

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ
ПРОИЗВОДСТВЕННОГО ПРОЦЕССА РАЗРАБОТКИ
ГОСУДАРСТВЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С УЧЕТОМ
ПРИМЕНЕНИЯ ИТЕРАЦИОННОГО ПОДХОДА К РАЗРАБОТКЕ**

(ПРОЕКТ)

ОГЛАВЛЕНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ2
2. ПОРЯДОК СОЗДАНИЯ И РАЗВИТИЯ СИСТЕМЫ4
 - 2.1 Модель требований к Системе4
 - 2.2 Этапы создания и развития Системы5
 - 2.3 Процесс реализации Групп требований к Системе8
 - 2.4 Организация команд разработки10
 - 2.5 Особенности реализации первой очереди Системы16
 - 2.5.1 Разработка технического проекта на Систему в особых случаях16
 - 2.5.2 Разработка минимально достаточной версии Системы17
3. СТАДИЯ ПЛАНИРОВАНИЯ18
4. СТАДИЯ ВЫПОЛНЕНИЯ23
 - 4.1 Базовый уровень24
 - 4.1.1 Общие рекомендации по организации работ на базовом уровне24
 - 4.1.2 Ролевая структура Рабочей группы Системы25
 - 4.1.3 Выполнение работ по созданию и развитию Системы на базовом уровне на основе итерационного подхода27
 - 4.2 Расширенный уровень45
 - 4.2.1 Общие рекомендации по организации работ на расширенном уровне45
 - 4.2.2 Ролевая структура45
 - 4.2.3 Особенности выполнения работ на расширенном уровне47
5. СТАДИЯ ПРОВЕРКИ ВЫПОЛНЕННЫХ РАБОТ52
6. СТАДИЯ КОРРЕКТИРОВКИ54

1. ОБЩИЕ ПОЛОЖЕНИЯ

1. Настоящий документ содержит методические рекомендации по созданию и развитию государственных информационных систем (далее – Методические рекомендации, Система) с использованием современных методов и практик управления разработкой цифровых продуктов и сервисов.

2. Документ соответствует положениям законодательства Российской Федерации, регулирующим создание и развитие Систем.

3. Положения данного документа распространяются на деятельность федеральных органов исполнительной власти, органов исполнительной власти субъектов Российской Федерации, органов управления государственными внебюджетными фондами Российской Федерации (далее - Ведомства), а также их подведомственных организаций, частных партнеров в соответствии с соглашениями о государственно-частном партнерстве, концессионеров в соответствии с концессионными соглашениями, коммерческих организаций выполняющие (или планирующие выполнять) работы по созданию и развитию Систем в рамках Государственных контрактов и организаций иных организационно-правовых форм.

4. Цели документа:

1) сформировать для всех Ведомств и организаций, указанных в п.3 настоящего документа, единое понимание методов и практик создания и развития Систем;

2) разработать предложения по использованию Методических рекомендаций в зависимости от масштаба создаваемых Систем (по количеству компонентов Систем, связей между ними и охвату пользовательской аудитории) и сложности организации работ по созданию и развитию Систем (количества привлекаемых специалистов).

5. Использование настоящих Методических рекомендаций направлено на обеспечение:

1) достижения целей предоставления государственных услуг, государственных функций, включая контрольно-надзорную деятельность в условиях изменяющихся потребностей клиентов и условий (контекста) использования Систем;

2) сокращения сроков и повышение качества предоставления клиентам государственных сервисов с использованием создаваемых Систем;

3) создания и развития Систем в рамках установленных бюджетов и с минимальными рисками недостижения целей создания Системы.

6. В ходе выполнения работ по созданию (развитию) Систем рекомендуется использовать взаимосвязанные методы и практики итерационной разработки, бережливого производства и DevSecOps¹, указанные в Приложении № 1 к настоящему документу.

7. В ходе создания (развития) Систем требуется выполнять мероприятия, направленные на предотвращение появления и/или устранение уязвимостей разрабатываемого программного обеспечения, устанавливаемые законодательством Российской Федерации и действующими нормативными актами в области защиты обрабатываемой информации и создания Систем в защищенном исполнении, в том числе указанные в документе «Методические рекомендации по обеспечению разработки безопасного программного обеспечения на ЕЦП «ГосТех».

8. В настоящем документе использованы термины, их определения и сокращения, приведенные в Приложении № 2, и (или) определенные действующим законодательством.

¹ DevSecOps – акроним из слов Development, Security и Operations - набор методов практик, основанных на автоматизации рабочих процессов создания и эксплуатации Системы, предполагающих непрерывное улучшение Системы путём непрерывной интеграции реализуемых компонентов, проверок безопасности и непрерывной доставки реализованной функциональности до клиента

2. ПОРЯДОК СОЗДАНИЯ И РАЗВИТИЯ СИСТЕМЫ

9. Порядок реализации мероприятий по созданию, развитию, вводу в эксплуатацию, эксплуатации и выводу из эксплуатации Систем определен Постановлением Правительства Российской Федерации от 6 июля 2015 г. № 676 «О требованиях к порядку создания, развития, ввода в эксплуатацию, эксплуатации и вывода из эксплуатации государственных информационных систем, и дальнейшего хранения содержащейся в их базах данных информации» (далее – Постановление № 676).

10. В настоящем документе представлены особенности производственного процесса итеративной разработки Систем, детализирующие ключевые положения Постановления № 676.

2.1 Модель требований к Системе

11. Система разрабатывается в соответствии с требованиями, определяемыми техническим заданием на создание Системы, порядок формирования, согласования и утверждения которого определяется Постановлением № 676. При этом в техническом задании на создание и развитие Системы (или ее части) может быть предусмотрено требование о применении итерационного подхода к разработке.

12. В случае применения итерационного подхода к разработке полный объем требований к Системе, устанавливаемых в техническом задании, должен быть сформирован в виде Групп требований, реализация каждой из которых должна обеспечивать отдельную самостоятельную ценность для клиентов. При этом для Групп требований должны быть установлены конечные сроки их реализации и критерии приёмки.

13. Каждая Группа требований в общем случае включает (Рисунок 1):

1) функциональные требования, реализация которых создает ценность для клиентов;

2) нефункциональные требования – требования к видам обеспечения, в том числе требования о защите информации, реализация которых обеспечивает создание функциональных требований.

14. Приоритеты реализации Групп требований, указанных в техническом задании, могут определяться (уточняться) в порядке, устанавливаемым Ведомством, ответственным за создание и развитие Системы, например, с использованием Методики определения приоритетов, представленной в Приложении №1. Управление приоритетами обеспечивает выполнение в первую

очередь наиболее важных работ, дающих максимальный эффект, и снижение рисков реализации запланированной функциональности.

15. Документом, определяющим приоритеты создания или развития Системы, является План-график итерационной разработки. План-график представляет собой упорядоченный по приоритету список Группы требований, которые планируется выполнить в рамках создания или развития Системы с указанием планового срока реализации каждой Группы требований.

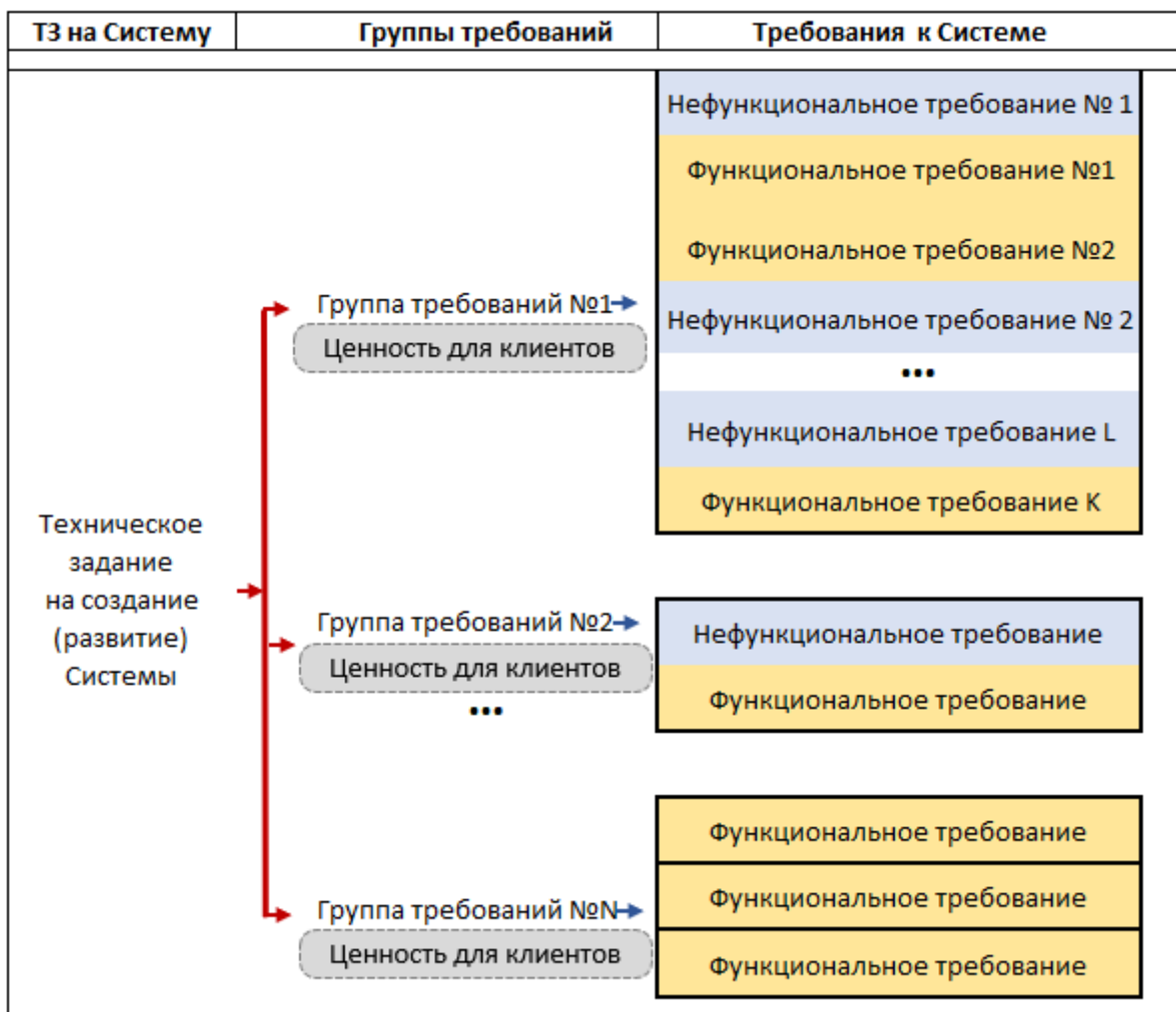


Рисунок 1. Структура требований к Системе

2.2 Этапы создания и развития Системы

16. Постановление № 676 определяет следующие этапы выполнения работ по созданию или развитию Системы:

- 1) разработка документации на Систему и ее части;
- 2) разработка рабочей документации на Систему и ее части;
- 3) разработка или адаптация программного обеспечения;

- 4) пусконаладочные работы;
- 5) проведение предварительных испытаний Системы;
- 6) проведение опытной эксплуатации Системы;
- 7) проведение приемочных испытаний Системы.

17. Методические рекомендации предлагают использовать итерационный подход к организации разработки Системы, основанный на цикле Деминга-Шухарта (Рисунок 2) и включающий следующие повторяющиеся **стадии**:

1) **Планирование**: определение целей, планирование работ по их достижению, а также выделение и распределение ресурсов, необходимых для выполнения запланированных работ;

2) **Выполнение**: непосредственное выполнение запланированных работ;

3) **Оценка результатов** сбор информации результатов выполнения запланированных работ, например, ключевых показателей эффективности (КПЭ), оценка результатов, выявление и анализ отклонений от плановых значений показателей, установление причин отклонений;

4) **Корректировка**: принятие мер по устранению причин отклонений от запланированного результата, изменение целей (если все варианты по корректировке отклонений исчерпаны), изменение планов и распределения ресурсов.

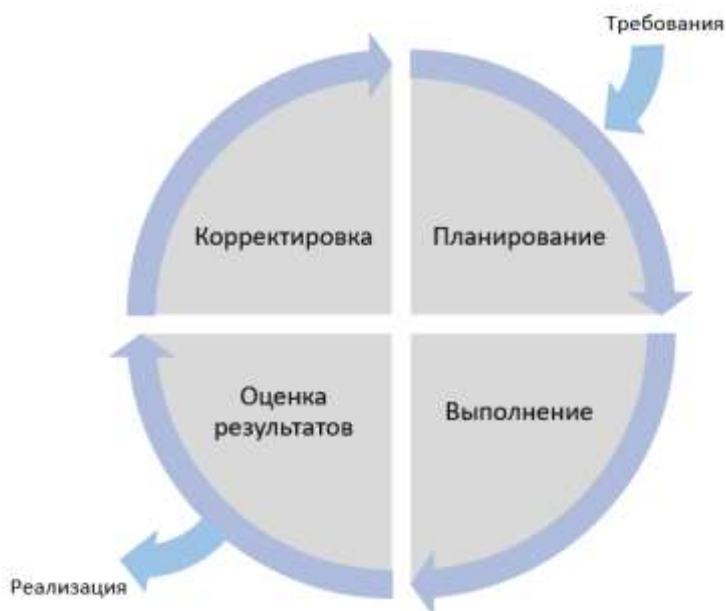


Рисунок 2. Типовой цикл выполнения работ с использованием итерационного подхода

18. В зависимости от сложности создаваемой Системы предлагается использовать несколько уровней разработки, на каждом из которых работа организуется итерационно. В общем случае предлагается использовать трехуровневую модель процесса создания и развития Системы (Рисунок 3):

1) Первый уровень состоит из последовательно реализуемых **Очередей** создания и развития Системы, на каждом из которых осуществляется реализация Технического задания на Систему (или ее части) в целом.

2) Второй уровень состоит из последовательно реализуемых **Инкрементов** (фиксированной продолжительностью от 8 до 12 недель), в рамках которых осуществляется реализацией Групп требований.

3) Третий уровень состоит из последовательности **Итераций** (фиксированной продолжительностью от 1 до 4 недель), на которых осуществляется реализация функциональности Системы, определяемой отдельными требованиями Группы требований.

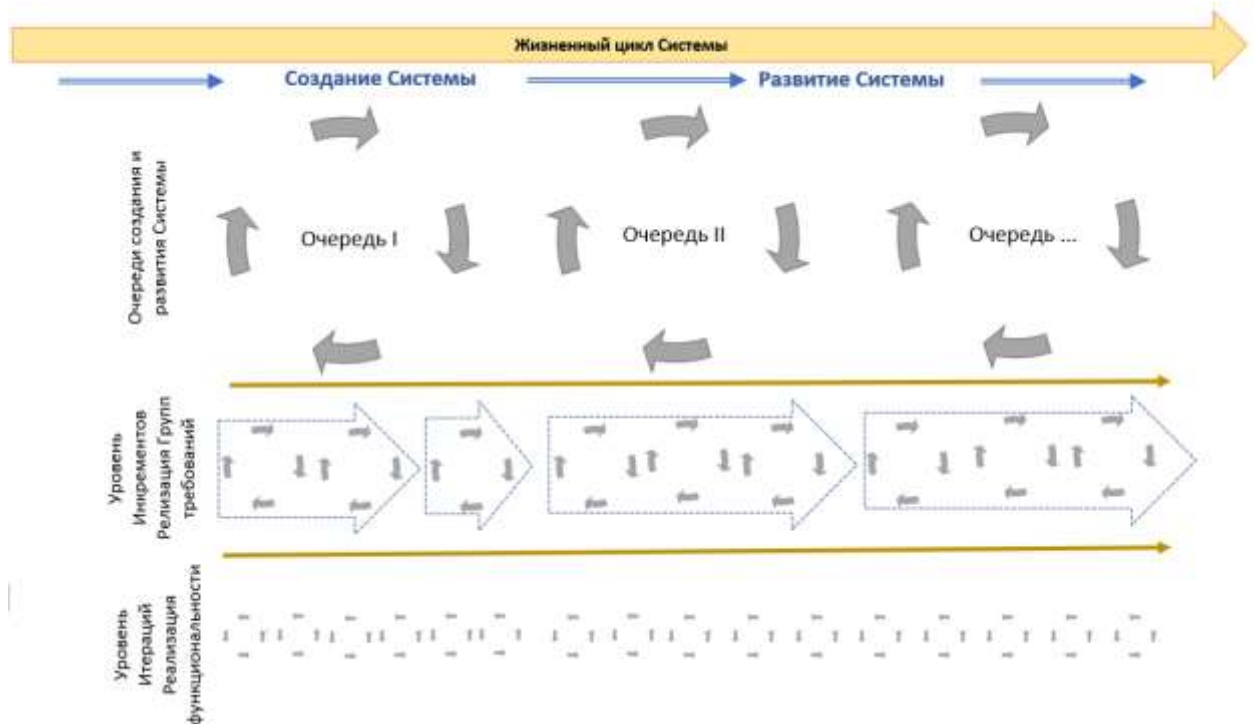


Рисунок 3. Многоуровневая модель процесса создания и развития Системы

19. В рамках каждой Очереди могут быть созданы Система, часть Системы, развитие Системы, иные мероприятия, указанные в разделе 2.5 настоящего документа.

20. Сроки реализации каждой Очереди Системы рекомендуется привязывать к бюджетному процессу в Российской Федерации.

21. Содержание мероприятий по реализации очереди Системы, определяются этапами выполнения работ, указанными в Постановлении № 676, а также иными дополнительными мероприятиями для обеспечения итерационного подхода к разработке (Таблица 1).

Таблица 1. Стадии создания Системы

№	Стадия создания (развития) Системы	Мероприятия по созданию (развитию) Системы
1.	Планирование	1.1. Создание Рабочей группы 1.2. Разработка (уточнение) Концепции, финансово-экономического обоснования, правовых актов, являющихся основанием для создания или развития Системы 1.3. Разработка технического задания, технико-экономического обоснования 1.4. Определение Исполнителя 1.5. Разработка план-графика итерационной разработки 1.6. Разработка документации на Систему и ее части
2.	Выполнение работ	2.1. Разработка и адаптация программного обеспечения 2.2. Разработка Рабочей документации на Систему 2.3. Пусконаладочные работы 2.4. Демонстрация Системы 2.5. Предварительные испытания
3.	Проверка выполненных работ	3.1. Опытная эксплуатация 3.2. Приемочные испытания 3.3. Ввод в эксплуатацию 3.4. Мониторинг функционирования
4.	Корректировка	4.1. Определение требований к развитию Системы 4.2. Оценка и согласование бюджета развития Принятие решения о развитии 4.3. Переход к стадии «Планирование» следующей очереди Системы

22. Организация работ с использованием итерационного подхода на уровне Итераций и Инкрементов описана в разделах 3 – 6 настоящего документа.

2.3 Процесс реализации Групп требований к Системе

23. Итерационный подход к разработке предполагает выполнение последовательности Инкрементов, каждый из которых включает в себя реализацию Групп требований к Системе в плановые сроки, в соответствии с

предопределенными целями, критериями приемки каждого требования и критериями готовности к вводу в эксплуатацию. Это обеспечивает в рамках каждого Инкремента предсказуемый непрерывный процесс разработки новой ценности для клиентов (повышение качества результатов оказания Государственных услуг и выполнения государственных функций с использованием создаваемой и развиваемой Системы).

24. В рамках каждого Инкремента реализуются одна или несколько Групп требований к Системе (Рисунок 4), включенных в План-график итерационной разработки. В свою очередь Инкременты реализуются более короткими циклами разработки – Итерациями, в рамках которых реализуются отдельные функциональные и нефункциональные требования.

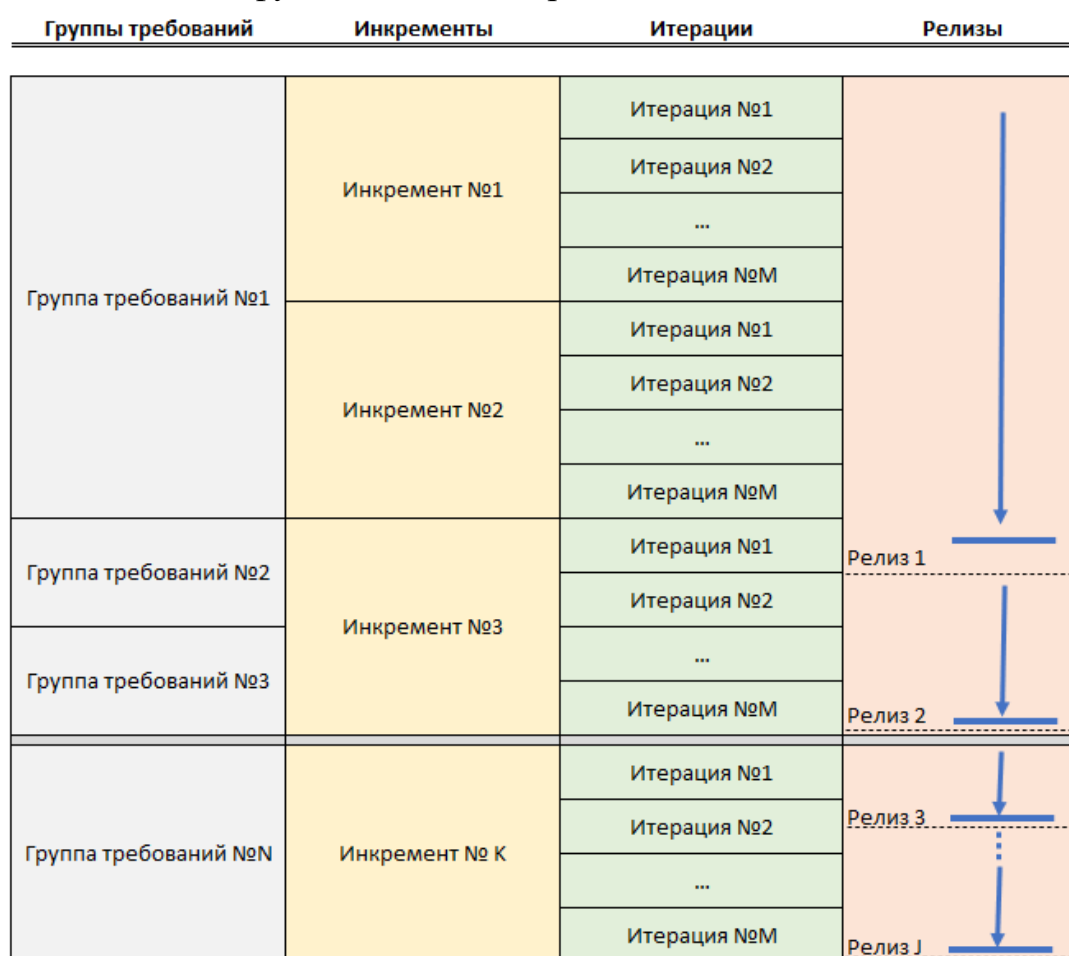


Рисунок 4. Реализация групп требований в рамках итераций

25. В рамках каждого Инкремента регулярно оцениваются результаты выполненных работ. Эта регулярная оценка обеспечивает управление рисками создания и развития Системы, необходимое для обеспечения того, чтобы бюджетные затраты на создание и развитие Системы приносили соразмерную отдачу в виде предоставляемой клиентам ценности. Регулярные оценки выполняемых работ проводятся в интересах обеспечения целевого эффекта

оказания государственных услуг и выполнение государственных функций с использованием Системы.

