

# **ОПИСАНИЕ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ**

Программное обеспечение  
Джет Сайбер Кэмп («Jet  
CyberCamp»)

## **Аннотация**

В документе приведено описание методологии и подходов, которые применяются в процессах жизненного цикла разработки программного обеспечения Джет Сайбер Кэмп («Jet CyberCamp») (далее – Система).

# Содержание

Аннотация .....	2
Содержание.....	3
Термины и определения.....	4
1 Описание процесса управления жизненным циклом.....	6
1.1 Инициация проекта .....	8
1.2 Исполнение проекта.....	8
1.3 Финальная стадия .....	10
1.3.1 Сдача-приемка готового решения .....	10
1.3.2 Гарантийные обязательства .....	10
1.4 Описание модели жизненного цикла Системы.....	11
Зона ответственности по ролям Scrum-модели .....	12
Создание беклога продукта.....	13
1.4.3 Планирование спринта и создание беклога спринта .....	14
1.4.4 Работа над спринтом. Ежедневные совещания .....	14
1.4.5 Демонстрация.....	14
1.4.6 Ретроспектива.....	14
1.4.7 Поставка.....	15
1.4.8 Сопровождение.....	15
1.4.8 Устранение неисправностей.....	15
2 Обоснование выбора модели жизненного цикла.....	16
3 Информация и персонале.....	17
Приложение А: Процессы разработки ПО.....	18
Приложение Б: Процессы сопровождения.....	19

# Термины и определения

В документе используются термины и определения, приведенные в таблице 1.

Таблица 1 – Термины и определения

Термин	Определение
Agile	Серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля
Scrum	Одна из нескольких методологий гибкой разработки ПО
SCRUM-команда	(Scrum team) Многофункциональная команда, которая является основой методологии Scrum, отвечает за разработку продукта итерациями
SCRUM-мастер	(Scrum master) Сотрудник, который организует работу команды проекта по методологии Scrum
Бета-версия	Построение Системы, не имеющей известных критических проблем. Предназначена для ознакомления заинтересованных лиц со способом реализации в Системе новых возможностей и проведения тестирования. Не обязательно содержит реализацию всех новых возможностей, планируемых в релизе
Бэклог продукта	(Product backlog) Список всех задач, которые необходимо выполнить для разработки релиза Системы
Бэклог спринта	(Sprint backlog) Список всех требований, которые необходимо выполнить в ближайший спринт
Владелец продукта	(Product owner). Представитель подразделения, которое владеет разрабатываемой Системой
Ежедневные совещания	(Scrum meetings) Собrania, целью которых является определение статуса и прогресса работы SCRUM-команды над спринтом
Жизненный цикл	Развитие системы, продукта, услуги и пр. Начинается со стадии разработки концепции и заканчивается прекращением применения
Модель ЖЦ	Структура процессов и действий, связанных с жизненным циклом, организуемых в стадии, которые также служат в качестве общей ссылки для установления связей и взаимопонимания сторон
Объект оценки	Подлежащая сертификации (оценке) версия Системы
Система	Программное обеспечение Джет Сайбер Кэмп («Jet CyberCamp»)
ПО	Программное обеспечение
Политика безопасности Системы	Совокупность правил, регулирующих управление, защиту и распределение информационных ресурсов, контролируемых Системой
Пользовательская история	(User story) Элемент списка требований, предъявляемых к разрабатываемому релизу Системы

Термин	Определение
Продукт	Объект разработки
Промежуточный релиз	Версия Системы, которая представляет собой результаты работы пяти спринтов
Разработчик	Лицо, осуществляющее разработку программного обеспечения
Релиз	Финальная версия Системы, предназначенная для сертификации и передачи к использованию
Спринт	(Sprint) Отрезок времени, который берется для выполнения определенного (ограниченного) списка задач
Стадия	Период в пределах жизненного цикла некоторого объекта, который относится к состоянию его описания или реализации

# 1 Описание процесса управления жизненным циклом

При реализации проекта по созданию и сопровождению Системы используются гибкая методология Agile и SCRUM как способ ее применения.

