подготовки Программная инженерия с точки зрения стандарта и представлений отраслевой среды отличается тем, что в нем углублённо изучаются процессы разработки программ: организация команд разработки, механизмы коллективной разработки (в том числе территориально распределёнными коллективами), средства контроля версий, разделение проекта на подзадачи, определение набора этапов работы над задачами, механизм верификации результатов работы (ревью, тестирование), оценка производительности работы над проектом и качества конечного продукта.

Особенности направления подготовки касаются главным образом третьей главы типовой структуры ВКР.

Для тех ВКР, где используется классический процесс разработки ПО (вариации водопадного подхода) рекомендуется следующая структура третьей главы.

* Описание инструментальных используемых средств работа над проектом (системы контроля версий, системы управления задачами и т.п.) и особенностей работы с ними в рамках проекта.
* Если проект коллективный – описание ролей на проекте и их распределения между участниками
* Перечень возможных состояний задач и правил перехода между ними.
* Описание регламента проведения инспекций (ревью).
* Описание программы измерений проекта (метрик оценки продуктивности работы и качества конечного продукта)
* Перечень требований к проекту
* Перечень задач проекта (code issue list)
* Описание рекомендаций по кодированию
* Перечень тестовых сценариев (с матрицей покрытия требований)
* Результаты тестирования
* Оценка метрик программы измерений и интерпретация их значений
* Примеры использования программы
* Технико-экономическое обоснование эффективности вложений в разработку

Для тех ВКР, где используется процесс разработки ПО на основе идей Agile и Scrum рекомендуется следующая структура третьей главы.

* Описание инструментальных используемых средств работа над проектом (системы контроля версий, системы управления задачами и т.п.) и особенностей работы с ними в рамках проекта.
* Перечень возможных состояний задач и правил перехода между ними.
* Описание регламента проведения инспекций (ревью).
* Описание программы измерений проекта (метрик оценки продуктивности работы и качества конечного продукта)
* Перечень пользовательских историй с оценкой их «стоимости». Здесь же описываются соответствующие историям тестовые сценарии
* Перечень задач проекта (code issue list)
* Описание рекомендаций по кодированию
* Результаты тестирования
* Конечная диаграмма сгорания задач
* Оценка метрик программы измерений и интерпретация их значений
* Примеры использования программы
* Технико-экономическое обоснование эффективности вложений в разработку

Глава 3

# 3.1 Описание ролей участников проекта

Для коллективных проектов.

# 3.2 Описание проектного инструментария

## 3.2.1 Работа с системой контроля версий

Краткая инструкция по работе с системой.

Особенности создания рабочих веток, слияния рабочих версий продукта и формирования релизных версий продукта.

## 3.2.2 Работа с системой управления задачами

Краткая инструкция по работе с системой.

Перечень возможных состояний задач и правил перехода между ними

# 3.3 Регламент проведения инспекций (ревью)

Для коллективных проектов описывается инструкция проведения инспекции рабочего продукта: как проверяющие приглашаются к проверке, как до них доводится проверяемая версия рабочего продукта, как они дают обратную связь и верифицируют изменения. Указывается на что нужно обращать внимание при инспекции в первую очередь.

Для индивидуальных проектов описывается чек-лист, которому должен соответствовать рабочий продукт (если есть несколько типов рабочих продуктов – разные чек-листы).

# 3.4 Программа измерений проекта

## 3.4.1 Метрики эффективности процесса разработки

## 3.4.2 Метрики качества рабочего продукта

# 3.5. Требования или пользовательские истории проекта

Для тех ВКР, где используется классический процесс разработки ПО (вариации водопадного подхода) формулируются требования. Хорошим тоном считается структурировать требования (по подсистемам или по типу) и вводить систему их нумерации. Требования должны подробно и конкретно раскрывать функционал продукта, как видимый, так и «под капотом». В идеале их должно быть достаточно для завершения разработки.

Для тех ВКР, где используется процесс разработки ПО на основе идей Agile и Scrum, роль требований играют пользовательские истории. Они имеют следующий шаблон:

КАК *роль пользователя* Я ХОЧУ *какой функционал иметь* ДЛЯ ТОГО ЧТОБЫ *что полезного даёт этот функционал*

Пользовательские истории желательно структурировать (по ролям пользователя, по области функционала, календарно и т.п.).

В идеале пользовательских историй должно быть достаточно для завершения разработки.

Некоторые функции или архитектурные особенности продукта не удаётся сформулировать в виде пользовательских историй. В этом случае их дополняют перечнем дополнительных требований к продукту.