26. После реализации Группы требований и после выполнения мероприятий по проверке выполненных работ, указанных в разделе 5, может быть подготовлен и по решению уполномоченных должностных лиц Ведомства, введен в эксплуатацию релиз Системы, что предполагает предоставление клиентам части реализованной функциональности. Такой подход обеспечивает:

1) оперативное предоставление ключевых, наиболее важных, возможностей Системы клиентам в рамках первого релиза;

2) сокращение сроков подготовки последующих релизов к вводу в эксплуатацию при условии, что не потребуются повторения мероприятий по подготовке и вводу Системы в эксплуатацию, выполненных при вводе в эксплуатацию первого релиза. Например, если не осуществляется модернизация Системы, приводящая к повышению класса ее защищенности и (или) к изменению архитектуры системы защиты информации в части изменения видов и типов программных, программно-технических средств и средств защиты информации, изменения структуры системы защиты информации, состава и мест расположения Системы и ее компонентов – действия ранее выданного аттестата соответствия после модернизации (установки обновлений) не прекращается и эксплуатация Системы может быть продолжена, а в дальнейшем - проведена повторная ее аттестация (ФСТЭК, Приказ от 29.04.2021 № 77 «Об утверждении порядка организации и проведения работ по аттестации объектов информатизации на соответствие требованиям о защите информации ограниченного доступа, не составляющей государственную тайну», п. 33).

2.4 Организация команд разработки

27. Основной организационной единицей, на которую возлагается выполнение работ на уровне Итераций, являются **Команды разработки**. Каждая команда разработки представляет собой кросс-функциональную группу специалистов, содержащую необходимые компетенции для решения возложенных на них задач. Так как сложность взаимодействия в таких группах увеличивается по мере увеличения числа специалистов в группе, рекомендуется формирование небольших команд (например, обычно лучше иметь две команды по пять человек, чем одну команду из десяти). Рекомендуется в Команду разработки включать 5...11 человек, которые проектируют, реализуют, тестируют и обеспечивают предоставление пользователям заданной функциональности.

28. При формировании Команды разработки необходимо учитывать специфику решаемых задач. При этом, если оценка времени использования специалиста с требуемой компетенцией превышает 80%, то такого специалиста необходимо включить в состав команды.

29. Поскольку сложность и объем задач по созданию Системы может потребовать более 11 человек, необходимо использование нескольких Команд разработки. Кроме этого, создание и развитие Системы требует широких и разнообразных компетенций и навыков, поэтому рекомендуется использование различных типов Команд разработки. Условно может быть использована следующая типизация:

- команда ценности – организована вокруг потока создания ценности и имеет возможность доставлять ценность непосредственно клиенту (конечному пользователю);
- команда сложных подсистем – организована вокруг конкретных подсистем, требующих глубоких специальных навыков и опыта;
- команда платформы – организована вокруг разработки и поддержки платформ, предоставляющих услуги другим командам;
- команда поддержки – организована, чтобы помочь другим командам со специализированными возможностями и новыми технологиями.

30. **Команды ценности** отвечают за создание и доставку требуемых функциональных возможностей Системы для клиентов (конечных пользователей) как можно быстрее, безопаснее и, не требуя передачи другим командам для выполнения возложенных на них задач. Командам, ориентированным на создание ценности потребуется набор компетенций для реализации требований к Системе. Эти компетенции включают (но не ограничиваются):

- создание проектных и архитектурных решений;
- разработка (адаптация) программного обеспечения Системы, включая серверные компоненты и пользовательские интерфейсы;
- мониторинг и обеспечение устойчивой работоспособности;
- тестирование и обеспечение качества;
- обеспечение безопасности приложений.

31. **Команда сложных подсистем** отвечает за создание и поддержку частей Системы, которые в значительной степени зависят от специальных знаний. Команда сложных подсистем берёт на себя ответственность за создание и обслуживание тех частей системы, которые требуют глубоких и постоянных

специализированных технических знаний, включая узкоспециализированные компоненты Системы, критически важные для безопасности элементы Системы, которые имеют высокую стоимость отказа, специальные алгоритмы.

Примеры сложных подсистем могут включать, например, математические модели, алгоритмы согласования сделок в реальном времени, подсистемы отчетности о транзакциях для финансовых услуг, алгоритмы распознавания лиц.

В общем случае, в зависимости от особенностей создаваемой Системы может быть сформировано несколько команд сложных подсистем.

32. Команды платформ создаются для поддержки цифровых платформ², используемых в ходе создания Системы, и предоставления консультационных услуг командам ценности. Предоставляемые услуги, должны быть хорошо документированы, просты в использовании, соответствовать цели, и предлагать возможности повторного использования.

Использование цифровых платформ призвано существенно сократить время реализации функциональных возможностей Системы, , в случае необходимости, обеспечивает быстрое прототипирование создаваемой функциональности.

33. Команды поддержки создаются для развёртывания и поддержки технологий разработки, а также оказания помощи командам в приобретении этих новых знаний и навыков в освоении новых технологий, обеспечивающих основные рабочие процессы.

Команды поддержки предоставляют экспертные знания и обеспечивают поддержку технологий:

- механизмов DevOps, включая механизмы непрерывной интеграция и сборки;
- автоматизированного тестирования;
- непрерывную интеграцию
- автоматическую сборку релизов;
- инженерные практики качества;
- средства безопасности при разработке;
- развёртывания и поддержки контура разработки, тестового контура и контура эксплуатации.

При этом команды поддержки не предназначены для устранения проблем с качеством реализации функциональности Системы, создаваемой командами

² Цифровая платформа — это набор готовых сервисов, к которым могут получить доступ разработчики, например, через набор прикладных программных интерфейсов (API).

ценности. Команды поддержки могут быть постоянными или временными, а также обеспечивать поддержку различных команд.

34. С учетом описанных типов команд и их особенностей для создания Системы могут быть организованы Группы команд разработки, отвечающие особенностям создаваемой Системы (Рисунок 5).



Рисунок 5. Обобщённая структура Группы команд разработки

35. Эффективные Группы команд обычно состоят из 50-150 человек. Верхний предел основан на числе Данбара, который предполагает ограничение на количество людей, с которыми можно сформировать эффективные, стабильные рабочие отношения. Нижний предел основан на эмпирических наблюдениях.

36. Учитывая указанные ограничения по размеру, существует два основных шаблона организации разработки: меньшие потоки создания ценности могут быть реализованы с помощью одной Группы команд, а большие потоки создания ценности должен поддерживаться несколькими Группами команд разработки (Рисунок 6). В последнем случае Группы команд разработки, чтобы обеспечить их координацию в ходе создания и развития крупномасштабных Систем (совокупная стоимость владения которых превышает 100 млн.руб.) предлагается объединять в Пулы команд разработки.

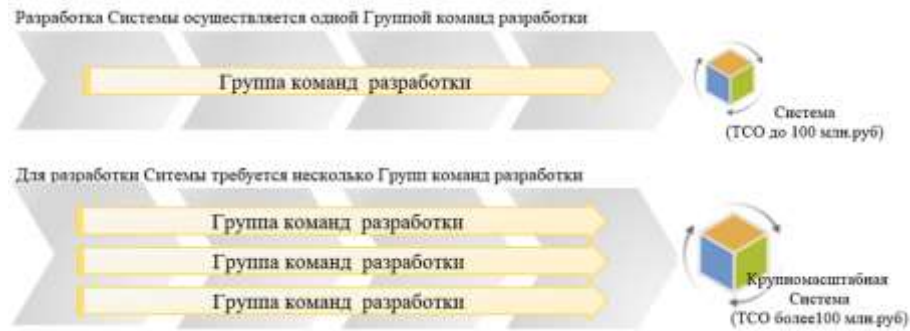


Рисунок 6. Варианты организации Групп команд разработки

37. Для организации Пулов команд может быть использован подход определения типов Групп команд разработки аналогично представленному выше для Команд разработки.

Группы команд разработки, ориентированный на создание ценности, как и команда ценности должна включать необходимых специалистов, навыки и компетенций которых достаточно для решения возложенных на них задач. Сферы ответственности этих групп разработки, как правило, такие же, как и для команд ценности.

Группы команд сложных подсистем могут быть использованы при построении крупномасштабных Систем для снижения нагрузки на Группы команд разработки ценности. Например, подсистема наведения для автономного транспортного средства вполне может потребовать привлечения Группы команд разработки сложной подсистемы.

Группы команд разработки, ориентированные на создание и поддержку платформы, предоставляют услуги Группам команд, ориентированным на создание ценности. Например, система связи, которая управляет данными, передаваемыми между различными подсистемами автономного транспортного средства, может быть представлена как платформа с четко определенными интерфейсами. Одним из дополнительных преимуществ Группы команд платформы является то, что она также поддерживает единые цифровые платформу для всех Групп команд разработки.

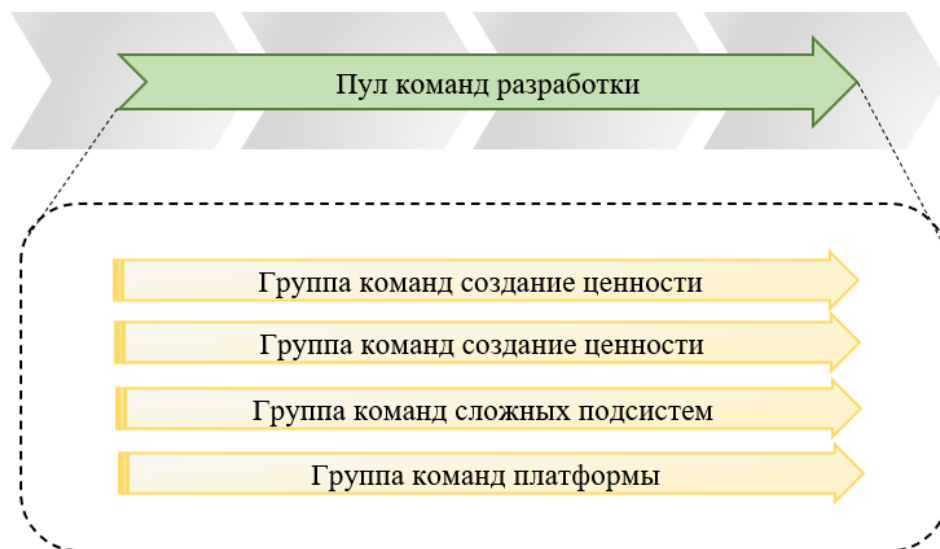


Рисунок 7. Обобщённая структура Пула команд разработки

38. Для управления работами по созданию и развитию Системы предлагается сочетание децентрализованного и централизованного подхода. Децентрализованный подход предполагает использование для выполнения работ кросс-функциональных самоорганизующихся команд, которые в рамках установленных полномочий могут принимать самостоятельные решения, в том числе - в части порядка (последовательности) выполнения работ по реализации функций Системы в рамках отдельных итераций при условии выполнения принятых обязательств по достижению целей, определяемых Планом итерационной разработки. Это сокращает временные и ресурсные издержки выполнения работ и позволяет использовать инновационные решения, предлагаемые разработчиками.

При этом сохраняется необходимость централизованного принятия критичных решений (архитектурных, финансовых, контрактных и иных) должностными лицами Ведомств, а также обеспечения координации команд.

39. Использование единых по продолжительности циклов выполнения работ позволяет координировать Команды разработки в рамках Группы команд (также как и Группы команд разработки в рамках Пула команд) и установить единые временные рамки и единые процессы для них. Для согласования выполняемых работ по созданию и развитию Системы в целом между Командами разработки должны проводиться регулярные мероприятия по координации работ, а также единые мероприятия по планированию предстоящих Инкрементов и подведение итогов выполнения работ (Рисунок 8).

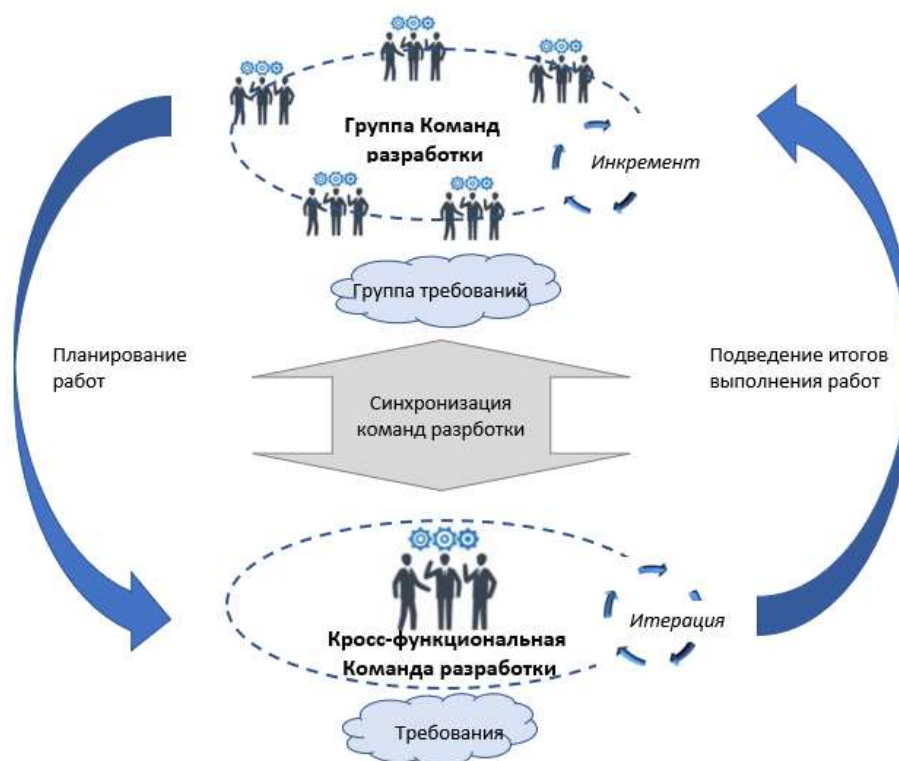


Рисунок 8. Координация Группы команд разработки

2.5 Особенности реализации первой очереди Системы

40. При планировании очередей создания и развития Системы целесообразно в качестве первой очереди рассмотреть:

1) Проектирование Системы в случае, если в соответствии с финансово-экономическим обоснованием совокупная стоимость владения системой составляет более 1 миллиарда рублей – согласно Постановлению № 676.

2) Создание минимально достаточной версии Системы для подтверждения проектных решений и оперативной демонстрации ключевых возможностей Системы (представление ценности клиентам).

2.5.1 Разработка технического проекта на Систему в особых случаях

41. В случае, если в соответствии с финансово-экономическим обоснованием, совокупная стоимость владения Системой составляет более 1 миллиарда рублей, в соответствии с Постановлением № 676 разрабатывается техническое задание на проектирование Системы, к которому прилагается технико-экономическое обоснование, включающее в себя оценку финансовых, трудовых и материальных ресурсов, необходимых для проектирования системы. Техническое задание на проектирование Системы с приложением технико-

экономического обоснования согласуется с уполномоченным федеральным органом исполнительной власти, осуществляющим функции по выработке и реализации государственной политики и нормативно-правовому регулированию в сфере информационных технологий.

42. В результате работ по проектированию Системы (или ее части) должны быть разработаны технический проект на систему (или ее часть), а также проект технико-экономического обоснования, включающий в себя оценку финансовых, трудовых и материальных ресурсов, необходимых для реализации разработанных проектных решений.

43. Технический проект на Систему (или ее часть) и технико-экономическое обоснование согласовываются и утверждаются в порядке, определяемым Постановлением № 676, и используются для выполнения работ на последующей очереди.

44. Если для выполнения работ по проектированию Системы привлекаются внешние исполнители, рекомендуется заключение отдельного Государственного контракта.

2.5.2 Разработка минимально достаточной версии Системы

45. С целью проверки правильности разработанных проектных решений и **в интересах снижения рисков, потерь времени и выхода за рамки бюджета** рекомендуется использовать подход «бережливого запуска» создания Системы, основанный на методах бережливого производства (см. Приложение № 1) и предполагающий создание первой очереди Системы, реализующей одну или несколько Групп требований в соответствии с техническим заданием и разработанными и согласованными (при необходимости) проектными (техническими) решениями в рамках расчётных сроков и бюджета.

46. Подход «бережливого запуска» предполагает создание минимально достаточной версии Системы (МДВ), разрабатываемой в рамках запланированного периода, состоящего из одного или нескольких Инкрементов, выполняемых в пределах установленных сроков и бюджета до тех пор, пока не будет реализована и проверена функциональность Системы, предоставляющая клиентам необходимую ценность, либо разработанные проектные решения продемонстрирует свою несостоятельность, например, в силу превышения запланированных сроков и/или бюджета или невостребованности результата клиентами.

47. В рамках «бережливого запуска» проводятся исследовательские работы по выбору технологий реализации Системы и выбор соответствующей

инфраструктуры (программно-аппаратных средств) разработки. В том числе, проводится проверка возможности реализации предусмотренных в техническом задании требований (части требований) к программному обеспечению Системы посредством использования программ для электронных вычислительных машин, размещенных в Национальном фонде алгоритмов и программ, а в случае, если такие программы для электронных вычислительных машин выявлены в ходе указанной проверки, - установление наличия в техническом задании соответствующих требований по их использованию для создания Системы.

48. Если для создания Системы привлекаются внешние исполнители рекомендуется заключение отдельного Государственного контракта или соглашения о государственном частном партнёрстве, концессии или иных соглашений на создание МДВ.

49. Подход «бережливого запуска» упрощает проведение оценок сроков и бюджета создания последующих очередей Системы в связи с тем, что функциональность и продолжительность МДВ существенно меньше, чем функциональность Системы в целом. Оценка полученных результатов в ходе создания МДВ также позволит уточнить прогноз затрат в рамках реализации последующих очередей Системы, в том числе - Совокупную стоимость владения Системой в случае ее реализации в архитектуре, подтвержденной созданием МДВ.

50. Создание МДВ обеспечит:

- 1) предоставление ключевых возможностей Системы клиентам за минимальные сроки;
- 2) сокращение сроков выпуска последующих релизов Системы за счет сокращения мероприятий по вводу Системы в эксплуатацию, установленных руководящими документами, после ввода в эксплуатацию первого релиза Системы (МДВ);
- 3) получение обратной связи от клиентов Системы, что может быть использовано для корректировки работ по созданию и развитию Системы.

3. СТАДИЯ ПЛАНИРОВАНИЯ

51. **Стадия планирования** проводится Ведомством с целью выполнения всех подготовительных работ, необходимых для дальнейшего создания и/или развития Системы. При этом Ведомство берет на себя обязательства по организации процесса создания или развития Системы, в том числе, с учетом основных рекомендаций настоящего документа.

52. До начала стадии планирования Ведомство формирует **Рабочую группу по созданию Системы** в составе, утверждаемой решением должностного лица Ведомства, имеющего соответствующие полномочия. Рабочая группа, необходимая для выполнения планирования должна включать следующие категории:

1) **Владелец Системы** - должностные лица Ведомства, которые несут финансовую и операционную ответственность за создание и развитие Системы в течение всего ее жизненного цикла и обеспечение требуемых результатов оказания государственных услуг и выполнения государственных функций с использованием Системы. Владельцы Системы наделяются полномочиями по согласованию концепции, бюджета и технического задания на Систему внутри Ведомства, определяют приоритеты реализации функциональности в соответствии с концепцией и ТЗ, утверждают результаты финальной приемки каждого релиза Системы, а также принимают решение о вводе релиза Системы в эксплуатацию.

2) **Группа ответственных за Систему** – должностные лица и специалисты Ведомства, отвечающие за сопровождение процессов разработки Системы, разработку необходимых документов в ходе разработки Системы, в том числе - формирование и контроль реализации требований к Системе, соответствующих потребностям клиентов Системы, зафиксированных в ТЗ на Систему и в рамках установленного бюджета.

53. Группа ответственных за Систему в общем случае может включать следующих должностных лиц Ведомства:

1) руководитель Группы - директор департамента (начальник управления), курирующего создание Системы - определяет основные проектные и архитектурные решения и руководит приемкой Системы (версий Системы);

2) начальники управлений, отделов, ответственных за предоставление услуг, выполнение функций государственного управления и контроля - участвуют в разработке Концепции, ТЗ, а также в испытаниях Системы и приемке выполненных работ;

3) представитель (представители) управления (департамента) информационной безопасности Ведомства ответственные, в том числе, за внедрение и использование системы менеджмента информационной безопасности (СМИБ) в соответствии с рекомендациями ГОСТ Р ИСО/МЭК 27002-2021;

4) представитель (представители) управления информационных технологий и обеспечения эксплуатации Ведомства;

5) иные должностные лица и специалисты по решению руководителя Ведомства.