Достижение целей, стоящих перед проектной командой, обеспечивается за счет следующих факторов:

1. Применение комплекса мер, «зашитых» в SCRUM по умолчанию.

Основные из них:

1.1. Активное вовлечение членов проектной команды в разработку продукта. Позволяет избежать проблем, когда продукт формально соответствует исходным требованиям, но пользоваться им неудобно, сложно и т. п.

1.2. Создание прозрачной среды реализации продукта, позволяющей:

1.2.1. вносить изменения и уточнения в требования;

1.2.2. активно участвовать в поиске и выборе оптимальных методов решения поставленных задач;

1.2.3. иметь высокую степень контроля проекта/продукта.

1.3. Тщательный анализ поступающей информации не только с точки зрения «что нужно сделать», но и с точки зрения «какую ценность это принесет».

1.4. Нацеленность команды на результат.

2. Использование возможностей, с помощью которых можно упростить взаимодействие членов проектной команды и реализацию проекта:

2.1. Система учета задач, позволяющая:

2.1.1. гибко настраивать производственные процессы;

2.1.2. использовать автоматически обновляемую виртуальную доску задач, доступ к которой будут иметь все члены проектной команды;

2.1.3. упростить процедуры подготовки поставок и выпуска релизов за счет интеграции с репозиториями кода и инструментами непрерывной интеграции и непрерывного развертывания.

2.2. Wiki-система, позволяющая:

2.2.1. хранить все текстовые и визуальные артефакты в одном месте;

2.2.2. обеспечивать одновременную работу пользователей над артефактами с минимизацией конфликтов слияний.

2.3. Использование шаблона описания пользовательских историй и запросов на изменение вида.

2.4. Постоянный мониторинг требований на предмет соответствия критериям качества:

<b>Критерий качества</b>	<b>Описание</b>
Единичность	Требование описывает одну и только одну реализуемую сущность
Завершенность	Требование полностью определено в одном месте, вся необходимая информация присутствует
Последовательность/ Непротиворечивость	Требование не противоречит другим требованиям и полностью соответствует внешней документации
Атомарность	Требование «атомарно». Не может быть разбито на ряд более детальных требований без потери завершенности
Отслеживаемость	Требование полностью или частично соответствует деловым нуждам, как заявлено заинтересованными лицами, и задокументировано
Актуальность	Требование не стало устаревшим с течением времени
Выполнимость/ достижимость	Требование может быть реализовано в пределах проектных работ
Недвусмысленность	Требование определено без использования технического жаргона, акронимов и других скрытых формулировок. Оно выражает объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация. Определение не содержит нечетких фраз. Использование отрицательных утверждений и составных утверждений запрещено
Обязательность	Требование представляет определенную заинтересованным лицом характеристику, отсутствие которой приведет к неполноценности решения, которая не может быть проигнорирована. Необязательное требование – противоречие самому понятию требования
Измеримость	Требование может быть измерено количественными либо качественными показателями

2.5. Максимально глубокое погружение всех специалистов в предметную область и проблематику. Это позволяет обеспечить многоступенчатый контроль не

только с точки зрения функционального качества, но и с точки зрения бизнес-потребностей.

2.6. Привлечение к реализации проектов специалистов с опытом и знанием предметной области.

2.7. Укрупнено проект по созданию Системы можно разделить на три фазы:

2.7.1. Инициация;

2.7.2. Исполнение;

2.7.3. Финал.

## **1.1 Инициация проекта**

Проект запускается после разработки следующих артефактов:

- Концепция системы (функциональная и программно-аппаратная архитектура);

- План проекта;

- Оценка трудозатрат;

- Оценка стоимости.

## **1.2 Исполнение проекта**

Процесс реализации разделяется на следующие этапы:

- Анализ;

- Проектирование;

- Разработка;

- Документирование;

- Тестирование.