Каждой пользовательской истории приписывается её «стоимость». Хорошей практикой считается оценивать стоимость в человеко-часах.

Для каждой пользовательской истории и дополнительного требования создаётся тестовый сценарий.

Вариант списка пользовательских историй в ВКР может быть таким:

|  |  |  |  |
| --- | --- | --- | --- |
| № истории | Формулировка истории | Стоимость истории | Номер тестового сценария |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| № теста | Описание тестового сценария | Номер покрываемых историй |
|  |  |  |
|  |  |  |

# 3.6 Перечень задач проекта (code issue list)

Требования и пользовательские истории раскрываются задачами (тикеты, таски, ишью), которые нужно реализовать в коде или данных. Обычно на одно требование или пользовательскую историю приходится несколько задач, в простых случаях – одна задача. Задача должна иметь небольшой объём реализации в коде (до 200 строк, но это не строгая рекомендация) и небольшой срок исполнения (неделя). Задачи имеют свою систему нумерации и всегда привязаны к какому-либо требованию или пользовательской истории. Именно задачи, результат их выполнения, оцениваются во время инспекции.

Вариант списка задач в ВКР может быть таким.

|  |  |  |  |
| --- | --- | --- | --- |
| №  требования или истории | № задачи | Краткое описание задачи | Подробное описание задачи |
| REQ\_UI\_1 | TASK\_UI\_1\_1 | Кнопка ВХОД | Добавить кнопку ВХОД на форму администратора системы |
| TASK\_UI\_1\_2 | Кнопка ВЫХОД | Добавить кнопку ВЫХОД на форму администратора системы |
| REQ\_UI\_2 | TASK\_UI\_2\_1 | Иконка СКОРОСТЬ | Добавить иконку СКОРОСТЬ на форму пользователя системы |
| TASK\_UI\_2\_2 | Иконка ОСВЕЩЁННОСТЬ | Добавить иконку СКОРОСТЬ на форму пользователя системы |
| TASK\_UI\_2\_3 | Иконка ДАЛЬНОСТЬ | Добавить иконку ДАЛЬНОСТЬ на форму пользователя системы |
| TASK\_UI\_2\_4 | Иконка ПОГОДА | Добавить иконку ПОГОДА на форму пользователя системы |
|  |  |  |  |
|  |  |  |

# 3.7 Описание рекомендаций по кодированию

Рекомендации по кодированию в больших организациях и на больших проектах могут представлять собой большой подробный документ. Представляется, что в ВКР их достаточно представить в виде чек-листа с пояснениями.

# 3.8 Перечень тестовых сценариев (с матрицей покрытия требований)

Этот пункт для тех ВКР, где используется классический процесс разработки ПО (вариации водопадного подхода), для ВКР на основе идей Agile и Scrum тесты уже сформулированы выше, вместе с пользовательскими историями.

|  |  |  |
| --- | --- | --- |
| № теста | Описание тестового сценария | Номер покрываемых требований |
|  |  |  |
|  |  |  |

Это – матрица покрытия требований тестами. Её можно использовать и для пункта 3.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тесты  Требования | TEST\_UI\_1 | TEST\_UI\_2 | TEST\_CX\_1 | TEST\_CX\_2 |
| REQ\_UI\_1 | x | x |  |  |
| REQ\_UI\_2 |  |  |  |  |
| REQ\_CX\_1 | x |  | x | x |
| REQ\_CX\_2 |  |  |  | x |

# 3.9 Результаты тестирования

Приводятся результаты тестирования, в простейшем случае – пройдено/не пройдено, в более сложных тестах приводятся наборы тестовых данных, характеризующих тот или иной тест.

|  |  |
| --- | --- |
| № теста | Результат тестирования |
|  |  |
|  |  |

# 3.9 Конечная диаграмма сгорания задач

Этот пункт для тех ВКР, где используется процесс разработки на основе идей Agile и Scrum.

По мере выполнения задач строится график, где по горизонтали откладывается время (например, недели), а по вертикали – число оставшихся задач (с учётом вновь добавленных в процессе).

На основе диаграммы нужно сделать о ходе процесса разработки.

# 3.10 Оценка метрик программы измерений и интерпретация их значений

Нужно оценить метрики, введённые в пункте 3.4 и сделать вывод об эффективности процесса разработки и качестве конечного продукта.

# 3.11 Примеры использования программы

Приводятся скриншоты, результаты расчётов и т.п.

3.12 Технико-экономическое обоснование эффективности вложений в разработку