54. Стадия планирования, выполняемая созданной Рабочей группой, включает:

1) разработку проекта Концепции создания Системы, в соответствии с Методическими рекомендациями по формированию концепции создания государственной информационной системы, включающей:

- основания для создания Системы;
- предпосылки, предшествующие решению о создании Системы;
- роль и место Системы в государственном управлении Российской Федерации;
- анализ документов, регламентирующих разработку Системы;
- анализ решений аналогичного назначения;
- цели и задачи создания Системы, целевые показатели результативности создания Системы;
- описание возможных вариантов построения Системы и обоснование выбранного варианта реализации;
- формирование требований к системе защиты информации в соответствии с ГОСТ Р 51583-2014, и иным законодательством в сфере защиты информации, обрабатываемой в государственных информационных системах;
- план мероприятий по созданию Системы (проект);
- обеспечение правовой основы создания Системы.

2) разработку финансово-экономического обоснования (ФЭО) реализации Системы, включающего в себя сведения об общем объеме финансового обеспечения, рассчитанном на 3 года с момента принятия правового акта о создании Системы, необходимом для реализации мероприятий по созданию Системы, вводу ее в эксплуатацию, эксплуатации и в случае, если установлен срок эксплуатации Системы, для вывода Системы из эксплуатации и дальнейшего хранения содержащейся в ее базах данных информации (далее – совокупная стоимость владения);

3) разработку проекта правового акта, являющегося основанием для создания Системы;

4) обеспечение проведения оценки целесообразности проведения мероприятий по информатизации и (или) их финансирования Министерством цифрового развития, связи и массовых коммуникаций Российской Федерации в соответствии с Постановлением Правительства Российской Федерации от 1

июня 2004 г. № 260 «О Регламенте Правительства Российской Федерации и Положении об Аппарате Правительства Российской Федерации»;

5) обеспечение утверждения проекта Концепции, ФЭО и правового акта уполномоченным должностным лицом;

6) разработку, обеспечение согласования с уполномоченным федеральным органом исполнительной власти, осуществляющим функции по выработке и реализации государственной политики и нормативно-правовому регулированию в сфере информационных технологий (в соответствии с Постановлением № 676) и обеспечение утверждения уполномоченным должностным лицом Технического задания на создание или развитие Системы (и ее части), включая требования к защите информации и модели угроз безопасности информации, и технико-экономического обоснования (ТЭО) создания Системы в соответствии с требованиями и в порядке, установленными Постановлением № 676;

7) формирование Плана-графика итерационной разработки, определяющим очередность реализации сформированных в техническом задании Групп требований к Системе, плановые сроки реализации и критерии приемки каждой группы требований;

8) определение Исполнителя³ в соответствии с законодательством Российской Федерации, включение должностных и специалистов Исполнителя в состав Рабочей группы.

55. После определения Исполнителя и заключения Государственного контракта, концессионного или иного соглашения с Исполнителем, в Рабочую группу включаются его представители:

1) **Архитектор Системы** – технический специалист (или команда специалистов), которые разрабатывают и уточняют проектные (технические) решения, определяют архитектуру Системы и осуществляют архитектурный надзор в процессе создания и развития Системы.

2) **Системная команда** – команда для решения специальных задач по созданию инфраструктуры разработки, включающей необходимые программно-аппаратные средства для интеграции разработанной функциональности в составе Системы, проведения всех видов тестирования, демонстрации Систем и

3 «Исполнитель» - юридическое лицо независимо от организационно-правовой формы или индивидуальный предприниматель, определяемые в соответствии с законодательством Российской Федерации о контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд либо в соответствии с бюджетным законодательством, выполняющее работы по созданию или развитию государственной информационной системы либо оказывающее услуги по эксплуатации государственной информационной системы (Постановление № 676).

решений, развертывания новых версий Системы, а также предоставления доступа к инфраструктуре разработки для Рабочей группы.

56. Архитектор Системы в соответствии с требованиями, указанными в техническом задании, выполняет разработку решений по архитектуре Системы и документации на Систему, в том числе - технической документации в объеме, необходимом для описания полной совокупности проектных решений (в том числе по защите информации) и достаточном для дальнейшего выполнения работ по созданию Системы, а также описания проектных (технических) решений, обеспечивающих реализацию одного или нескольких процессов предоставления государственных услуг, государственных функций, включая контрольно-надзорную деятельность, подлежащих автоматизации и/или цифровой трансформации и реализуемых посредством Системы.

57. Группа ответственных за Систему осуществляет приёмку и обеспечивают согласование и утверждение документации на Систему и ее части в соответствии с порядком, установленным Постановлением № 676.

58. Системная команда подготавливает инфраструктуру (программно-аппаратных средств) разработки Системы с учетом согласованных проектных (технических) решений и решений по архитектуре Системы.

4. СТАДИЯ ВЫПОЛНЕНИЯ

60. **Стадия выполнения** в соответствии с итерационным подходом к разработке осуществляется в виде серий итераций, каждая из которых включает следующие виды работ, выполняемые итерационными циклами (см. Раздел 4.1.1):

1) разработка или адаптация программного обеспечения Системы с использованием итерационного подхода. Использование итерационного подхода позволит проводить регулярные демонстрации промежуточных результатов разработки Системы, оперативно корректировать проектные решения в рамках требований технического задания на создание или развитие Системы, а также осуществлять выпуск релизов Системы непосредственно в ходе ее создания;

2) разработка и уточнение рабочей документации на Систему и ее части, необходимой для выполнения работ по вводу Системы в эксплуатацию и ее эксплуатации, а также методической документации и порядка эксплуатации Системы, содержащего сведения, необходимые для выполнения работ по поддержанию уровня эксплуатационных характеристик (качества) Системы (в том числе по защите информации), установленных в проектных решениях;

3) пусконаладочные работы;

4) регулярное проведение системной демонстрации промежуточных результатов создания или развития Системы (рекомендуется каждые две недели);

5) проведение предварительных испытаний Системы по завершении реализации Групп требований.

61. На основе ТЭО, разрабатываемого на стадии «Планирование», определяется состав и численность разработчиков, привлекаемых для создания Системы. В зависимости от полученных оценок рекомендуется выбор одного из вариантов организации разработки с использованием итерационного подхода:

1) **базовый уровень**, предполагающий разработку Систем Рабочей группой численностью до 150⁴ человек (раздел 4.1),

2) **расширенный уровень**, включающий дополнительные, по отношению к базовому уровню, рекомендации по разработке Систем Рабочей группой численностью более 150 человек (раздел 4.2).

⁴ Указанная оценка является ориентировочной и может колебаться в диапазоне от 80 до 150 человек.

4.1 Базовый уровень

4.1.1 Общие рекомендации по организации работ на базовом уровне

62. Базовый уровень представляет собой минимальный набор ролей, методик и практик, необходимых для создания или развития Системы с использованием итерационного подхода. На базовом уровне рекомендуется выполнение работ коллективом до 150 человек.

63. В соответствии с итерационным подходом к разработке на базовом уровне работы выполняются итерационными циклами с рекомендуемой продолжительностью 12 недель (~ 1 квартал) - **Инкрементами**. В рамках каждого Инкремента по согласованию с представителями Ведомства, входящими в состав Рабочей группы, реализуется функциональность Системы (определяемая Группой требований), представляющая ценность для потребителей государственных услуг и государственных функций (клиентов), предоставляемых через Систему.

64. Непосредственное выполнение работ по созданию и развитию Системы выполняют **Команды разработки** – самостоятельные и самоорганизующиеся кросс-функциональные команды численностью 5...11 человек, имеющие в своём составе все необходимые компетенции для выполнения работ. Принципы и возможные варианты формирования Команд разработки указаны в разделе 2.4. За непосредственное формирование команд отвечает Исполнитель.

65. Для успешного применения настоящих Методических рекомендаций на базовом уровне рекомендуется соблюдение нижеперечисленных условий:

#1 Использование принципов бережливого производства и итерационного подхода в разработке (см. Приложение № 1). Эти принципы гарантируют, что выполнение работ осуществляется непрерывно, с минимальными временными потерями и с наилучшим качеством.

#2 Планирование Инкрементов с непосредственным участием всей Группы разработки Системы. Планирование определяет перечень приоритетных задач на очередной Инкремент и связывает стратегическое понимание создания (цели создания Системы) и развитие Системы с реализацией.

#3 Постоянная синхронизация статуса работ и корректировка взаимосвязанных планов Команд разработки, привлекаемых к созданию и развитию Системы. Синхронизация позволяет обеспечивать взаимодействие одновременно нескольких Команд разработки, ведущих разработку различных составных частей Системы, что в свою очередь обеспечивает устойчивость процесса разработки.

#4 Включение в Инкремент специальной итерации для осуществления планирования, проверки и корректировки результатов Инкремента, используемой в качестве резервного временного буфера для завершения ранее запланированных работ итерации, проведения необходимых для дальнейшей разработки Системы исследований, подведения итогов выполненных работ, а также для подготовки к планированию и проведению планирования последующего Инкремента.

#5 Проведение регулярной демонстрации Системы должностным лицам Ведомства и иным заинтересованным сторонам в интересах получения обратной связи, выявления и решения проблем, а также адаптации хода разработки к изменяющимся условиям.

#6 Обеспечение частых и качественных релизов Системы в соответствии с требованиями ТЗ в интересах получения обратной связи от клиентов.

4.1.2 Ролевая структура Рабочей группы Системы

66. Для непосредственного выполнения работ на базовом уровне Рабочая группа (Рисунок 9) по созданию Системы помимо ролей, указанных в разделе «Этап планирования», должна включать следующие роли.

1) **Команда разработки** - самоорганизующаяся кросс-функциональная команда, включающая группу специалистов из 5...11 человек, работающих в рамках Инкремента короткими фиксированными циклами (рекомендуется 2 недели) – Итерациями.

2) **Группа команд разработки** – несколько Команд разработки, совместно работающих над созданием и развитием Системы итерациями фиксированной длительности (рекомендуется – 12 недель) - Инкрементами.

3) **Менеджер процессов группы команд** – руководитель со стороны Исполнителя, отвечающий за организацию и содействие выполнению Группой команд Системы согласованных рабочих процессов, рекомендуемых настоящим документом, в интересах эффективной реализации требований ТЗ.

4) **Ответственный за реализацию** - член Команды разработки, ответственный за определение приоритетов и реализацию требований командой в рамках двухнедельных итераций (циклов) при сохранении концептуальной и технической целостности компонентов Системы и ее устойчивого функционирования.

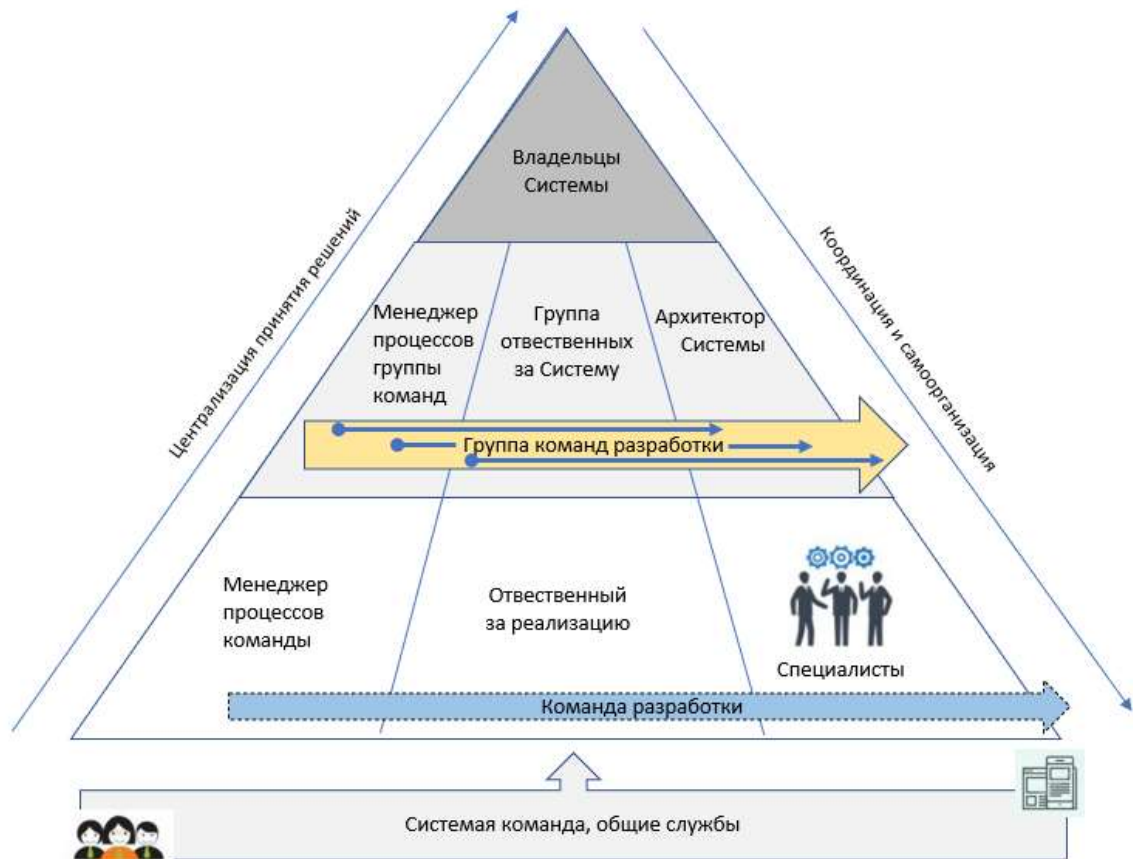


Рисунок 9. Структура Рабочей группы базового уровня

5) **Менеджер процессов команды** – член команды разработки, основной обязанностью которого является содействие команде в следовании рабочим процессам в соответствии с требованиями настоящего документа.

67. Формирование Рабочей группы в соответствии с п.66 обеспечивает Исполнитель.

68. Команды разработки должны иметь в своем составе специалистов, обеспечивающих во взаимодействии с Системной командой и другими Командами разработки проектирование, разработку, тестирование и развертывание функциональности Системы, предоставляющую ценность для клиентов.

69. Обязанности членов Рабочей группы определены в Приложении № 3 к настоящему документу.

4.1.3 Выполнение работ по созданию и развитию Системы на базовом уровне на основе итерационного подхода

70. Типовая схема выполнения работ на основе итерационного подхода представлена на рисунке 7. В общем случае предполагается, что стадия выполнения работ выполняется серией Инкрементов, каждый из которых включает несколько итераций, реализующих:

- 1) планирование Инкремента,
- 2) выполнение Инкремента,
- 3) проверки результатов Инкремента,
- 4) корректировки выполняемых работ.

71. Рабочая группа может изменять типовую схему исходя из практического опыта. При этом все Команды разработки, входящие в Группу, должны иметь одну и ту же частоту внутренних циклов выполнения работ.

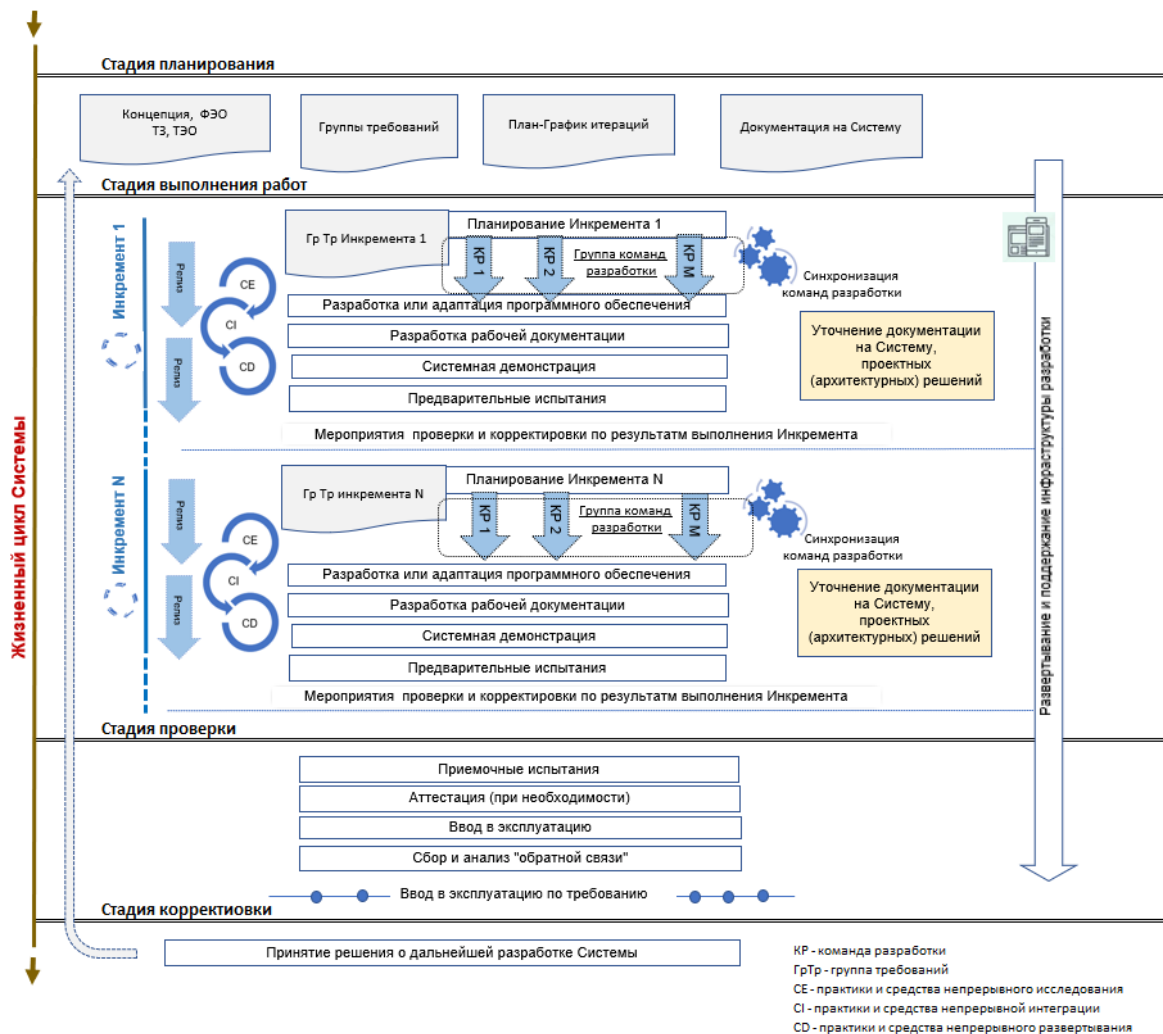


Рисунок 10. Типовая схема выполнения работ на базовом уровне

4.1.3.1 Планирование Инкремента

72. Планирование Инкремента — это регулярное мероприятие, которое служит для выработки единых целей, задач и согласования взаимосвязанных работ всех Команд разработки, участвующих в создании и развитии Системы. Планирование Инкремента включает подготовку к планированию и собственно проведение планирования.

73. В Планировании Инкремента участвуют Владельцы Системы, Группа ответственных за Систему, Команды разработки, архитекторы / инженеры Системы, системная команда и другие заинтересованные стороны. Все участники должны быть уведомлены заранее о проведении планирования Инкремента для подготовки к планированию.

74. Подготовка к планированию Инкремента осуществляется Группой ответственных за Систему с помощью Менеджера процессов Группы команд и с привлечением Архитектора Системы.

75. При подготовке к планированию:

1) Группа ответственных за Систему уточняет перечень задач, которые планируется выполнить в рамках предстоящей итерации, в том числе:

- используя предстоящую к реализации в рамках планируемого Инкремента Группу требований, формируют цели (ожидаемые результаты) следующего Инкремента;

- проводят работу с Ответственными за реализацию от Команд разработки по определению для каждой команды приоритетных функций предстоящего Инкремента;

- определяют и оценивают основные факторы, риски и проблемы, влияющие на разработку Системы;

- оценивает достаточность ресурсов Группы Команд разработки для выполнения предстоящего Инкремента.

2) Архитектор Системы с учетом предполагаемой к реализации функциональности Системы:

- уточняет проектные (архитектурные) решения, обеспечивающие реализацию и интеграцию функциональности, реализуемой различными Командами разработки в предстоящей итерации;

- определяет необходимость создания вспомогательных технических и программных средств, необходимых для создания новых функций и возможностей Системы;

- определяет основные технологии, планируемые к использованию при разработке Системы и нефункциональные требования.

3) Менеджера процессов Группы команд совместно с Группой ответственных за Систему:

- определяет место проведения планирования, которое должно обеспечивать условия работы всех членов Рабочей группы разработки, участвующих в планировании;

- обеспечивает доступ в режиме реального времени к информации и инструментам для поддержки распределенного планирования или удаленных участников;

- обеспечивает готовность аудио и видео каналов для взаимодействия участников планирования, а также наличие иных презентационных материалов и технических средств.

76. По результатам подготовки формируются и предоставляются участникам планирования следующие входные данные и документы:

1) Концепция, ТЗ, План-график итерационной разработки, а также, при наличии - дополнения к ТЗ;

2) приоритеты реализации функциональных нефункциональных требований в составе Группы требований в предстоящем Инкременте;

3) контекст выполнения планируемых работ, определяющий текущее состояние разработки Системы, применяемые технологии и средства для выполнения работ, ранее выявленные проблемы выполнения работ, внешние зависимости выполняемых работ от поставщиков и иных организаций (например, надзорных органов).

77. **Непосредственное планирование** проводится, как правило, в течении двух дней.

78. **Первый день** планирования Инкремента включает следующие мероприятия.

1) Менеджер процессов группы команд представляет регламент процесса планирования и ожидаемые результаты.

2) Владельцы Системы описывают текущее состояние разработки Системы и дает оценку - насколько эффективно существующие решения удовлетворяют текущие потребности клиентов, а также определяет ожидаемые результаты (ценности для клиентов), которые должны быть реализованы в ходе выполнения последующего Инкремента.

3) Группа ответственных за Систему представляет предложения по работам в рамках следующего Инкремента.