На этапе анализа формируется детальный план проекта, проводится бизнес анализ, разрабатываются техническое задание и матрица соответствия. Этап подразделяется на работы:



- Обследование предметной области, детализация и конкретизация бизнес-требований;

- Разработка технического задания;

- Описание функциональности системы и бизнес-процессов «как будет».

Здесь же определяется логическая структура объектов и их атрибутивный состав.

- По итогам этапа разрабатываются документы:

- Отчет об обследовании (необязательный документ)

- Техническое задание (обязательный документ).

Review Технического задания и других результатов этапа происходит на двух этапах: анализа и разработки технического проекта.

Этап технического проектирования подразделяется на работы:

- Проектирование (детализация) архитектуры разрабатываемого решения;

- Проектирование физических структур объектов хранения;

- Детальное определение интерфейсов взаимодействия.

Одновременно определяется стратегия тестирования. Результат проектирования – документы и модели, которые дорабатываются и корректируются в процессе реализации и оформляются на этапе документирования.

На этапе разработки программного обеспечения формируется детальный план разработки, включая разбиение функций по релизам, и разработка программного обеспечения.

Тестирование подразделяется на следующие работы:

- Определение стратегии тестирования;

- Разработка программы и методики испытаний и сценариев тестирования;

- Проведение тестирования;

- Исправление дефектов по результатам тестирования.

Документирование подразумевает разработку пользовательской и эксплуатационной документации, сопровождающей поставку решения, а также оформление проектной документации.

Документацию можно разделить на релизную, эксплуатационную и сертификационную.

Схема бизнес-процессов разработки Системы приведена в приложении А.

### **1.3 Финальная стадия**

Финальная фаза включает сдачу-приемку релиза Системы, ее регистрацию и гарантийное обслуживание.

#### **1.3.1 Сдача-приемка готового решения**

Релиз Системы принимает комиссия компании-разработчика. При испытаниях проверяется реализованная функциональность Системы, а сами испытания проводятся в соответствии с положениями и проверками Системы и методик испытаний.

#### **1.3.2 Гарантийные обязательства**

Гарантийный срок сопровождения составляет 12 месяцев начиная с даты покупки лицензии, а техническая поддержка Системы предоставляется в течение всего этого срока. Сопровождение системы необходимо для отсутствия простоя в предоставлении услуг Заказчикам по причине невозможности функционирования Системы (аварийная ситуация, ошибки в работе системы) и обеспечения гарантий корректного функционирования системы и дальнейшего развития её функциональности.

Обнаруженные в гарантийный период недостатки устраняются бесплатно. Также гарантийная поддержка распространяется на консультационные услуги компании разработчика. К процессу привлекаются менеджеры и сотрудники отделов Технического департамента компании-разработчика: технической поддержки, разработки и тестирования.

Сервисная поддержка осуществляется по следующим каналам связи:

- Телефон: +7 (495) 411-76-01

- Электронная почта: [CYBERCAMP@jet-soft.ru](mailto:CYBERCAMP@jet-soft.ru)

В зависимости от уровня купленной поддержки, сервис может оказываться в рабочие часы, с 10:00 до 18:00 по Москве или круглосуточно, в соответствии регламентом ТП

Постгарантийное обслуживание регламентируется отдельным договором. Такой договор предусматривает заключение двухстороннего соглашения об уровне сервиса.

#### **1.4 Описание модели жизненного цикла Системы**

При разработке и сопровождении Системы применяется модель жизненного цикла (ЖЦ) Scrum.

Отличительной особенностью Scrum-модели является итеративность. Процесс разработки разбивается на отдельные отрезки времени – спринты, в течение которых выполняется работа по созданию Системы.

Спринт всегда ограничен по времени и имеет одинаковую продолжительность на протяжении всей жизни Системы.

В процессе реализации Системы применяются двухнедельные спринты. По окончании спринта создается новая рабочая версия Системы (бета-версия). Пять спринтов составляют этап проекта, по окончании которого формируется промежуточный релиз.

Модель ЖЦ представлена на рисунке 1.