4) Архитектор / Инженер представляет архитектурные решения для реализации функциональности в предстоящем Инкременте, а также соответствующие изменения в методах разработки и в использовании практик DevSecOps, таких как непрерывная интеграция, непрерывное развертывание и автоматизация тестирования.

5) Команды разработки оценивают свои возможности и формируют проекты планов работ на предстоящий Инкремент, а также зависимости от выполнения работ других Команд разработки.

6) Команды разработки представляют участникам Планирования ключевые результаты планирования, которые включают в себя цели, трудоемкость, потенциальные риски, проблемы и зависимости.

Проводится анализ и решение проблем, связанных с объемом работы, ресурсными ограничениями и зависимостями. Члены Рабочей группы должны договориться об изменении объема работ и согласовать иные корректировки в планах каждой Команды разработки. При этом Менеджер процессов группы команд консультирует и обеспечивает совместную работу заинтересованных сторон до тех пор, пока не будут согласованы планы Команд разработки. В качестве инструмента достижения консенсуса рекомендуется использование досок зависимостей, визуализирующих задачи каждой Команды разработки и связи с задачами, решаемыми другими командами (Рисунок 11).

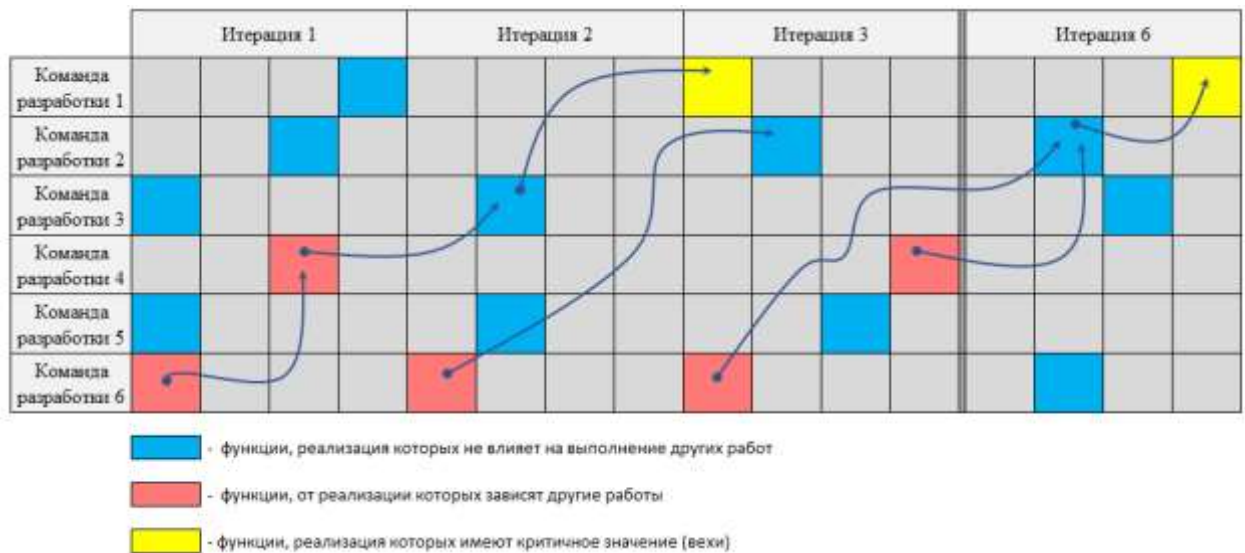


Рисунок 11. Доска зависимостей между командами разработки

79. Второй день планирования Инкремента включает следующие мероприятия.

1) Владельцы системы представляют изменения в объеме планирования и ресурсах по результатам договорённостей первого дня планирования.

2) Команды вносят коррективы в свои первоначальные планы, а также окончательно определяют свои цели на предстоящую итерацию и представляют свои планы всем участникам планирования.

3) Если Владельцы Системы считают план неприемлемым, командам предоставляется возможность их скорректировать.

4) Владельцы Системы определяют важности целей⁵ (см. п. **Error! Reference source not found.**) для каждой Команды разработки на предстоящую итерацию.

5) Проводится обсуждение рисков и зависимостей, которые во время планирования выявили Команды разработки. Риски рассматриваются и классифицируются в одну из следующих категорий:

- команды согласны с тем, что риск больше не является проблемой;
- команды принимают риски и планируют мероприятия по управлению этими рисками в ходе выполнения Инкремента с целью снижения влияния риска;
- команды эскалируют риск на уровень Группы ответственных за Систему и/или Менеджера процесса группы команд – в случае если у команды нет полномочий и возможностей по управлению риском.

6) Команды голосуют о своей уверенности в достижении целей Инкремента с учетом выявленных рисков. При необходимости команды перерабатывают свои планы до тех пор, пока не будет достигнут высокий уровень доверия (~ 80%).

7) Менеджер процессов группы команд проводит краткую ретроспективу планирования.

80. По результатам планирования должны быть получены два основных результата:

1) Цели предстоящего Инкремента для каждой Команды разработки. При этом цели объединяют цели развития функциональности Системы и технические цели, которые Группа создания Системы намерена реализовать.

2) План работ для каждой Команды разработки с указанием дат реализации новых функций, с учетом наличия зависимостей между работами других командам.

81. Все результаты процесса планирования фиксируются в целях Инкремента (Рисунок 9) для каждой Команды разработки.

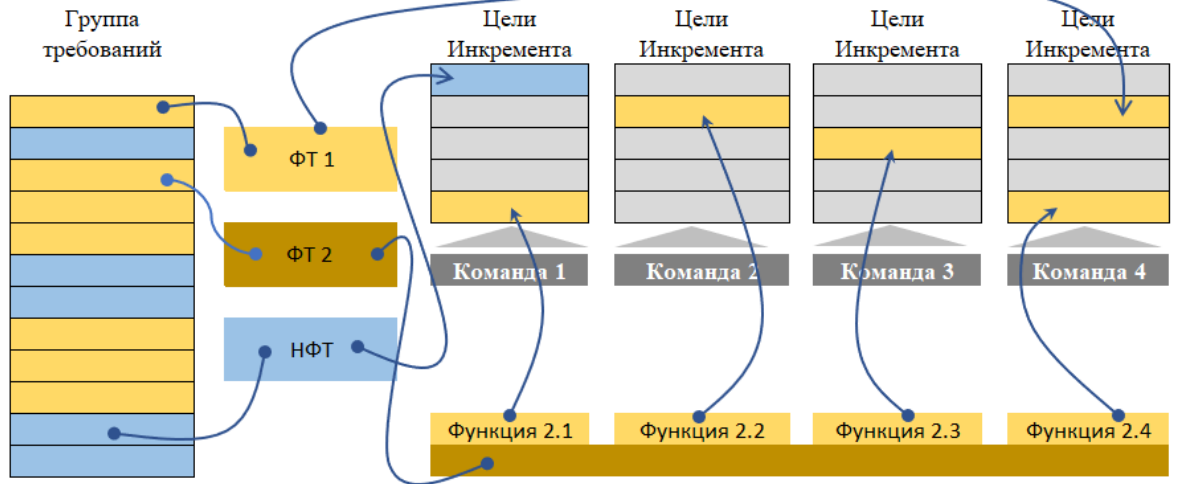
⁵ Важность целей определяется весовыми коэффициентами (оценками), для определения которых могут использоваться как неформальные, так и формальные методы, например указанные в Приложении № 1.

Цели Инкремента № ____		ВЦ
1	Формирование отчетов о выполнении плана	
2	Периодический контроль выполнения работ	
3	Автоматическая регистрация выполнения работ	
4	Автоматическое уведомление об отклонениях от плана	
5	Ведение журнала оперативных событий	
6	Формирование отчетов о выполнении плана	
7	Экспорт отчетов	

ВЦ -важность цели

Рисунок 12. Цели Инкремента для Команды разработки

82. Цели каждой Команды разработки часто напрямую связаны с предполагаемыми к реализации функциями. Однако такое сопоставление не всегда является однозначным, поскольку некоторые функции требуют совместной работы нескольких команд, как показано на рисунке 10. Некоторые функции (например, компонент ФТ 1) могут быть реализованы отдельной командой, тогда как другие (компонент ФТ 2) требуют совместной работы нескольких команд. Дополнительно для команд в ходе планирования могут быть определены технические цели, которые обеспечивают реализацию функциональности, а также служат улучшению инфраструктуры разработки (НФТ).



ФТ - функциональное требование
 НФТ - нефункциональное требование

Рисунок 13. Распределение работ между Командами разработки

83. Существуют обстоятельства, при которых по результатам планирования у команды остается низкая уверенность в достижении целей:

- 1) зависимости от другой команды или поставщика, которые не могут быть гарантированы;
- 2) команда не имеет опыта работы с функциональностью данного типа;
- 3) команда уже загружена почти на полную мощность, выполняя критические задачи, от которых зависит успех Инкремента в целом.

Такие цели (не рекомендуется более трёх) также могут быть включены в План работ команды на предстоящий Инкремент и для их выполнения должны быть спланированы соответствующие ресурсы. При этом команды не фиксируют их в качестве своих обязательств, чтобы в случае невыполнения не ухудшать свои показатели по результатам выполнения Инкремента. Дополнительные цели, обязательства достижения которых не определены, называются **незафиксированными целями**.

84. После завершения определения целей Инкремента, Владельцы Системы с привлечением Группы ответственных за Систему определяют важность целей для каждой Команды разработки в рамках очного обсуждения с ними в ходе второго дня планирования (Рисунок 14). При этом следует выполнять следующие правила:

- 1) использовать шкалу от 1 (самая низкая) до 10 (самая высокая) для оценки каждой цели;

2) оценки не должны быть «нормализованы»^б между командами, то есть у каждой команды может быть несколько самых приоритетных (рейтинг 10) пунктов;

3) принимать к учету как цели, которые обеспечивают прямую и непосредственную ценность, так и другие цели, которые обеспечивают создание будущей ценности, такие как, например, изменения в области инфраструктуры, сред разработки и инициатив в области качества;

4) включать в оценку незафиксированные цели.

Цели Инкремента №		ВЦ
1	Формирование отчетов о выполнении плана	6
2	Периодический контроль выполнения работ	2
3	Автоматическая регистрация выполнения работ	10
4	Автоматическое уведомление об отклонениях от плана	8
5	Ведение журнала оперативных событий	3
6	Формирование отчетов о выполнении плана	7
7	Экспорт отчетов	2
Незафиксированные цели		
9	Реестр контрольных мероприятий	2
10	Реестр оперативных событий	3

ВЦ -важность цели

Рисунок 14. Конечные цели Команды разработки с незафиксированной целью и оценкой ценности

4.1.3.2 Выполнение Инкрементов

85. Инкремент выполняется Группой команд разработки, при этом каждая Команда разработки, входящая в состав Группы команд, обеспечивает достижение целей, согласованных в ходе Планирования инкремента, а также реализацию зависимостей с другими командами, при этом план (порядок) выполнения работ (итераций) может корректироваться.

86. Команды разработки выполняют работы синхронно, итерациями фиксированной продолжительности (рекомендуется – 2 недели) с проведением синхронизации в рамках Группы команд разработки после завершения каждой итерации (Рисунок 15). Организация процесса итерации определяется выбранным методом итерационной разработки. Например, для Команд разработки, использующих метод Скрам (Приложение 1) итерация включает:

^б Под нормализацией в общем случае понимается преобразование всего набора данных к безразмерным единицам в рамках заданного диапазона, например: 1...10

1) планирование итерации, предполагающее формирование задач и целей на итерацию и определение приоритетов их выполнения;

2) реализацию функциональных и нефункциональных требований, в соответствии с планом и с учетом коррекций и согласований выполняемых работ между специалистами команды во время ежедневных совещаний;

3) поверку результатов работ в ходе итерации, включающую подведение итогов итерации и демонстрацию реализованной функциональности Системы Ответственному за реализацию, с целью получения обратной связи;

4) корректировку – проведение ретроспективы итерации, включающей оценку процесса разработки, а также выявление проблем, которые необходимо решить на следующей итерации.



Рисунок 15. Мероприятия Группы команд разработки на базовом уровне

87. За контроль соответствия процессов выбранному методу итерационной разработки, а также наставничество над специалистами Команды разработки отвечает Менеджер процессов.

88. В ходе выполнения работ Команды разработки в рамках согласованных Планов работ осуществляют:

1) разработку и адаптацию программного обеспечения – реализацию функциональных и нефункциональных требований к Системе;

2) разработку и актуализацию рабочей документации;

3) проведение всех видов тестирования – функционального, интеграционного, регрессионного, а также, при необходимости - нагрузочного;

4) выполнение иных мероприятий, предусмотренные согласованным Планом работ.

89. В качестве основного инструмента для выполнения работ Команде разработки рекомендуется использовать Конвейер непрерывной разработки, основанный на принципах DevSecOps (Приложение №1) и обеспечивающий повторяемые рабочие процессы и их автоматизацию, необходимые для создания новой функциональности Системы, начиная от разработки проектных решений до выпуска релиза в эксплуатацию и, в последующем - обеспечения эксплуатации. Конвейер состоит из четырех компонентов (Рисунок 16):

1) Непрерывное исследование (CE) – набор практик и необходимых технических и программных средств, обеспечивающий возможность Командам разработки постоянно повышать уровень знаний в своей предметной области, формировать технологический задел и определять рациональные пути создания и развития Системы, а также быстро адаптировать средства разработки и применяемые технологии к изменяющимся потребностям клиентов (ключевой практикой CE является создание МДВ).

2) Непрерывная интеграция (CI) – набор практик и средств автоматизации процессов разработки, сокращающих временные затраты на проверку и выпуск версий Системы путём регулярного объединения программного кода, разработанного разными командами в специальном репозитории – тестовом контуре, с последующей автоматической сборкой, тестированием и запуском Системы с реализованной новой функциональностью.

3) Непрерывное развертывание (CD), технология автоматизации сборки версии Системы, тестирования и развертывания с последующим автоматическим (без ручных вмешательств) развертыванием в среде эксплуатации.

4) Релиз по требованию (непрерывная доставка) -технология, обеспечивающая нахождение Системы в состоянии готовности к передаче в эксплуатацию, с учетом последних изменений, внесенных Командами разработки.

90. Дополнительно используемые инструменты Команд разработки должны соответствовать требованиям ГОСТ Р 56939, ГОСТ Р 58412 и включать инструменты, обеспечивающие автоматические проверки информационной безопасности и мониторинг Системы на предмет атак и утечек данных на всех участках производственного процесса, включая планирование, программирование, интеграцию, тестирование, выпуск релиза и развертывание.

91. Наличие инструментов DevSecOps обеспечивают Исполнитель, при этом Группа ответственных за Систему проводит контроль развернутых инструментов DevSecOps на этапе планирования создания и развития Системы,

в том числе на соответствие используемого конвейера требованиям информационной безопасности.

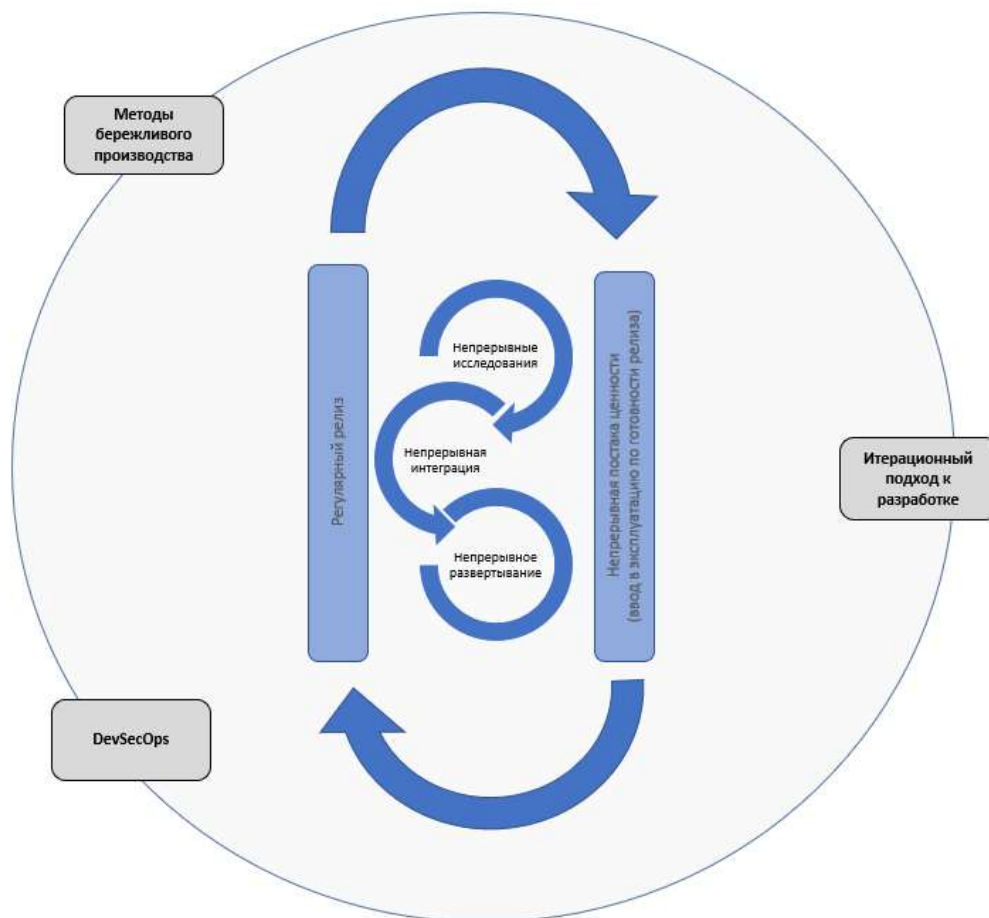


Рисунок 16. Конвейер непрерывной разработки

92. В ходе выполнения работ Командам разработки рекомендуется использование методов визуализации и управления задачами, например, с использованием досок Канбан (Приложение №1), обеспечивающих:

- 1) демонстрацию текущего статуса выполнения работ и проблемных мест, в случае их возникновения;
- 2) соблюдение ограничений на количество одновременно выполняемых работ с учетом ресурсных возможностей Команды разработки;
- 3) определение «узких» мест в процессе разработки и возможностей для их разрешения.

Ведение досок Канбан возлагается на представителей Команд разработки, а контроль – на Менеджера процессов команд.

93. С целью минимизации влияния рисков и контроля влияния зависимостей в ходе выполнения работ проводятся регулярные мероприятия по синхронизации работ, выполняемых различными Командами разработки:

1) Менеджер процессов группы команд и представители каждой Команды разработки (как правило – Менеджеры процессов команды) регулярно проводят еженедельные встречи и рассматривают прогресс в достижении целей Инкремента, выявляют проблемы, а также проводят анализ принятых рисков зависимостей между командами. Результаты еженедельных встреч представляются Группе ответственных за Систему.

2) Группа ответственных за Систему с привлечением Ответственных за реализацию от каждой Команды разработки проводит еженедельные совещания с целью получить представление о том, насколько хорошо Группа команд разработки обеспечивает достижение целей Инкремента, обсудить проблемы с разработкой функций и выработать, в случае необходимости, корректировки. Для подготовки к мероприятиям по синхронизации необходимые материалы готовят и предоставляют Ответственные за реализацию.

94. Команды разработки и Архитектор в соответствии с техническим заданием разрабатывают **рабочую документацию**, включающую техническую документацию, необходимую для выполнения работ по вводу Системы в эксплуатацию и ее эксплуатации, методическую документацию и порядок эксплуатации Системы, содержащего сведения, необходимые для выполнения работ по поддержанию уровня эксплуатационных характеристик (качества) Системы (в том числе по защите информации), установленных в проектных (технических) решениях, в том числе:

1) перечень действий сотрудников при выполнении задач по эксплуатации Системы, включая перечень, виды, объемы и периодичность выполнения работ по обеспечению функционирования Системы;

2) контроль работоспособности Системы и компонентов, обеспечивающих защиту информации;

3) перечень неисправностей, которые могут возникнуть в процессе эксплуатации Системы, и рекомендации в отношении действий при их возникновении;

4) перечень режимов работы Системы и их характеристики, а также порядок и правила перевода Системы с одного режима работы на другой с указанием необходимого для этого времени.

95. Согласование и утверждение рабочей документации осуществляет Группа ответственных за Систему в порядке, установленном Постановлением № 676.

96. В конце каждого Инкремента должна быть выполнена интеграция результатов работ, выполненных различными Командами разработки.

97. В интересах сокращения временных и ресурсных затрат на подготовку к интеграции в рамках Инкремента может выполняться частичная интеграция реализованной функциональности в рамках одной или нескольких итераций, выполняемых Командами разработки. Решение о частичной интеграции принимает Группа ответственных за Систему и согласовывает его с Ответственными за реализацию.

98. Для подтверждения успешности интеграции результатов выполненных работ Команды разработки проводят совместное интеграционное и регрессионное тестирование Системы с установленными обновлениями функциональности в соответствии с предварительно разработанными и согласованными с Группой ответственных за Систему программами и методиками тестирования.

99. В ходе выполнения работ рекомендуется использовать практику раннего тестирования. Вместо того, чтобы выполнять большое количество тестов в конце Инкремента, Команды разработки должны выполнять тесты на ранних стадиях разработки кода – в рамках двухнедельных итераций. Раннее тестирование применяется как к функциональным требованиям, так и к нефункциональным требованиям, к производительности, надежности и другим требованиям к качеству Системы, указанных в ТЗ.

Поскольку тесты могут расти с течением времени (по мере роста объема кода) и они начинают вносить задержки в работу команд допустимо использование сокращенных тестов. Кроме этого, повышение скорости тестирования может быть обеспечено использованием более производительного оборудования в тестовых контурах. Для этого Команды разработки должны взаимодействовать с Системной командой, чтобы сбалансировать скорость и качество тестирования.

4.1.3.3 Проверка результатов Инкремента

100. Проверка результатов Инкремента осуществляется путём поведения демонстрации реализованных всеми Командами разработки запланированных функциональных возможностей Системы. Организует проведение демонстрации Менеджер процессов группы команд совместно с Группой ответственных за Систему.

101. Перед демонстрацией Системы должны быть выполнены пусконаладочные работы, включающие автономную наладку технических средств и разработанного (или адаптированного) программного обеспечения Системы, загрузку информации в ее базу данных, комплексную наладку и

тестирование технических средств и программного обеспечения Системы, включая средства защиты информации.

102. **Демонстрация Системы** — важное событие, которое обеспечивает комплексное представление новых функций, представленной Группой команд разработки в рамках Инкремента. Каждая демонстрация дает Владельцам Системы, Группе ответственных за Систему и иным заинтересованным сторонам объективную оценку прогресса разработки Системы. По результатам демонстрации Системы проводится оценка текущего состояния разработки Системы на основе критических отзывов представителей Группы ответственных за Систему.

103. Демонстрация проводится в тестовом контуре, который максимально точно должен соответствовать контуру эксплуатации Системы. В рамках подготовки к демонстрации:

- 1) Команды разработки разрабатывают и согласовывают методики регрессионного и интеграционного тестирования;
- 2) Системная команда проводит установку и настройку тестового контура Системы, включая подготовку тестовых данных, не содержащих конфиденциальной информации и персональных данных;
- 3) Команды разработки устанавливают обновления Системы, включающие реализованные как функциональные, так и нефункциональные требования, а также проводят интеграционное и регрессионное тестирование;
- 4) Проводится тестирование в соответствии с разработанными и согласованными программами и методиками;
- 5) По результатам тестирования фиксируются выявленные ошибки;
- 6) Принимается решение о исправлении выявленных ошибок – либо в рамках текущего Инкремента, либо в рамках следующего Инкремента.

104. В процессе выполнения работ по созданию и развитию Системы Группа команд разработки должна обеспечивать готовность к развертыванию релиза Системы в контуре эксплуатации каждый раз, когда это требуется Ведомству. При этом в состав релиза включаются функции, реализованные и протестированные в рамках предшествующих Инкрементов.

105. В большинстве ситуаций завершение Инкремента и планируемый момент выпуска релиза Системы могут совпадать, но в некоторых случаях может потребоваться выпускать релиз Системы более или менее часто (например, через полтора Инкремента или через пол Инкремента).

106. Решение о развертывании релиза Системы, определение состава релиза и предоставление доступа к реализованной функциональности клиентам

принимают Владельцы Системы на основе предложений, разработанных Группой ответственных за Систему.

107. Развёртывание релиза Системы предполагает:

- 1) выполнение пусконаладочных работ;
- 2) проведение предварительных испытаний Системы, включая;
- 3) устранение выявленных ошибок.

108. Выполнение пусконаладочных работ предполагает наладку технических и программных средств, загрузку (заполнение, например, баз данных) Системы первоначальной информацией необходимой для проведения испытаний и начала функционирования Системы, комплексную наладку технических средств и программного обеспечения Системы, включая средства защиты информации.

109. Проведение предварительных испытаний включает:

1) разработку программы и методики предварительных испытаний, в соответствии с которыми осуществляется проверка Системы на работоспособность и соответствие техническому заданию на ее создание;

2) проверку Системы на работоспособность и соответствие техническому заданию на ее создание;

3) устранение выявленных при проведении таких испытаний неисправностей и внесение изменений в документацию и рабочую документацию на Систему;

4) оформление протокола испытаний и акта о приемке Системы в опытную эксплуатацию.

110. По решению Владельца Системы по результатам демонстрации версия Системы, реализующая функциональность одной или нескольких групп требований и обеспечивающая определенные возможности клиентам Системы, может быть введена в эксплуатацию. С этой целью выполняются мероприятия Стадии проверки выполненных работ, указанные в разделе 5 настоящего документа. При этом в особых случаях по решению Владельца Системы «промежуточный» релиз Системы может быть выпущен до завершения устранения выявленных ошибок, если они не приводят к невыполнению требований, указанных в ТЗ.

4.1.3.4 Корректировка выполняемых работ по результатам Инкремента

111. Корректировка выполняемых работ по созданию и развитию Системы по результатам Инкремента проводится в рамках мероприятия «Проверка и Адаптация», организуемого Менеджером процессов группы команд с привлечением Группы ответственных за Систему.

112. **Проверка и адаптация** выполняется по завершению Инкремента и является временем для анализа выполненных работ, решения проблем и принятия мер по улучшению, необходимых для увеличения скорости, качества и надежности следующего Инкремента.

113. Участниками мероприятия «Проверки и Адаптация» должны быть, при возможности, все члены Рабочей группы.

114. Мероприятие «Проверка и адаптация» (ПА) включает:

- 1) рассмотрение количественных и качественных показателей результатов выполненных работ;
- 2) проведение ретроспективы;
- 3) выполнение дополнительных мероприятий по решению представителей Группы ответственных за Систему и Ответственных за реализацию.

Цели Инкремента №		ВЦ	
		План	Факт
1	Формирование отчетов о выполнении плана	6	5
2	Периодический контроль выполнения работ	2	2
3	Автоматическая регистрация выполнения работ	10	7
4	Автоматическое уведомление об отклонениях от плана	8	8
5	Ведение журнала оперативных событий	3	3
6	Формирование отчетов о выполнении плана	7	6
7	Экспорт отчетов	2	2
Незафиксированные цели			
9	Реестр контрольных мероприятий	2	0
10	Реестр оперативных событий	3	3
ИТОГО		43	36
% достижения цели		84%	

ВЦ - важность цели

Рисунок 17. Оценка результатов квартального цикла

115. Во время **рассмотрения количественных и качественных показателей** рассматриваются все цели, которые Команды разработки определили в ходе планирования Инкремента. В ходе мероприятия «Проверка и адаптация» Группа ответственных за Систему совместно с представителями

каждой Команды разработки оценивает результаты, достигнутые для каждой из поставленных в ходе планирования Инкремента целей (Рисунок 17), на основе которых могут выполнены оценки результатов выполнения Инкремента в целом, например по сумме оценок для каждой цели (за исключением оценок незафиксированных целей).

116. Одним из основных показателей, принимаемый к учету для оценки результатов Команд разработки является показатель предсказуемости результатов Команд разработки, отражающий отношение фактических и запланированных оценок выполнения Инкрементов каждой Командой разработки на интервале нескольких Инкрементов (Рисунок 18).

117. Надежные Команды разработки должны работать в диапазоне предсказуемости 80–100 %, что в свою очередь обеспечивает предпосылки для надежного планирования. При этом незафиксированные цели не учитываются в обязательствах команд, но учитываются в фактическом достижении цели.

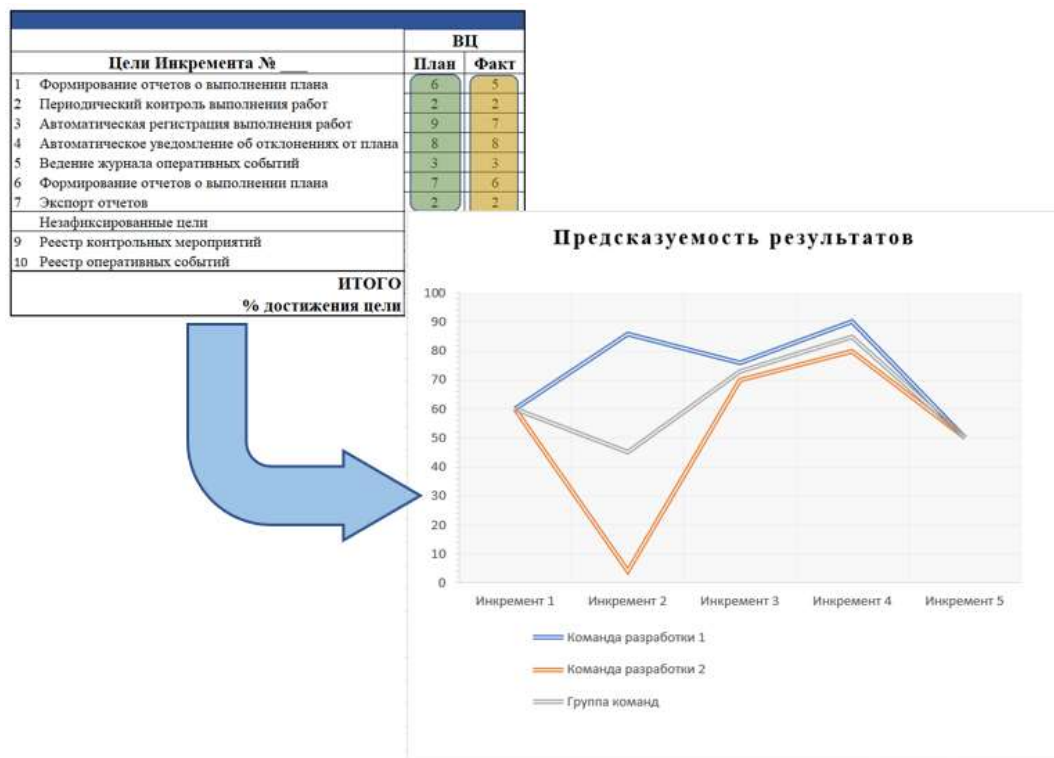


Рисунок 18. Показатель предсказуемости работы Команды разработки

118. **Ретроспектива** проводится с целью определить проблемы, с которыми столкнулась Рабочая группа в рамках текущего Инкремента и провести их дальнейшее обсуждение и решение в ходе ретроспективных семинаров. При этом, каждая проблема может обсуждаться с привлечением Команд в полном составе или с привлечением специально сформированных групп из представителей разных Команд разработки. Такой подход к

организации проведения семинара позволяет обеспечить различные взгляды на проблему, и учесть мнение тех, кто больше всего мотивирован для решения проблемы.

119. Ретроспективный семинар организуют и проводит Менеджер процессов группы команд с привлечением Менеджеров процессов команд разработки. Владельцы Системы и Группа ответственных за Систему принимают участие в ретроспективных семинарах в интересах обеспечения решения проблем, которые находятся вне сферы ответственности Команд разработки.

120. В ходе ретроспективного семинара проводится выявление источников выявленных проблем, ранжирование источников проблем и определения инструментов и способов их решения, а также формируются предложения по учету сформированных рекомендаций в ходе планирования следующего Инкремента.

121. Дополнительно в рамках мероприятий проверки и адаптации по решению Группы ответственных за Систему и Ответственных за реализацию могут проводиться:

1) Работа над технической инфраструктурой и инфраструктурой поддержания рабочих процессов, в том числе:

- развитие новых инструментов конвейера непрерывной разработки;
- внедрение и поддержку инструментов автоматизации тестирования;
- внедрение инструментов управления работами.

2) Обучение (развитие навыков) Рабочей группы, а также проведение исследовательских работ, необходимых для последующего создания и развития Системы.

3) Устранение ошибок, выявленных в ходе тестирования новой функциональности Системы, разработанной в рамках текущего Инкремента.

4.2 Расширенный уровень

4.2.1 Общие рекомендации по организации работ на расширенном уровне

122. Расширенный уровень представляет собой набор ролей, методик и практик, необходимых для создания или развития Систем, (представляющих собой крупномасштабные решения)⁷ в формате итерационной разработки, требующих ресурсных возможностей больших, чем 150 человек.

123. Увеличение численности разработчиков, как правило, обусловлено увеличением сложности и масштаба разрабатываемых Систем, состоящих из взаимосвязанных функциональных подсистем, каждая из которых имеет самостоятельное значение для предоставления государственных услуг клиентам и реализацию государственных функций.

124. Рекомендации по использованию расширенного уровня являются опциональными и используются по решению Ведомства. При принятии решения Ведомство должно учитывать, что необходимость использования расширенного уровня обусловлена следующими обстоятельствами:

1) Мероприятий по организации работ, принятых на базовой уровне, может оказаться недостаточно для координации большого числа Групп разработки. Также может возникнуть необходимость в корректировке принятых на базовом уровне процессов с целью снижения временных и ресурсных издержек. Например, планирование Инкремента с привлечением всей многочисленной Рабочей группы (более 150 человек) может оказаться экономически нецелесообразным.

2) С ростом сложности создаваемых Систем возрастает цена ошибки за некорректные проектные и архитектурные решения. Сбой в работе крупных решений может иметь неприемлемые социальные или экономические последствия. Поэтому требуются дополнительные меры контроля результатов и процессов создания таких Систем, а также проверки их на соответствие отраслевым и нормативным стандартам и предоставления объективных доказательств соответствия требованиям ТЗ.

4.2.2 Ролевая структура

125. Ролевая модель разработки на расширенном уровне (Рисунок 19) основана на ролевой модели базового уровня, при этом:

⁷ Крупномасштабные системы обеспечивают поддержку нескольких потоков ценностей для потребителей Государственных услуг и имеют совокупную стоимость владения более 100 млн.руб.,

1) Несколько Групп команд разработки объединяются в Пул команд разработки.

2) Роли, установленные для базового уровня, предполагают выполнение задач аналогичных задачам базового уровня рамках разработки функциональных подсистем. На уровне разработки функциональных подсистем ролевая структура включает:

- Группу ответственных за подсистему;
- Архитектора подсистемы;
- Группу команд разработки (подсистемы);
- Менеджера процессов Группы команд.

3) На расширенном уровне в состав Рабочей группы вводятся роли, обеспечивающие создание Системы в целом:

- Группа ответственных за Систему, включающих в том числе Ответственных за подсистему - должностных лиц Ведомства, отвечающих за предоставление клиентам определенной функциональности;

- Главный архитектор Системы, которому функционально подчинены архитекторы подсистем;

- Менеджер процессов пула команд разработки, координирующего деятельность менеджеров процессов Группы команд.

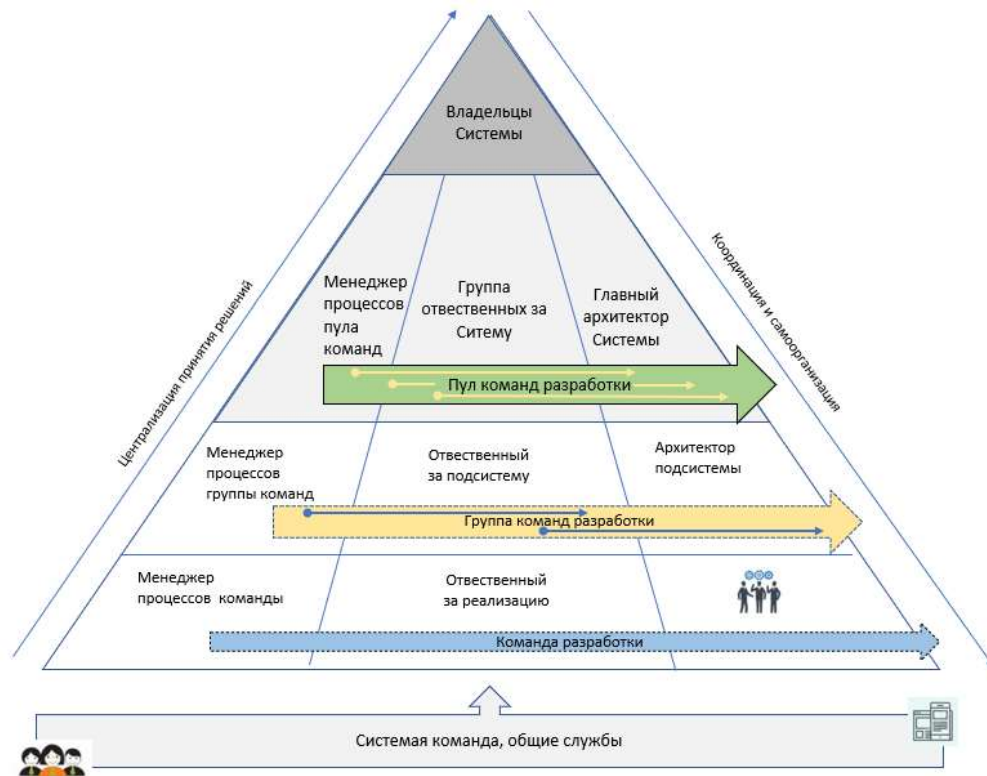


Рисунок 19. Ролевая структура на расширенном уровне

126. Пул команд разработки — это организационная конструкция, обеспечивающая создание и развитие Системы на расширенном уровне. Пул команд разработки координирует работы нескольких Групп команд разработки, каждая из которых обеспечивает разработку отдельных функциональных подсистем Системы. Пул команд разработки в общем случае может включать следующие типы команд:

- 1) Группы команд разработки, решающие задачи разработки (или адаптации) программного обеспечения подсистем.
- 2) Группы команд специальных подсистем, обеспечивающие создание подсистем, требующих глубоких специальных навыков и опыта, например, подсистем информационной безопасности.
- 3) Группы команд платформы, решающих задачи разработки и поддержки новых технологий и технологических платформ, на базе которых создается Система, и предоставляющих услуги другим командам.

127. Главный архитектор Системы, определяет общее техническое и архитектурное видение разрабатываемых проектных решений и организует работу команды, включающей Архитекторов подсистем.

128. Группа ответственных за Систему отвечает за создание Системы, управляет и координирует работу Групп ответственных за подсистемы.

129. Менеджер процессов пула команд является руководителем, отвечающим за организацию и содействие выполнению Пулом команд разработки согласованных рабочих процессов, рекомендуемых настоящим документом.

130. Дополнительно в состав Рабочей группы могут быть включены:

- 1) **Системная команда** обеспечивает инфраструктуру разработки и развертывания Системы.
- 2) **Общие службы** предоставляют специальные навыки — например, специалисты по информационной безопасности, финансовые специалисты, юристы.

4.2.3 Особенности выполнения работ на расширенном уровне

131. Типовая схема выполнения работ на расширенном уровне представлена на рисунке ниже (Рисунок 20). По аналогии с Базовым уровнем предполагается, что стадия выполнения работ выполняется серией Инкрементов, каждый из которых включает несколько итераций, реализующих:

- 1) планирование Инкремента,
- 2) выполнение Инкремента,

- 3) проверки результатов Инкремента,
- 4) корректировки выполняемых работ.

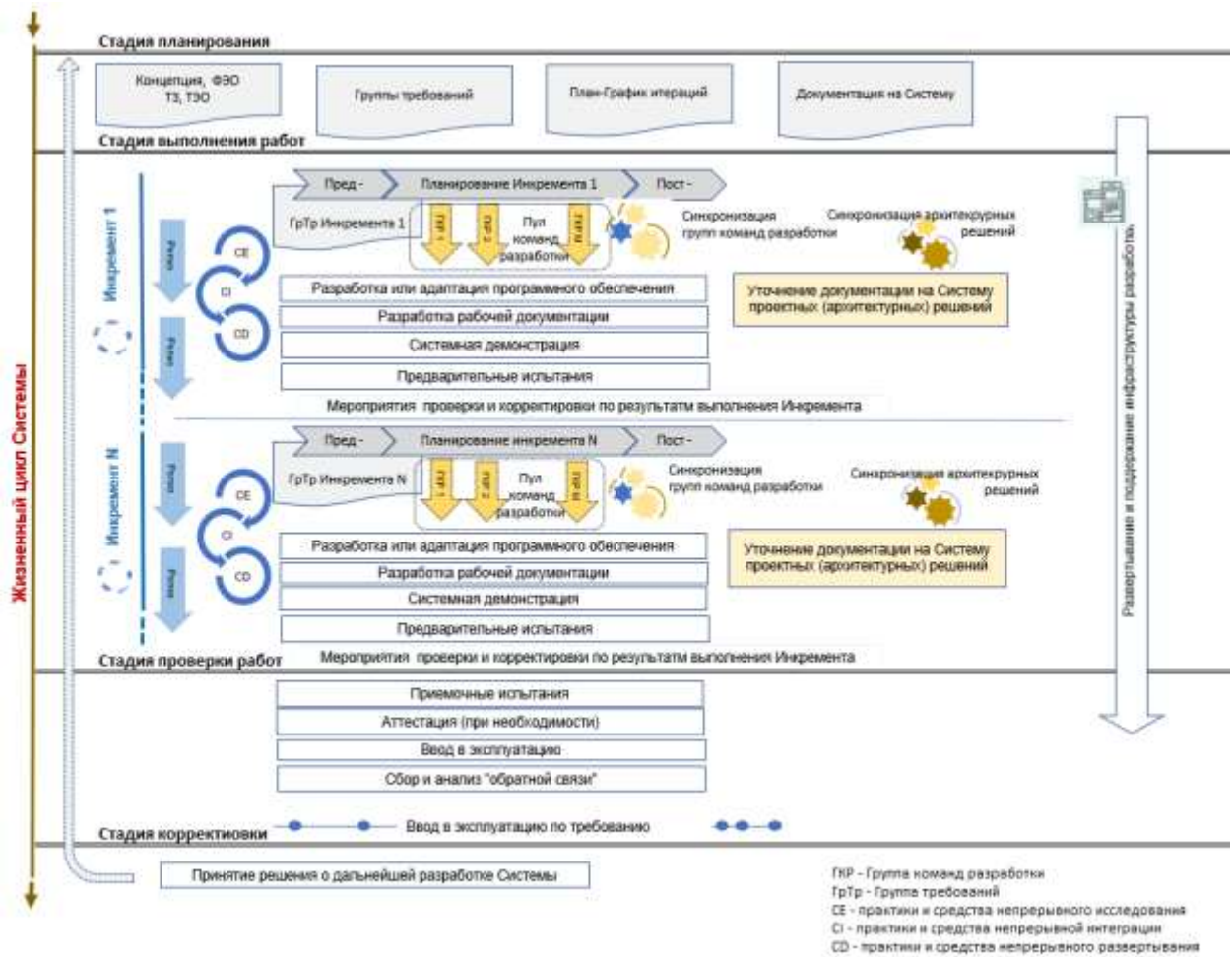


Рисунок 20. Типовая схема выполнения работ на расширенном уровне

4.2.3.1 Особенности планирование Инкремента на расширенном уровне

132. Планирование Инкремента является критической точкой синхронизации Инкрементов для каждой Группы команд разработки в рамках Пула команд. При этом в целом, для Пула команд разработки, рекомендуется два дополнительных мероприятия:

- 1) Мероприятие «**пред- планирование**» Инкремента используется для координации входных целей, этапов, а также контекста разработки решения для сессий планирования Инкремента для каждой Группы команд разработки.
- 2) Мероприятие «**пост- планирование**» Инкремент используется для интеграции результатов планирования Группы команд в рамках единого плана Инкремента. В конце мероприятия по после- планирования должно быть достигнуто соглашение о наборе целей Инкремента на уровне Системы, которые

должны быть реализованы и продемонстрированы всеми Группами команд разработки к концу Инкремента.