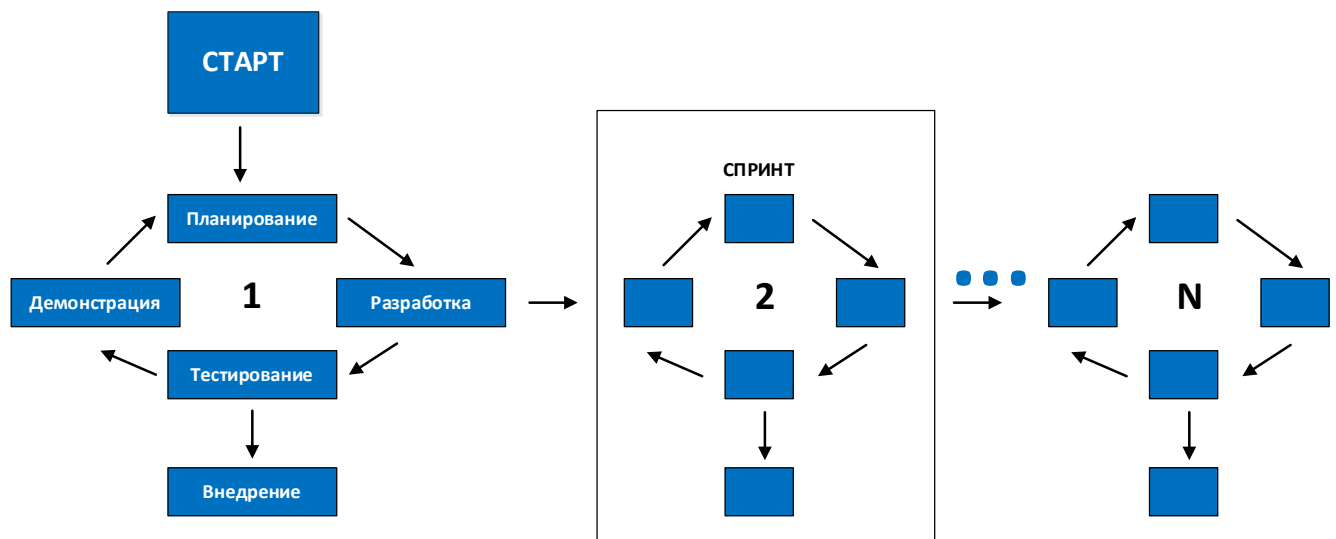


Рисунок 1 – Модель ЖЦ

### 1.4.1 Зона ответственности по ролям Scrum-модели

Основные роли Scrum-модели:

1. Владелец продукта:

1.1. Единственный ответственный за продукт и список задач продукта сотрудник.

Эксперт в предметной области, понимающий, каким должен стать продукт, какие задачи он должен решать, как выглядеть и как работать (с точки зрения бизнеса).

1.2. Единая точка принятия окончательных решений для команды в проекте. Это всегда один человек. Такой подход позволяет избежать проблемы противоречивости и неоднородности требований.

1.3. Четко понимает, какие задачи должны быть выполнены, какую ценность они несут и каковы их приоритеты.

1.4. Принимает результаты в конце каждой итерации/спринта.

2. Scrum-мастер следит за соблюдением принципов Scrum-разработки, координирует процесс, проводит ежедневные собрания (Scrum meetings). Задача SCRUM-мастера – помочь команде максимизировать ее эффективность за счет устранения препятствий, помощи, обучении и мотивации команде, помощи Владельцу продукта:

2.1. Член команды – Agile-лидер команды, который помогает правильно использовать SCRUM-технология.

2.2. Следит за тем, чтобы команда выполняла принятые решения, соблюдала agile- и иные установленные практики.

2.3. Проводит командные встречи/митинги (планирование, демонстрацию, ретроспективу).

2.4. Устраняет проблемы, мешающие команде эффективно работать (самостоятельно или с привлечением управленческого персонала).

3. Scrum-команда – занимается разработкой программного решения. В Scrum-команду входят программисты, специалисты тестирования, аналитики и прочие специалисты.