133. Мероприятия по пред- и пост- планирование итерации обеспечивают несколько преимуществ:

- 1) открытое и продуктивное общение и достижение компромиссов лицом к лицу;
- 2) синхронизация Групп команд разработки в рамках Группы решения и целей Инкремента Решения;
- 3) выявление зависимостей и содействие координации между Группами команд разработки;
- 4) обеспечение уточнения архитектуры на уровне Решения;
- 5) соответствие потребностей (целей) разработки Решения в рамках предстоящего Инкремента возможностям Групп команд разработки.

134. Непосредственное планирование Инкремента происходит синхронно для всех Групп команд разработки в одно и то же время аналогично тому, как данное мероприятие проводится на базовом уровне (Раздел 4.1.3.1).

135. Мероприятия пред- и пост- планирование Инкрементов обеспечивают согласованность и создают единый план для всех Групп команд разработки, а также управляют зависимостями между ними. Результатом этих мероприятий является обобщенное формирование целей Инкремента для согласования с заинтересованными сторонами на уровне Пула команд разработки.

4.2.3.2 Особенности выполнения Инкрементов на расширенном уровне

136. Реализация итерационного подхода к разработке на расширенном уровне предполагает использование единой для всех Групп команд разработки частоту Инкрементов и синхронизацию для управления разработкой. При этом следуют учитывать следующие особенности выполнения работ:

- 1) В рамках Системы, как правило, интегрируются подсистемы, созданные и предоставляемые различными Исполнителями, в условиях специфических нормативных и технологических ограничений;
- 2) При развитии Системы, объединяющих несколько подсистем, необходимо соблюдение баланса между новыми возможностями и технологиями – с одной стороны и наследуемой функциональностью, реализованной с использованием разработанных ранее архитектурных решений и технологий – с другой стороны;

3) Необходимо создание конвейера непрерывной разработки, единого для всех Групп команд разработки, а реализация процессов непрерывного и скоординированного развития функциональности должна обеспечиваться как отдельных подсистем, так и Системы в целом.

4) Создание и развитие Систем на расширенном уровне предполагает:

- регулярное уточнение проектных решений, в том числе изменение архитектуры решения для обеспечения масштабирования и удобства обслуживания Системы в будущем, а также обеспечение требований информационной безопасности;

- постоянное решение проблем, связанных с соблюдением нормативных требований (прежде всего требований информационной безопасности), касающихся разработки как отдельных подсистем, так и Системы в целом.

137. В ходе создания Системы на расширенном уровне должна обеспечиваться синхронизация работ Групп команд разработки, включающая (Рисунок 21):

1) координацию целей и задач на уровне Групп ответственных за подсистемы;

2) постоянную координацию архитектурных решений, используемых различными Группами команд разработки;

3) интеграцию подсистем, создаваемых различными Группами команд разработки.

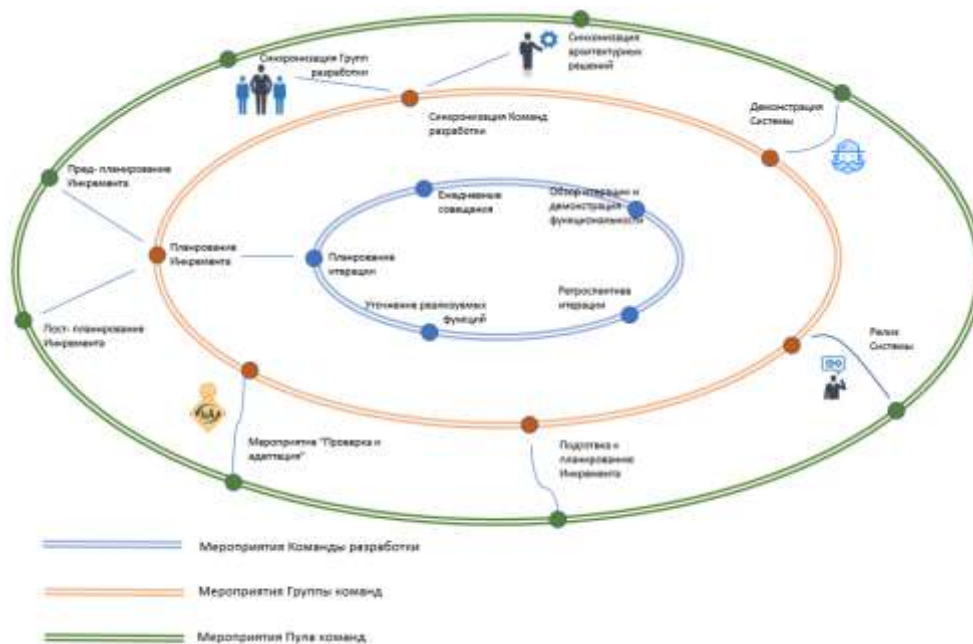


Рисунок 21. Мероприятия Пула команд по выполнению работ на расширенном уровне

4.2.3.3 Проверка результатов Инкремента на расширенном уровне

138. По аналогии с базовым уровнем на расширенном уровне основным мероприятием для изучения решений и получения объективной оценки и обратной связи является демонстрация Системы.

139. Демонстрация Системы происходит в конце каждого Инкремента. Во время демонстрации представители Пула команд разработки представляют Группе ответственных за Систему и Владельцу Системы интегрированный результат всех Групп команд разработки.

140. Каждая демонстрация Системы должна обеспечивать исправление всех выявленных ошибок и несоответствий техническому заданию, а также обеспечивать снижение рисков разработки Системы за счет получения обратной связи.

141. Пул команд разработки должен обеспечивать возможность развернуть Систему или отдельные подсистемы в контуре эксплуатации в любое время, когда это потребуется Ведомству по решению Владельцев Системы.

142. Мероприятия по вводу в эксплуатацию на расширенном уровне аналогичны мероприятиям на базовом уровне.

4.2.3.4 Корректировка выполняемых работ на расширенном уровне

143. Мероприятие «Проверка и адаптация» на расширенном уровне для Пула команд разработки осуществляется аналогично проведению проверки и адаптации в контексте одной Группы команд на базовом уровне (см. Раздел 4.1.3.4).

144. С целью сокращения временных и материальных затрат на проведение мероприятия «Проверка и адаптация» рекомендуется в состав участников мероприятий включать ключевых представителей Групп команд разработки, обладающих всеми необходимыми полномочиями для решения проблем.

145. В конце мероприятия «Проверка и адаптация» проводится ретроспективный семинар для улучшения процессов создания и развития Системы.

5. СТАДИЯ ПРОВЕРКИ ВЫПОЛНЕННЫХ РАБОТ

146. Стадия проверки выполненных работ проводится Группой ответственных за Систему с целью определения соответствия выполненных работ по созданию Системы требованиям технического задания, планирования и ввода Системы (или ее части) в эксплуатацию, мониторинга показателей функционирования Системы и сбора обратной связи от клиентов Системы.

147. Реализация функциональности Системы с использованием итерационного подхода предполагает возможность ввода в эксплуатацию (после проведения испытаний) отдельных релизов Системы, включающих функциональность по результатам реализации отдельных групп требований. Такой подход обеспечивает оперативное предоставление Системы клиентам, получение обратной связи от клиентов и формирование предложений по развитию Системы.

148. Мероприятиям по проверке предшествуют пусконаладочные работы и предварительные испытания Системы (или ее части), выполняемые на стадии «Выполнение (Раздел 4.1.3.3 настоящего документа).

149. В рамках стадии проверки проводятся следующие виды испытаний в соответствии с Постановлением № 676:

- 1) опытная эксплуатация Системы или ее части;
- 2) приемочные испытания Системы или ее части.

150. Опытная эксплуатация Системы или ее части проводится, начиная с завершения реализации первой Группы требований и до завершения разработки (адаптации) программного обеспечения, и включает:

- 1) разработку программы и методики опытной эксплуатации;
- 2) проведение опытной эксплуатации Системы в соответствии с программой и методикой опытной эксплуатации;
- 3) оформление акта о завершении опытной эксплуатации, включающего перечень недостатков, которые необходимо устранить до начала эксплуатации Системы;
- 4) проведение дополнительной итерации по доработке программного обеспечения Системы и дополнительной настройке технических средств в случае обнаружения недостатков, выявленных при опытной эксплуатации Системы.

151. Приемочные испытания Системы или ее части включают:

- 1) разработку программы и методики приемочных испытаний;

2) испытания Системы на соответствие техническому заданию на ее создание в соответствии с программой и методикой приемочных испытаний;

3) анализ результатов устранения недостатков, указанных в акте о завершении опытной эксплуатации;

4) проведение итерации устранения замечаний, выявленных в ходе приемочных испытаний.

152. По результатам проведенных испытаний и после устранения выявленных недостатков Владелец Системы принимает решение о вводе в эксплуатацию Системы (части Системы) и утверждает акт о приемке Системы в эксплуатацию.

153. Ввод Системы или ее части в эксплуатацию включает следующие мероприятия, выполняемые в соответствии с требованиями Постановления № 676:

1) подготовка правового акта Ведомства о вводе Системы в эксплуатацию, определяющего перечень мероприятий по обеспечению ввода Системы в эксплуатацию и устанавливающим срок начала эксплуатации;

2) разработка и утверждение организационно-распорядительных документов, определяющих мероприятия по защите информации в ходе эксплуатации Системы, разработка которых предусмотрена нормативными правовыми актами и методическими документами федерального органа исполнительной власти в области обеспечения безопасности и федерального органа исполнительной власти, уполномоченного в области противодействия техническим разведкам и технической защиты информации, а также национальными стандартами в области защиты информации;

3) аттестация Системы по требованиям защиты информации, в результате которых в установленных законодательством Российской Федерации случаях подтверждается соответствие защиты информации, содержащейся в Системе, требованиям, предусмотренным законодательством Российской Федерации об информации, информационных технологиях и о защите информации;

4) оформление прав на использование компонентов Системы, являющихся объектами интеллектуальной собственности;

5) ввод Системы или ее части в эксплуатацию.

154. Выполнение мероприятий, указанных в пункте 153, по вводу в эксплуатацию отдельных версий Системы (например, начиная с МДВ – раздел 3.2.2 настоящего документа) в дальнейшем обеспечит существенное сокращение времени на ввод в эксплуатацию последующих версий или Системы в целом. Например, мероприятия по повторной аттестации Системы при вводе в

эксплуатацию последующих релизов могут не проводиться в случае, если требований по защите информации и модель угроз не изменяются.

155. После ввода Системы или ее части в эксплуатацию должен быть организован мониторинг показателей качества оказания государственных услуг, государственных функций, включая контрольно-надзорную деятельность с использованием Системы (далее – мониторинг) в соответствии со Стандартом мониторинга оказания сервисов (утвержден Протоколом заседания президиума Правительственной комиссии по цифровому развитию, использованию информационных технологий для улучшения качества жизни и условий ведения предпринимательской деятельности от 29 декабря 2021 г. № 50).

156. По результатам мониторинга выявляются предложения по развитию Системы, а также замечания к функционированию Системы, устраняемые Исполнителем, например, в рамках гарантийного срока в соответствии с законодательством Российской Федерации.

157. В рамках эксплуатации Системы осуществляется сбор обратной связи от клиентов и проводится анализ потребностей клиентов с целью разработки предложений по развитию Системы.

6. СТАДИЯ КОРРЕКТИРОВКИ

158. Стадия корректировки включает:

- 1) разработку предложений по развитию Системы на основе анализа результатов мониторинга и сбора информации обратной связи от клиентов;
- 2) определение требований к развитию Системы с учетом изменения потребностей клиентов и контекста использования Системы;
- 3) оценка Ведомством объемов финансирования, необходимого для развития Системы, внесение, при необходимости, изменений в ВПЦТ;
- 4) принятие решения о дальнейшем развитии Системы или о выводе Системы из эксплуатации в соответствии с определяемыми Постановлением № 676 требованиями к порядку вывода Системы из эксплуатации и дальнейшего хранения содержащейся в ее базах данных информации;
- 5) переход к очередному циклу планирования реализации мероприятий по развитию Системы.

Приложение № 1
к Методическим рекомендациям
по организации производственного процесса разработки
государственных информационных систем
с учетом применения итерационного подхода к разработке

МЕТОДЫ И ПРАКТИКИ, ИСПОЛЬЗУЕМЫЕ ДЛЯ СОЗДАНИЯ И РАЗВИТИЯ ГОСУДАРСТВЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

1. ПЕРЕЧЕНЬ ОСНОВНЫХ МЕТОДОВ⁸ И ПРАКТИК⁹

Методические рекомендации по разработке государственных информационных используют следующие методы и практики (Рисунок 22), получившие широкое распространение для создания и развития цифровых сервисов и продуктов¹⁰:

- 5) Системный подход;
- 6) Методы итеративной разработки;
- 7) Методики масштабирования разработки;
- 8) Практики DevSecOps;
- 9) Встроенное качество;
- 10) Дизайн-мышление;
- 11) Конвейер непрерывной доставки.

8 Метод — систематизированная совокупность шагов, действий, которые необходимо предпринять, чтобы решить определенную задачу или достичь определенной цели (ГОСТ 56407 – 2015. Бережливое производство. Основные положения)

9 Практика – накопленный опыт, испытанные приёмы и навыки в какой-либо области деятельности

10 Продукт - изделие и/или услуга, которые в настоящее время и в будущем будут предлагаться производителем и обладают определенной ценностью для существующих или потенциальных клиентов (ГОСТ Р 58537- 2019. Управление продукцией. Основные положения)



Рисунок 22. Основные методы и практики создания продукта

Теоретическим обоснованием используемых методов и практик является Системный подход – методология изучения (рассмотрения) системы как целостного комплекса взаимосвязанных и взаимодействующих элементов для достижения поставленных целей.

В рамках настоящих Методических рекомендаций в рамках системного подхода использованы следующие направления:

12) **Адаптивность к изменениям**, предполагающая использование механизмов разработки систем, позволяющих достигать целей в условиях изменения контекста использования системы и потребностей клиентов, для которых создается система (является теоретическим обоснованием итерационного (гибкого) подхода к разработке).

13) **Обратная связь**, предполагающая в ходе создания системы получение обратной связи о степени достижения цели разработки для обеспечения устойчивости производственного процесса (является теоретическим обоснованием необходимости корректировки требований в процессе разработки).

14) **Интеграция элементов и обеспечение целостности**, предполагающие учет принципов системности и эмерджентности при разработке систем, то есть представление системы, состоящей из нескольких подсистем как единого целого, с одной стороны, и как подсистему для вышестоящих уровней иерархии, а также свойства системы обладать признаками, отличными от признаков подсистем ее составляющих (является теоретическим обоснованием механизмов масштабирования команд разработки и подходом к формированию моделей требований к системе).

15) **Координация и централизация** в многоуровневых системах, предполагающие организацию управления в сложных иерархических системах через гибкое сочетание принципов автономности нижестоящих элементов иерархии и централизованного формирования координирующих механизмов для обеспечения стабильности производственных процессов (является теоретическим обоснованием практик организации взаимодействия и координации крупномасштабных команд разработки).

2. МЕТОДЫ ИТЕРАЦИОННОЙ РАЗРАБОТКИ

Получившая широкое распространение и применяемая в настоящее время классическая водопадная модель создания систем основана на последовательном выполнении этапов проекта от планирования до сдачи Системы в эксплуатацию. При этом потребные привлекаемые ресурсы (бюджет) и сроки выполнения этапов проекта определяются в начале проекта исходя из оценки реализации заданных требований. Ошибки в оценках бюджета и сроков порождают риски невыполнения проекта в рассчитанных рамках, либо снижение качества выполненных работ.

Альтернативой классическому подходу стали методы итерационной разработки. Первый ставший популярным гибкий метод - это экстремальное программирование, основы которого подробно изложены в книге «Extreme Programming Explained» Кента Бека (2000 год). Следующим гибким методом, получившим распространение, стал Скрам, описанный в книге 2001 года «Agile Software Development with SCRUM» Кена Швабера и Майка Бидла.

Конечным результатом применения методов итерационной разработки является устранение рисков низкого качества разработки. В итерационной разработке фиксируются сроки и ресурсы (команда), а изменяются состав требований, которые должны быть реализованы итерационно для последовательного достижения целей разработки (Рисунок 23).



Рисунок 23. Смена парадигмы разработки

В конце каждой итерации должен достигаться конечный результат — версия цифрового продукта, в рамках которой реализован запланированный набор функций. Конечный результат предоставляется заинтересованным сторонам, возможно, конечным пользователям для получения обратной связи и

принятия решения о завершении разработки или определения направлений развития продукта.

Настоящие Методические рекомендации предполагают, что Команды разработки используют один из методов итерационной разработки – Скрам, Экстремальное программирование, систему Канбан и/или их комбинации. В частности, большинству Команд разработки рекомендуется использовать Скрам в силу простоты и универсальности этого метода. Для других команд, таких, как например, Системные команды предпочтительно применять систему Канбан в качестве основного метода в силу быстро меняющихся приоритетов их работы и отсутствия необходимости регулярного планирования итераций.

2.1 Методология Скрам

Методология¹¹ Скрам — это способ организации рабочего процесса разработки продукта, при котором задачи разработки разбиваются на мелкие блоки (подзадачи) и передаются их кросс-функциональным автономным Командам разработки, которые работают методом итераций, то есть короткими непрерывными циклами, в течение которых команда определяет, строит, тестирует и анализирует результаты реализации функциональности продукта. В Скрам для обозначения итерации используется термин «спринт».

Классический Скрам состоит из ролей, артефактов и процессов.

Роли Скрам:

- б) кросс-функциональная команда,
- 7) владелец продукта (ответственный за реализацию),
- 8) менеджер процессов команды (скрам-мастер).

Основные артефакты Скрам:

- 9) бэклог продукта – источник требований и их изменений на протяжении всего проекта,
- 10) бэклог спринта,
- 11) добавленная ценность продукта.

Рабочий процесс в Скрам проходит по определенной схеме, которая определяется **серией ограниченных по времени событий**, которые всегда происходят в одинаковом порядке (Рисунок 24) и включают:

- 12) планирование спринта;

¹¹ Методология - набор взаимосвязанных **методов и методик** и их практического применения (практик), оптимизированных для решения конкретных задач

- 13) ежедневные совещания на ногах (или «Дэйли скрам»);
- 14) подведение итогов спринта;
- 15) обзор итогов спринта, включая демонстрацию версии;
- 16) ретроспектива спринта;
- 17) уточнение бэклога продукта.

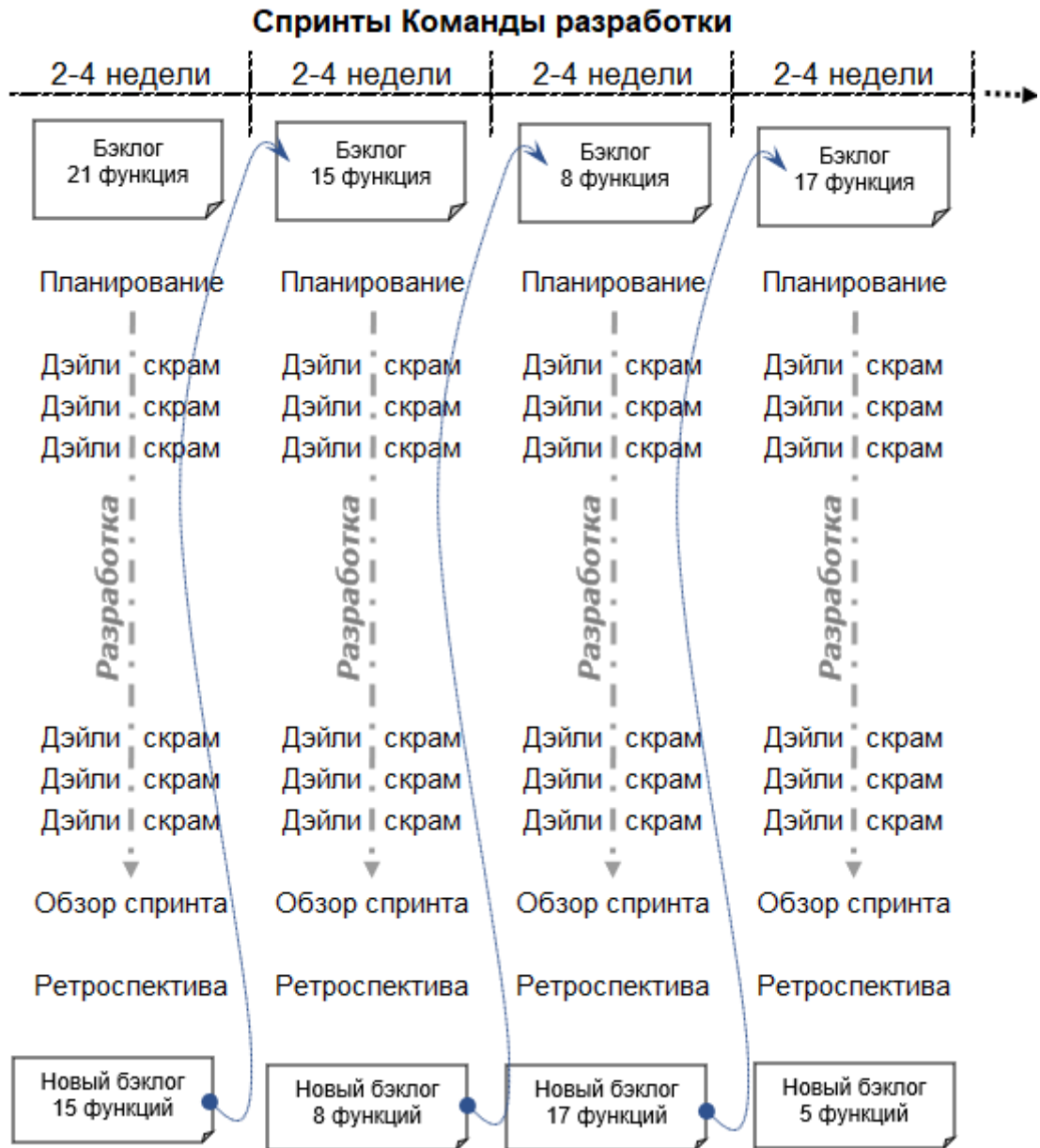


Рисунок 24. Артефакты и процессы Скрам

2.1.1 Планирование итерации (спринта)

Итерация начинается с планирования - события продолжительностью до восьми часов, в котором владелец продукта формулирует **цель спринта** (одно-

два предложения, описывающих, что должно быть сделано в ходе спринта) и представляет требования для планирования в виде **пользовательских историй**¹². Затем команда для уяснения работы, которая будет выполнена в предстоящей итерации, выполняет следующие действия:

18) проводит обсуждения, а при необходимости уточнение пользовательских историй,

19) разбивает пользовательские истории спринта на **задачи**¹³, занимающие один день или менее на разработку (это называется **декомпозицией**),

20) определяет для каждой задачи ответственного исполнителя из состава команды,

21) переводит наборы функций, которые команда может предоставить в предстоящей итерации, основываясь на их известной скорости, в бэклог спринта, в котором перечисляется все, что будет реализовано командой за время спринта,

22) фиксирует план спринта, включающий, том числе - цели спринта и пользовательские истории.