### **1.4.2 Создание беклога продукта**

Бэклог продукта представляет собой упорядоченный по степени важности список требований, которые предъявляются к разрабатываемой Системе. Элементы этого списка называются пользовательскими историями и определяют задачи спринта. Каждой истории соответствует уникальный идентификатор.

Описание каждой истории включает в себя набор атрибутов, необходимых для дальнейших проектных работ:

- Общая информация – обязательный атрибут. Информация атрибута отображает основную идею истории.
- Приоритет – приоритет выполнения истории.
- Этап реализации – обязательный атрибут. Определяет этап, в котором эта история реализуется.
- Ответственный – ответственный за выполнение истории.
- Область истории – обязательный атрибут. Определяет область, в рамках которой выполняется история.
- Условие выполнения – обязательный атрибут. Указывает на условия, определяющие выполнение пользовательской истории.
- Описание – описание истории с точки зрения пользователя (детальная информация с четким описанием функциональности).

### **1.4.3 Планирование спринта и создание бэклога спринта**

Во время планирования спринта проектная команда отбирает для реализации высокоприоритетные пользовательские истории из бэклога продукта. Приоритет задачи определяет владелец продукта, а трудозатраты на реализацию оценивает SCRUM команда.

Истории, выбранные для реализации в течение действующего спринта, составляют бэклог спринта. Объем бэклога выбирается так, чтобы каждая история была успешно реализована к концу спринта.

### **1.4.4 Работа над спринтом. Ежедневные совещания**

В рамках модели Scrum проводятся ежедневные совещания. Их задачи: определение статуса и прогресса работы над спринтом; обнаружение возникших препятствий; выработка решений по изменению стратегии, которые необходимы для достижения целей спринта.

### **1.4.5 Демонстрация**

По завершении каждого спринта демонстрируется работа бета-версии Системы.

Результатом демонстрации служит пересмотренный бэклог продукта, некоторые элементы которого могут быть перенесены на следующий спринт. Бэклог продукта может быть полностью пересмотрен из-за вновь открывшихся обстоятельств.

### **1.4.6 Ретроспектива**

На этапе подведения итогов выполненной работы проводится ретроспектива – совещание SCRUM-команды, в процессе которого анализируются проблемы, возникшие при реализации задач, и принимаются решения для предотвращения их появления в дальнейших спринтах.

#### **1.4.7 Поставка**

Поставка готового релиза Системы возможна после подписания лицензионного соглашения и покупки лицензии. После этого скачивание программного обеспечения становится доступным.

#### **1.4.8 Сопровождение**

Сопровождение обеспечивает эффективность эксплуатации Системы на протяжении всего жизненного цикла. Это процессы, которые объединяют действия по обеспечению работоспособности программного продукта, в том числе:

- создание новых версий; тестирование и настройки;
- мониторинг работы;
- выявление ошибок и их устранение;
- взаимодействие с пользователями в рамках гарантийной поддержки и соглашений о предоставляемых услугах и консультациях.

Схема процессов приведена в Приложении Б.

#### **1.4.8 Устранение неисправностей**

В процессе разработки и эксплуатации приложения могут быть выявлены дефекты, которые необходимо исправить. Для этого дефекты должны быть внесены в беклог с подробным описанием и обозначением приоритета исправления. В последствии исправления доставляются через процесс поставки ПО.

## **2 Обоснование выбора модели жизненного цикла**

Применение методологии гибкой разработки Scrum позволяет оперативно реагировать на изменение требований в процессе реализации программного продукта, создавать и тестировать промежуточные релизы и бета-версии. Как результат – уменьшается время на создания продукта и снижается его себестоимость.

SCRUM-модель полностью согласуется с каскадной моделью ЖЦ, где все стадии создания выполняются последовательно.



### **3 Информация и персонале**

Для полноценного обеспечения развития продукта, а также для качественной поддержки, в компании присутствует необходимая экспертиза и персонал.