Планирование пользовательских историй рекомендуется выполнять с использованием подхода «вертикальных срезов», предполагающего в ходе разработки внесение изменений в каждый архитектурный слой продукта. При этом реализация **функциональных требований** дополняется реализацией обеспечивающих **нефункциональных требований** (Рисунок 25).

12 Пользовательские истории (англ. User Story) — способ описания требований к разрабатываемому продукту, сформулированных как одно или более предложений на повседневном или деловом языке пользователя

13 Многие команды делят истории на задачи, оценивая их в часах, чтобы лучше улучшить свое понимание предстоящей работы



Рисунок 25. Использование вертикальных срезов включает работу во всех архитектурных слоях для реализации функций

2.1.2 Пользовательские истории

Пользовательская история — это наименьшая единица функциональности в методиках гибкой разработки. Пользовательская история представляет собой описание функций простыми, общими словами, без технической специфики, составленное с точки зрения конечного пользователя. Она пишется с целью разъяснить, как именно функциональная возможность принесет ценность пользователю. В Скрам пользовательские истории добавляют в спринты и отслеживают на диаграммах сгорания в течение спринта.

2.1.3 Визуализация работы

Визуализация работы обеспечивает **прозрачность** создания продукта (системы). Наиболее употребимыми средствами визуализации работы в рамках методологии Скрам являются информационные доски и диаграммы сгорания.

2.1.3.1 Информационные доски

Команды используют «Информационные доски» для понимания и отслеживания прогресса во время выполнения итерации. Информационная доска представляет распределение специалистов команды по задачам, визуализирует реализуемые функции и прогресс их реализации на протяжении всей итерации, как показано на рисунке ниже (Рисунок 26).

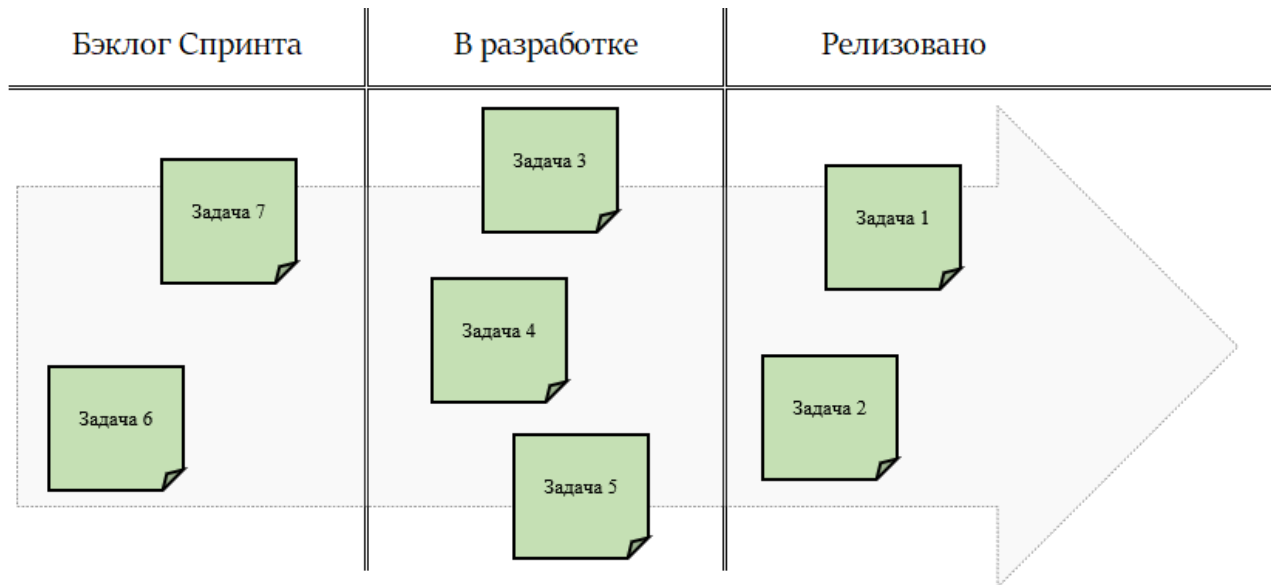


Рисунок 26. Пример информационной доски команды

Во время выполнения работы команда создает и тестирует одну - две функции каждые несколько дней. При этом рекомендуется следить за количеством задач, находящихся одновременно в разработке или тестировании, чтобы постоянно балансировать объем работы с их доступными ресурсами команды и увеличить ее пропускную способность.

Многие команды интегрируют лучшие практики Скрам и Канбан, чтобы облегчить поток работы через итерации. В этом случае распределение задач превращается в более структурированную доску Канбан.

2.1.3.2 Диаграммы сгорания

Диаграмма сгорания показывает количество оставшейся работы в начале каждой итерации. Она служит эффективным визуальным индикатором скорости продвижения команды к ее цели. Пример диаграммы сгорания представлен на рисунке ниже (Рисунок 27).

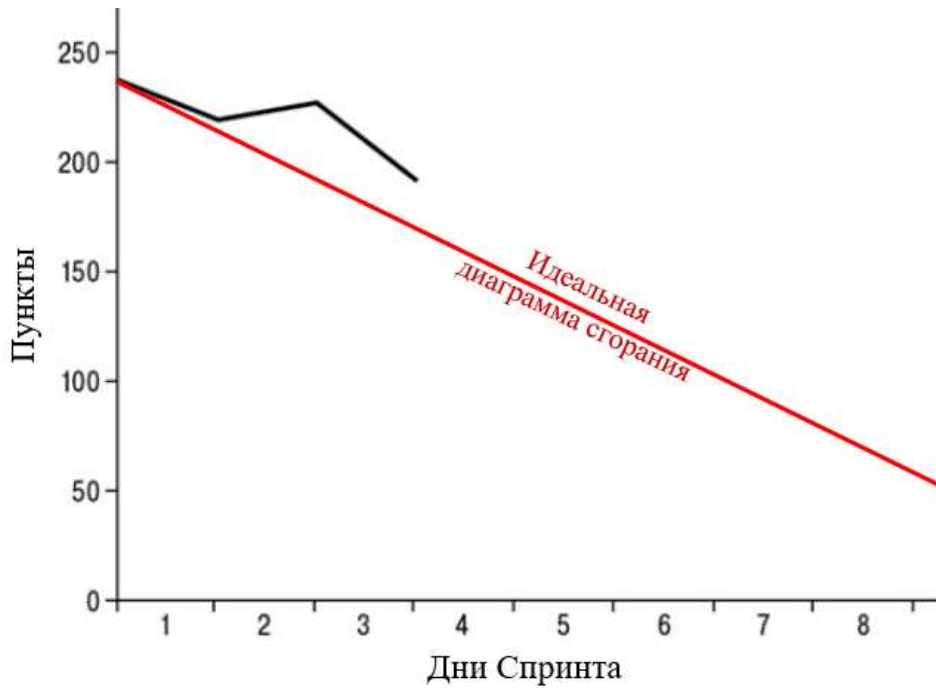


Рисунок 27. Диаграмма сгорания

На вертикальной оси откладывается количество оставшихся пунктов в Бэклоге Спринта. На горизонтальной оси откладываются дни Спринта. Данные диаграммы необходимо ежедневно обновлять, чтобы в реальном времени показывать прогресс и издержки в работе над Спринтом.

Рекомендуется, чтобы оставшаяся работа уменьшалась более-менее равномерно по ходу Спринта (см. идеальную прямую на рисунке). Это ведет к уменьшению числа параллельных задач, стимулирует команду помогать друг другу ради скорейшего перехода задач в «Готово» и в итоге ведет к повышению эффективности Команды.

При таком представлении одна диаграмма просто и ясно показывает два самых важных показателя, которые можно использовать для отслеживания проекта: объем оставшейся работы и чистый темп продвижения команды, не зависящий от изменений объема Бэклога.

2.1.4 Ежедневные совещания на ногах

Каждый день команда проводит официальное мероприятие — ежедневное совещание на ногах («Дэйли скрам») — чтобы понять текущий статус работ, проанализировать проблемы и получить помощь от других членов команды. Во время этого мероприятия каждый член команды описывает, что он сделал вчера для достижения целей Спринта, над чем он собирается работать сегодня и любые блокирующие обстоятельства (проблемы), с которыми он сталкивается при достижении целей итерации.

За организацию и проведения ежедневного совещания на ногах отвечает Ответственный за процессы команды (Скрам-мастер). Он обеспечивает его регламент и полноту. Совещание на ногах должно занимать не более 15 минут и завершается распределением текущих задач.

После проведения совещания на ногах командное общение не заканчивается, так как члены команды взаимодействуют непрерывно на протяжении всей итерации. В интересах облегчения такого общения необходимо, чтобы команда была размещена в одном месте, если это возможно.

2.1.5 Обзор итогов Спринта

В конце каждого Спринта команда проводит обзор итерации и ретроспективу во время которых команда демонстрирует каждую реализованную функцию (выполненную историю), обращая особое внимание на увеличение ценности для пользователей. Обзор итерации не является официальным докладом о положении дел, скорее, это обзор осязаемых результатов итерации. Основная задача проведения обзора Спринта заключается в получении обратной связи от заинтересованных сторон по результатам разработки запланированного Бэклога Спринта проведенной итерации (Рисунок 28).

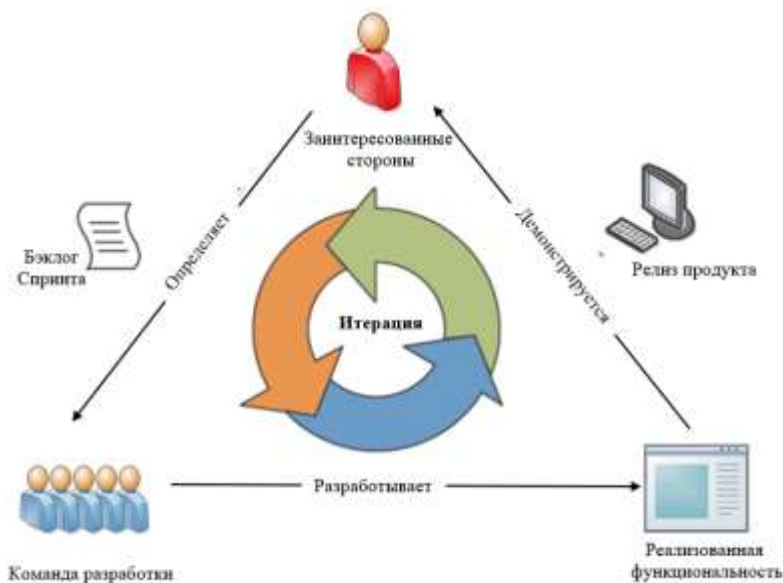


Рисунок 28. Получение обратной связи в итерации

2.1.6 Проведение ретроспективы Спринта

После обзора команда проводит короткую. Ретроспективу традиционно проводят после обзора спринта спустя небольшое количество времени, чтобы получить обратную связь после демонстрации.

Организует ретроспективу Ответственный за процессы команды (скрам-мастер) с привлечением Владельца продукта.

Обычно ретроспектива занимает от 30 минут до 3 часов (в зависимости от продолжительности Спринта). Традиционным на первом этапе ретроспективы проводится сбор данных, который заключается в ответах каждого из членов команды на вопросы:

- 16) Что было сделано хорошо?
- 17) Что можно улучшить?
- 18) Какие улучшения будем делать?

При этом количество улучшений, которые команда берет в реализацию, не должно превышать 2-3, и команда должна обязательно в том или ином виде составить план улучшений для контроля их исполнения.

На втором этапе желательно обсудить следующие темы:

- 23) скорость команды и ее изменение по сравнению с предыдущими спринтами;
- 24) качество работ, в том числе - нереализованные истории пользователей и причины опоздания, дефекты и их причины;
- 25) качество процессов команды (нарушения, отступления).

2.1.7 Уточнение Бэклога продукта

Уточнение Бэклога Продукта проводится не для пересмотра того, что уже взято в работу в текущем Спринте, а для будущих Спринтов. Хорошей практикой считается иметь в Бэклоге продукта детально проработанные элементы как минимум на два Спринта вперед. В этом случае Планирование Спринта существенно упрощается, поскольку Владелец продукта и Команда начинают планирование с понятным, прошедшим этап анализа и корректно оцененным набором пользовательских историй. Если же уточнение Бэклога не было проведено (или было проведено недостаточно хорошо), Планирование Спринта может занять много времени, вызвать большое количество вопросов и потребовать неоднократных уточнений в ходе планирования.

2.2 Практика экстремального программирования

Экстремальное программирование (англ. eXtreme Programming, XP) — одна из гибких методологий разработки программного обеспечения. Авторы методологии — Кент Бек, Уорд Каннингем, Мартин Фаулер и другие.

Ключевые особенности XP включают следующее.

19) Команда из 7.12 программистов работает в одном месте с представителем заказчика.

20) Разработка происходит в виде частых сборок или итераций, обеспечивающих дополнительную функциональность, которые могут быть или не быть выпущенными.

21) Требования задаются в виде пользовательских историй, каждая из которых представляет собой фрагмент новой функциональности, необходимой пользователю.

22) Программисты работают в парах, следуют строгим стандартам кодирования и проводят собственное модульное тестирование, что обеспечивает высокое качество кода.

23) Представитель Заказчика участвует в приемочных испытаниях.

24) Требования, архитектура и проектные спецификации появляются и изменяются в ходе проекта.

Двенадцать основных приёмов экстремального программирования могут быть объединены в четыре группы:

25) Короткий цикл обратной связи (Fine-scale feedback);

- Разработка через тестирование (Test-driven development);
- Игра в планирование (Planning game);
- Заказчик всегда рядом (Onsite customer);
- Парное программирование (Pair programming);

26) Непрерывный, а не пакетный процесс;

- Непрерывная интеграция (Continuous integration);
- Рефакторинг (Design improvement, Refactoring);
- Частые небольшие релизы (Small releases);

27) Понимание, разделяемое всеми;

- Простота проектирования (Simple design);
- Метафора системы (аналог архитектуры) - даёт команде представление о том, каким образом система работает в настоящее время, в каких местах добавляются новые компоненты, и как они должны взаимодействовать с существующими компонентами;

– Коллективное владение кодом (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership);

- Стандарт оформления кода (Coding standard or Coding conventions);

28) Социальная защищённость программиста (Programmer welfare);

- 40-часовая рабочая неделя (Forty-hour week).

2.3 Метод визуализации и управления работой Канбан

Система Канбан¹⁴ является методом визуализации и управления работой. Методика предполагает обсуждение командами производительности в режиме реального времени и полную прозрачность рабочих процессов. Канбан включают 3 базовых правила:

29) Визуализируйте производство:

- разделите работу на задачи, каждую задачу напишите на карточке и поместите на стену или доску,
- используйте названные столбцы, чтобы показать положение задачи в производстве.

30) Ограничивайте работу, выполняемую одновременно на каждом этапе производства.

31) Измеряйте среднее время на выполнение одной задачи и оптимизируйте постоянно процесс, чтобы уменьшить это время.

Рабочие задачи визуально представляются на доске Канбан, что позволяет участникам команды видеть состояние каждой задачи в любой момент времени. Доска Канбан должна стать достоверным источником информации о работе. Доски нужны, чтобы визуализировать работу команды, стандартизировать процесс, а также находить и устранять блокирующие элементы и зависимости.

На стандартной Канбан-доске процесс может состоять из трех шагов: «Запланировано», «В работе» и «Сделано». Однако доску можно настроить в

14 Компания «Тойота» применила эту систему в своих цехах, чтобы лучше соотнести внушительные складские запасы и реально используемые в производстве материалы. Для отслеживания объемов производства в цехе (и взаимодействия с поставщиками) в режиме реального времени использовалась специальная карточка, или Канбан, которую рабочие передавали между командами. Когда в корзине заканчивались используемые на участке производства материалы, на склад передавали Канбан с указанием необходимого материала, нужного количества и т. д. На складе уже стояла новая корзина с этим материалом: ее отправляли в цех, а складские работники отсылали поставщику свой Канбан. У поставщика корзина с этим материалом тоже была готова, и он отправлял ее на склад. Конечно, в современном мире сообщения передаются совсем не так, как в сороковых, но смысл остается тем же — все основано на процессе «своевременного» производства (JIT).

соответствии с процессом, принятым в той или иной команде, в зависимости от ее размеров, структуры и целей.

На рисунке ниже показан пример доски Канбан команды, которая определяет четыре шага реализации функциональности (Пользовательских историй): Анализ, Разработка, Интеграция и Тестирование, а также для шага Разработка использованы буферы «В работе» и «Готово», чтобы упростить управление изменениями потока работ.



Рисунок Ъ. Пример Канбан - доски

Каждая рабочая задача на доске представлена в виде отдельной карточки, на которой отображается важная информация о конкретной рабочей задаче: имя ответственного за выполнение задачи, краткое описание выполненной работы, оценка необходимого времени и т.д. Когда все члены команды видят состояние каждой рабочей задачи в любой момент времени, а также всю связанную с ней информацию, повышается концентрация членов команды, обеспечивается полная прозрачность процесса разработки.

2.3.1 Гибкость планирования

Канбан позволяет командам концентрироваться только на текущей работе. По завершении рабочей задачи команда забирает следующую задачу с верха бэклога. Владелец продукта (Ответственный за реализацию) может менять приоритет задач в бэклоге, не мешая работе команды, поскольку изменения происходят за пределами текущих рабочих задач. Если Менеджер процессов Команды разработки следит за тем, чтобы наверху бэклога были самые важные задачи, Команда разработки будет гарантированно реализовывать самую важную функциональность в первую очередь. При этом следует учитывать, что необходимости в спринтах фиксированной длительности, используемых в методике Скрам, нет. Менеджеру процессов Команды разработки рекомендуется привлекать команду разработчиков к изменениям в бэклоге.

2.3.2 Сокращение продолжительности цикла

Продолжительность цикла - ключевой показатель для Команд разработки использующих Канбан. Под продолжительностью цикла понимается время прохождения рабочей задачи цикла, от начала работы над задачей до завершения ее реализации. Оптимизировав продолжительность цикла, в будущем команда сможет с уверенностью предсказывать срок поставки задач.

Если теми или иными навыками обладает несколько человек, продолжительность цикла сокращается, если же только один — в рабочем процессе появляется «узкое» место. Благодаря обмену знаниями участники команды могут выполнять разнообразные задачи, что еще больше оптимизирует время цикла. Это также означает, что в случае возникновения узкого места в работе вся команда сможет взяться за него и восстановить нормальное течение процесса.

2.3.3 Непрерывная доставка функциональности.

Канбан и практика непрерывной поставки функциональности (раздел 5) идеально дополняют друг друга, поскольку обе методики основаны на своевременной (и последовательной) поставке ценности для пользователей. Поскольку Команды разработки, использующие Канбан, сконцентрированы на оптимизации процесса поставки продукта клиентам, чем быстрее команда сможет подготовить релиз продукта, тем более конкурентоспособным будет этот продукт.

2.4 Масштабирование кросс-функциональных команд

Хотя команды являются кросс-функциональными, не всегда реально для команды от 7 до 12 человек обеспечить ценность для конечного пользователя, когда решение включает в себя различные технологические платформы и спектр дисциплин, таких как аппаратное обеспечение, программное обеспечение и системная инженерия. Как правило, для реализации Системы требуется гораздо больше команд, которые должны работать согласованно. Чтобы решить эту проблему, кросс-функциональные команды предлагается объединять в Группы команд разработки (Рисунок 29).

Внутри пула обеспечивается согласование решаемых задач и среда совместной работы, основанная на конвейере непрерывной разработки (раздел 5), в которой команды могут сотрудничать с другими командами. В рамках Группы команд все команды **совместно** проводят все мероприятия производственного цикла, как показано на рисунке ниже, что позволяет избежать фокусировки Команд разработки исключительно на локальных проблемах

отдельной команды. Такое согласование позволяет кросс-функциональным командам **независимо** и **скоординировано** выполнять разработку функциональности Системы.



Рисунок 29. Масштабирование кросс-функциональных команд

3. ПРАКТИКИ DEVSECOPS

3.1 Общие положения DevSecOps

Разработке и развитию информационных систем присуще противоречие: процессы создания информационных технологий зависят от команд с противоположными целями и стимулами. Команды разработки должны быстро вносить изменения в том числе по результатам оценки удовлетворенности конечных пользователей текущей версией Системы, а операционные группы (техническая поддержка) должны регулировать поток изменений, чтобы поддерживать стабильность уже созданной функциональности. Кроме этого, специалисты по информационной безопасности должны внедрить политики, предотвращающие внесение изменений в уязвимости, которые могут привести к утечке данных (Рисунок 30).