В рамках Технического департамента, в компании ООО «Джет Софт» сопровождением и развитием продукта занимаются специалисты:

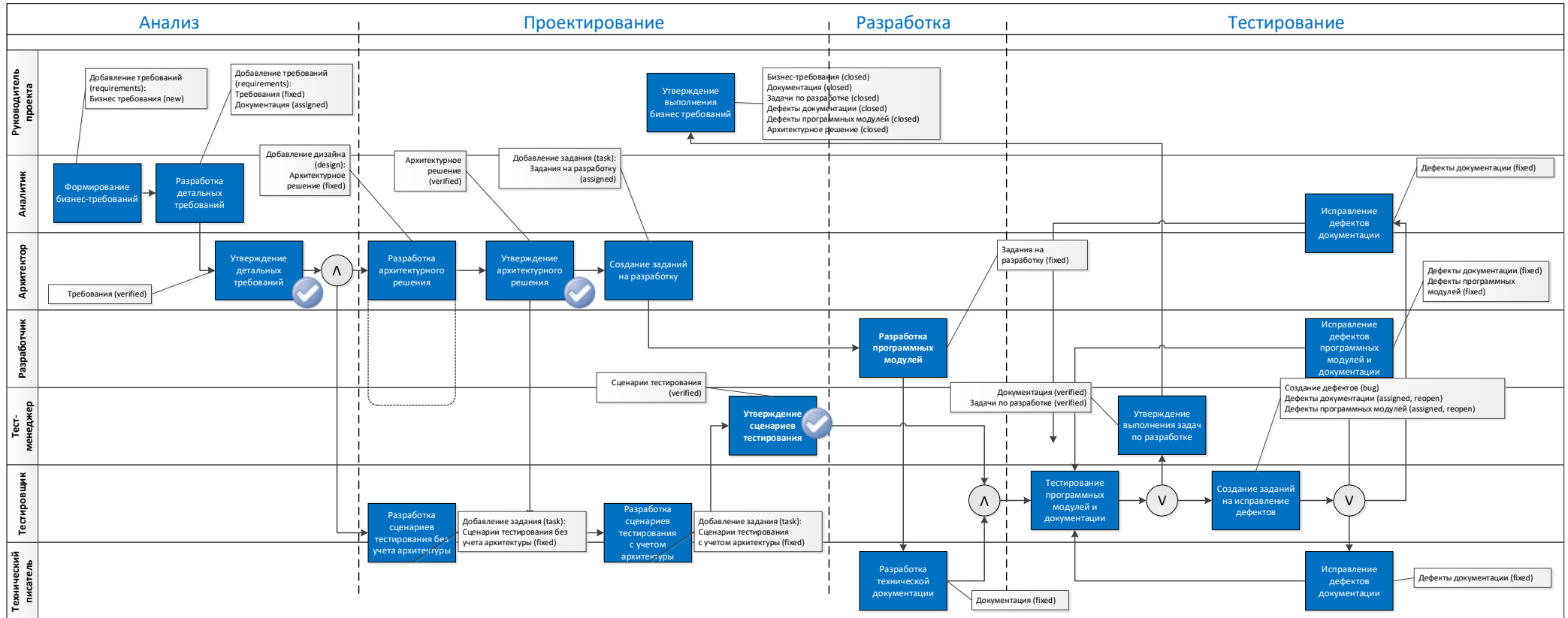
- Разработчики ПО (frontend, backend);
- Тест-инженеры (QA-инженеры);
- Системные аналитики;
- Системные администраторы (DevOps);

В рамках Департамента развития продуктов, в компании ООО «Джет Софт» сопровождением и развитием продукта занимаются специалисты:

- Менеджер продукта;
- Администратор проекта;
- Руководитель проекта;
- Продуктовый дизайнер;
- Инженеры внедрения.

Таким образом, специалисты ООО «Джет Софт» обладают необходимым набором компетенций и ресурсов для работы со всеми компонентами, входящими в состав ПО, при решении прикладных задач, соответствующих функционалу программы.

# Приложение А: Процессы разработки ПО



# Приложение Б: Процессы сопровождения