Для устранения указанного противоречия широкое распространение получил набор технических практик DevOps¹⁵, обеспечивающий связи, интеграцию, автоматизацию и тесное сотрудничество в Командах разработки, необходимыми для планирования, разработки, тестирования, развертывания, выпуска и эксплуатации Системы.



Рисунок 30. Схема DevSecOps

¹⁵ DevOps — акроним, состоящий из двух слов: Development (Разработка) и Operations (Эксплуатация).

DevOps - содержит набор ролей, инструментов и рекомендаций, а также «философия», которая состоит из следующих установок:

- коммуникация между командами разработки и технической поддержки должна быть прозрачной и эффективной;
- фокус — на продукты, а не проекты;
- большая часть этапов работы должна быть автоматизирована;
- работа над продуктом не заканчивается после релиза (выпуска версии);
- создание продукта считается завершенным только тогда, когда он снят с производства.

DevOps предполагает разделение полномочий между участниками процесса при высокой степени автоматизации процессов сборки, тестирования и развертывания. Применение данного подхода позволяет создавать масштабируемые программные продукты, наращивая по мере увеличения числа программных модулей количество команд и перераспределяя их полномочия, а также делегируя задачи производства и контроля качества сравнительно небольшим командам специалистов.

Важным конкурентным преимуществом DevOps является включение специалистов технической поддержки в команды разработки и возможности устранять ошибки и расширять функциональность программных систем в режиме непрерывного развития (до вывода из эксплуатации). С этой целью используется конвейер непрерывной разработки, включающий такие инструменты как CI (Continuous Integration – постоянная интеграция, оперативное расширение функциональности) и CD (Continuous Delivery – постоянная поставка, выпуск релиза), а также обобщенный инструмент CI/CD. Их дополняет Continuous Exploration (непрерывное исследование того, что нужно создать или изменить в Системе, в том числе, через обратную связь с конечным пользователем). Согласованное использование указанных практик образуют конвейер непрерывной разработки Системы, как базовый рабочий инструмент создания и развития Системы (см. Раздел 5).

Изначально DevOps не включал безопасность в качестве первоочередной задачи явно, как это было в отношении разработки и эксплуатации. Чтобы избежать риска того, что безопасность останется второстепенной, появился термин DevSecOps, который подчеркивает важность надежных методов информационной безопасности непосредственно в процессе создания (развития) Системы. Исследовательские проекты посвященные DevOps, показывают, что уровень безопасности организации улучшается, когда безопасность полностью интегрирована в инструменты разработки цифровых продуктов и сервисов.

3.2 Требования нормативно-правовых актов к безопасности

При использовании практик DevSecOps необходимо учитывать следующие актуальные нормативные документы, определяющие требования к обеспечению безопасности государства в сфере, связанной с созданием, хранением, передачей и преобразованием информации:

32) Федеральный закон от 26 июля 2017 г. № 187-ФЗ "О безопасности критической информационной инфраструктуры Российской Федерации";

33) Приказ ФСТЭК России от 25 декабря 2017 г. N 239 "Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации" (в ред. приказа ФСТЭК России от 26 марта 2019 г. № 60).

34) ГОСТ Р 57628–2017 "Информационная технология (ИТ).

35) ГОСТ Р 57628-2017, ГОСТ серии 15408, в части разработки профилей защиты и критериев оценки безопасности ИТ

36) ГОСТ Р 51583-2014 в части создания АСЗИ, если для определяется законодательством Российской Федерации или решением ее обладателя.

37) НПА в части обеспечения безопасности обработки информации, содержащих Государственную тайну)

38) Приказ ФСТЭК 77 от 29.4.2021, определяющего порядок аттестации ГИС.

39) ГОСТ Р 56939, ГОСТ Р 58412 в части безопасной разработки ПО

40) Федеральный закон от 27 июля 2006 года N 152-ФЗ "О персональных данных" (с изменениями от 2.07.2021)

Виды нарушений, касающиеся самой информации и средств ее обработки, представлены в следующих статьях УК РФ: 1 ст. 272. "Неправомерный доступ к компьютерной информации"; 1 ст. 273. "Создание, использование и распространение вредоносных компьютерных программ"; 1 ст. 274. "Нарушение правил эксплуатации средств хранения, обработки или передачи компьютерной информации и информационно-телекоммуникационных сетей"; 1 ст. 274.1. "Неправомерное воздействие на критическую информационную инфраструктуру Российской Федерации".

Также в Кодексе об административных правонарушениях (КоАП РФ) предусмотрен перечень общественно опасных деяний, связанных с разглашением либо сокрытием информации.

3.3 Документирование итеративной разработки

3.3.1 Проблемы поддержания документации в актуальном состоянии при итеративной разработке

При всех недостатках каскадной модели разработки, основанной на ГОСТ 19 и 34 серий и их аналогов, грамотно выстроенный процесс разработки позволяет порождать необходимую документацию в процессе перехода от этапа к этапу. Причем для специалиста, который только начинает знакомиться с системой, сложность возрастает линейно по мере погружения в предмет исследования – от общих проблем к более частным.

В рамках создания и развития Систем с использованием итерационного подхода происходит регулярное и частое обновление функциональности Системы. При этом изменения функциональности, видимые конечному пользователю, должны быть синхронизированы с внесением изменений в документацию. К примеру, конечный пользователь (клиент) может попросту не заметить новую функциональность, если комплект поставки не будет включать подробного описания новой функциональности и сценариев ее использования.

Применение рекомендуемых средств DevSecOps для автоматизированного тестирования и развертывания, позволяет с достаточной скоростью и степенью точности фиксировать, а затем многократно тиражировать состояние программно-аппаратного комплекса Системы, что в теории должно давать 100%-ную гарантию работоспособности продукта.

Поскольку DevOps не содержит формальных и жестких требований к оформлению документации, это приводит к тому, что специалисты, участвующие в создании Системы могут использовать несколько ссылок на репозитории проекта, базы знаний разработчиков, тестировщиков, менеджеров по качеству, техподдержки и маркетинга, а также трекеры задач и ошибок, что представляет собой существенные объемы плохо структурированного текста.

При первом приближении кажется, что достижение максимальной полноты информации о производственном процессе должно способствовать разностороннему пониманию технологического процесса, однако на деле для консолидации специалистов в схожих по сути, но в то же время различных по подходам предметных областях (например, тестирование) требует привлечения экспертов по формализации знаний, что существенно повышает стоимость разработки.

3.3.2 Предложения по разработке документации в рамках итеративной разработки

Обеспечить возможность внесения согласованных изменений в документацию можно с помощью включения этих работ в общий процесс итеративной разработки, начиная от планирования завершая демонстрацией реализованной функциональности, испытаниями и выпуском релиза. Это обеспечит согласованное инкрементальное наращивание и функциональности Системы и документации на нее. При этом, на этапе организации Команд разработки целесообразно либо включить соответствующих специалистов (технических писателей) в состав Команды разработки либо сформировать специальную команду обеспечения по разработке документации.

При планировании следует включать в состав работ по реализации функциональности, помимо разработки и/или адаптации программного обеспечения, тестирования, также и документирование реализованной функциональности. Таким образом, функциональность считается готовой только тогда, когда она в том числе описана в пользовательской документации. В итоге, после завершения реализации функциональности Система будет готова к испытаниям и вводу в эксплуатацию.

В ходе выполнения работ целесообразно использование традиционных решений, упрощающим следование гибким процессам, является визуализация работ с помощью доски Канбан (раздел 2.3). При этом задачи по документированию должны отображаться на доске наряду с задачами по разработке и тестированию.

В интересах установления связи между реализуемыми требованиями и их документированием рекомендуется декомпонировать документацию на отдельные детали, каждая из которых представляет атомарную автономную документацию конкретного реализованного требования. Тогда станет возможной двусторонняя навигация: от конкретной пользовательской истории к ее описанию в пользовательской документации и обратно от конкретной части документации к тому требованию, которое было реализовано и породило данное описание. В итоге, сценарий обновления документации превращается в механическую работу: найти и добавить в пользовательскую документацию описания требований, реализованных за время работы над последним выпуском, обновить в пользовательской документации описания измененных требований, удалить описания исключенных из выпуска функций.

Одна из проблем, указанных выше, заключается во внесении требуемых изменений при сборке пользовательской документации. Решение проблемы может заключаться в использовании средств автоматизации, обеспечивающих

синхронизацию отдельных спецификаций (например, описания пользовательских историй) из репозитория Команд разработки и финальных документов, определенных Техническим заданием.

4. БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В 1990 году в книге *The Machine That Changed the World*¹² («Машина, изменившая мир») было предложено новое имя для того, что до этого было известно как поставки «точно вовремя», или производственная модель Toyota. С этого момента принцип, на котором построено изготовление автомобилей в этой компании, стал называться бережливым производством.

Бережливое производство (от англ. *lean production*) — концепция управления производственным предприятием, которая основана на постоянном стремлении предприятия к устранению всех видов потерь. Бережливое производство предполагает вовлечение в процесс оптимизации каждого сотрудника и максимальную ориентацию на потребителя.

Бережливый подход к созданию программного обеспечения рассматривает гибкие методы (см. раздел 2) как методы, доказавшие свою эффективность. Однако данный подход предоставляет более широкую перспективу, позволяющую гибким методам быстро развиваться.

Во-первых, бережливый подход позволяет держать под контролем всю цепочку создания ценности (от концепции до реализации), а также выявлять все случаи потерь и задержек, имеющих место до и после разработки кода.

Во-вторых, он создает управленческую среду, делающую возможным развитие гибких подходов к разработке программного обеспечения.

В-третьих, бережливый подход предоставляет набор доказавших свою эффективность принципов, которые каждая организация может использовать для внедрения предлагаемых средств, приемов и методов в своей уникальной среде и с учетом своих возможностей.

Бережливая разработка программного обеспечения, как методология разработки программного обеспечения, использующая методы концепции бережливого производства, основан на следующих принципах.

Исключение потерь. Потерями считается всё, что не добавляет ценности для потребителя. В частности: излишняя функциональность; ожидание (паузы) в процессе разработки; нечёткие требования; бюрократизация; медленное внутреннее сообщение.

Встроенное качество. Принцип предполагает встраивание качества в программный код, а не тестирование этого кода после его создания. Сегодня существуют средства, позволяющие устранять дефекты из программного кода в ходе его создания. Группа разработчиков, используя разработку через тестирование (*test driven development - TDD*), создает блочные тесты и приемочные тесты до создания соответствующего кода. Разработчики

интегрируют код и тесты в систему так часто, как это возможно (каждый час или около того) и инициируют запуск тестовой нагрузки, чтобы убедиться, что со времени предыдущей проверки дефекты в коде не появились. Если тест не выполняется, они не создают новый код, пока проблема не будет устранена. В конце дня запускается более сложная и полная тестовая нагрузка. В конце недели система проверяется с помощью еще более сложной тестовой нагрузки. В организациях, где используется такой подход, дефекты возникают редко и их причины устраняются очень быстро.

Целью процесса тестирования (а также Команд разработки, разрабатывающих тесты и проводящих тестирование) должно быть предотвращение дефектов, а не поиск их. Разработка должна быть ориентирована на удаление дефектов из программного кода в процессе его создания (если угодно, это можно назвать встраиванием качества в код), а не на проверку качества кода позже.

Акцент на обучении. Принцип предполагает короткие циклы разработки, раннее тестирование, частая обратная связь с заказчиком.

Разработка программного обеспечения - это процесс создания знания. Хотя архитектурные решения могут быть разработаны до начала создания кода, подтверждение (или не подтверждение) правильности этой архитектуры имеет место всегда после начала разработки. На практике реальная архитектура программного обеспечения всегда материализуется в процессе разработки, даже если подробный документ, описывающий эту архитектуру, был составлен предварительно. При предварительном определении архитектуры невозможно учесть ни всей сложности, с которой приходится сталкиваться в процессе ее реализации, ни обратной связи с процессом разработки. Кроме этого, заданная предварительно и подробно архитектура не чувствительна к обратной связи с заказчиками. Процесс разработки, направленный на создание знания, предполагает, что архитектура программного обеспечения возникнет во время создания программного кода, и исключает затраты на его предварительное определение.

Исследователями определены четыре подхода, ведущие к созданию успешного программного обеспечения:

1. Ранний релиз с минимумом функциональных возможностей с тем, чтобы потребители могли его оценить и выразить свое мнение, пожелания и претензии.
2. Ежедневный выпуск сборок, их тестирование и учет (в дальнейшей разработке) результатов тестирования.
3. Наличие коллектива и (или) лидера с достаточным опытом и развитой интуицией, позволяющих принимать правильные решения.

4. Использование модульной архитектуры, поддерживающей возможность добавления новых функций.

При этом необходимо не только организовать процесс разработки, который поощряет систематическое и целенаправленное приобретение опыта, но и систематически совершенствовать этот процесс. Каждая аномалия должна инициировать поиск источника, вызвавшего проблему, а также поиск средств ее решения, а затем соответствующие изменения должны быть внесены в процесс разработки, чтобы предотвратить возникновение этой проблемы вновь. Усилия по совершенствованию процесса разработки должны стать стремлением и ответственностью Команд разработки, и в рамках производственного процесса необходимо выделить специальное время для работы над совершенствованием этого процесса на регулярной основе.

Предельно отсроченное принятие решений. Принцип предполагает, что решение следует принимать не на основе предположений и прогнозов, а после появления существенных фактов.

В процессе создания системы следует избегать принимать решения, которые будет трудно в последствии изменить. Необходимо, чтобы в определенных точках процесса (там, где, скорее всего, потребуется вносить изменения) сохранялась возможность выбора того или иного варианта. В условиях неопределенности, особенно когда она сопровождается сложностью, наиболее успешный подход заключается в том, чтобы экспериментировать с различными вариантами, выполняя **параллельную проработку альтернатив** и откладывая принятие решения до последнего момента.

Предельно быстрая доставка продукта клиенту. Принцип предполагает ликвидацию непроводительных затрат, что обеспечивает в свою очередь сокращение издержек на разработку. При этом скорость невозможна без высокого качества, что в свою очередь требует выстраивания производственного процесса таким образом, чтобы качество было встроено в создаваемый продукт. Кроме этого, чтобы обеспечить быстрое выполнение заказов, необходимо хорошо понимать потребности клиентов, для этого разработку рекомендуется выполнять короткими циклами с получением обратной связи от клиентов по результатам каждой итерации.

Мотивация команды. Принцип предполагает управление коллективами разработчиков, основанный на четырех предпосылках:

- 26) разработка продукции под руководством лидера;
- 27) технические специалисты высшей квалификации;
- 28) планирование и контроль, основанные на ответственности.

Уважение к людям означает, что коллективу дается общий план и определяются реальные задачи, а дальше коллективу предоставляется свобода действий в их выполнении. Работающие по этому принципу Команды разработки являются самоорганизующимися кросс-функциональными командами.

Оптимизировать целое. Принцип предполагает ориентацию процесса разработки на поставку завершеного продукта, ориентированного на удовлетворение потребностей клиента.

Процесс разработки программного обеспечения известен своей тенденцией к оптимизации отдельных частей системы или стадий процесса.

Бережливое производство позволяет оптимизировать поток создания ценности с момента принятия заказа и до поставки готового продукта и удовлетворения запросов клиента. Если организация сосредотачивается на оптимизации отдельных этапов создания продукта, а не на потоке поток создания ценности в целом, можно почти гарантировать, что процесс предоставления ценности будет неэффективным.

Для внедрения указанных принципов в процессы разработки необходимо участие и целостное видение процессов руководством организации, выполнение мероприятий по стандартизации, а также принятие и разделение всеми специалистами Команд разработки принципов бережливости.

5. КОНВЕЙЕР НЕПРЕРЫВНОЙ РАЗРАБОТКИ

Конвейер непрерывной разработки – инженерная практика, в рамках которой команды производят программное обеспечение в короткие циклы, гарантируя, что программное обеспечение может быть надежно выпущено в любое требуемое время. Он направлен на создание, тестирование и выпуск программного обеспечения с большей скоростью и частотой. Такой подход помогает снизить стоимость, время и риск внесения изменений за счет дополнительных обновлений приложений в производственной среде.

Конвейер непрерывной доставки представляет рабочие процессы, действия и их автоматизацию, необходимые для реализации новой функциональности Системы от разработки идей до выпуска версии, имеющей ценность для конечного пользователя.

Создание и обслуживание конвейера непрерывной разработки предоставляет каждой группе команд возможность предоставлять пользователям новые функциональные возможности гораздо чаще, чем в традиционных (водопадных) процессах. В некоторых случаях «непрерывный» может означать ежедневные выпуски версий или даже выпуск несколько раз в день. В других случаях непрерывный может означать еженедельные или ежемесячные выпуски — все, что удовлетворяет требованиям и целям создания Системы.

Конвейер непрерывной разработки (КНР) представляет простые и повторяемые рабочие процессы и их автоматизацию, необходимые для реализации новой функциональности Системы, начиная от разработки идей до выпуска версии конечного пользователя. Как показано на рисунке ниже, конвейер состоит из четырех аспектов (Рисунок 31):

- 41) Непрерывное исследование (CE),
- 42) Непрерывная интеграция (CI),
- 43) Непрерывное развертывание (CD),
- 44) Выпуск по требованию.

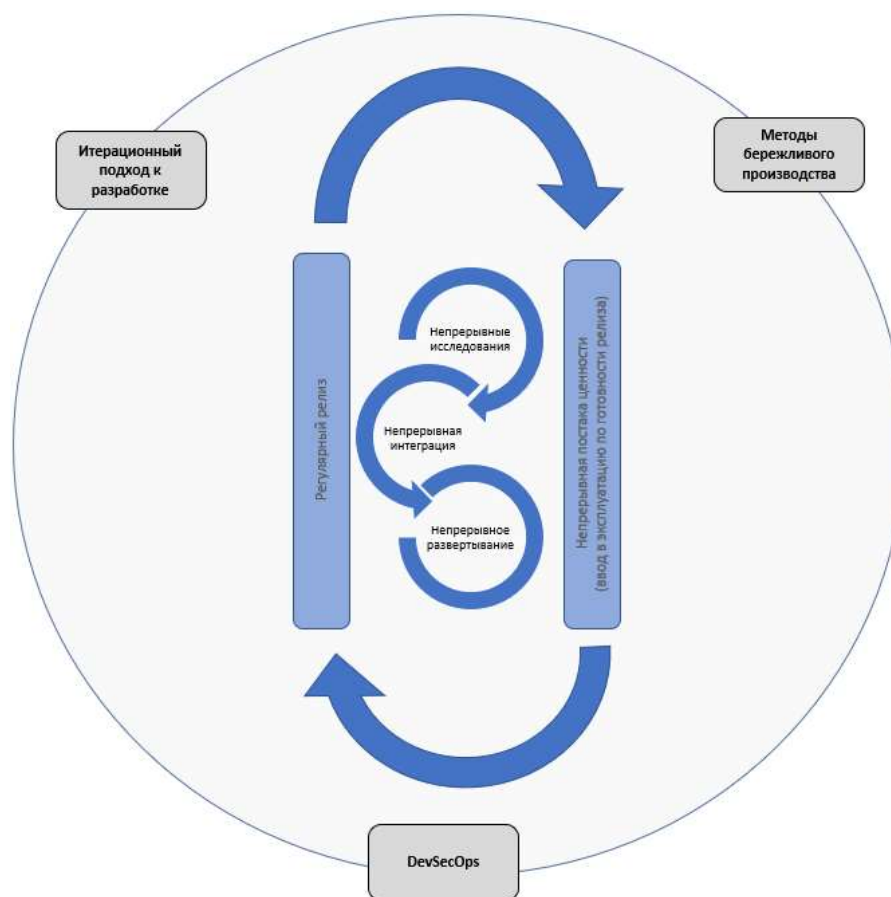


Рисунок 31. Конвейер непрерывной разработки

Внутренние циклы обратной связи направлены на улучшение процессов, в то время как внешняя обратная связь направлена на улучшение качества создаваемого продукта (Системы).

Каждый Пул кросс-функциональных команд создает и поддерживает или совместно использует конвейер с активами и технологиями, необходимыми для обеспечения ценности решения как можно более независимо. Первые три элемента конвейера (CE, CI и CD) работают вместе, чтобы поддержать поставку небольших партий новой функциональности, которые выпускаются для удовлетворения запросов пользователей.

Методика формирования и последовательных улучшений КНР в ходе выполнения работ представлена в Приложении к настоящему документу.

Непрерывное исследование (CE) фокусируется на понимании проблем/потребностей клиента и разработки предложений по решению для удовлетворения этой потребности. При этом используются практики дизайн-мышления (см. разделы 7). Исследование начинается с идеи или гипотезы о ценности для клиентов, как правило, в ответ на отзывы клиентов или исследования их запросов. Затем идеи анализируются и дополнительно исследуются, что приводит к пониманию того, что необходимо в качестве

минимально жизнеспособного продукта или минимальной достаточной версии Системы (МДВ). Это обеспечивает пространство решений для изучения того, как должна быть разработана архитектура Системы, а также какие функции могут удовлетворить потребности клиента.

Непрерывная интеграция (CI) фокусируется на реализации функций из Бэклога и последующем их уточнении. После того, как конкретные функции спроектированы, Команды разработки реализуют их и интегрируют в существующую функциональность Системы в контуре разработки для проведения комплексного тестирования.

Непрерывное развертывание (CD) предполагает передачу версий системы из контура тестирования в контур эксплуатации. Этот шаг делает функции доступными в рабочей среде, где Ведомство определяет подходящее время для их выпуска релиза Системы для клиентов. Этот аспект также позволяет Рабочей группе откатывать или исправлять ошибки, когда это необходимо.

Выпуск по требованию - это возможность сделать реализованную функциональность доступной для клиентов сразу или в определенном порядке на основе понимания актуальных потребностей клиентов. Это также позволяет тщательно контролировать величину риска, связанного с каждым вводом релиза в эксплуатацию. Выпуск по требованию также включает в себя критически важные действия конвейера непрерывной разработки, которые сохраняют стабильность релиза в течение длительного времени после выпуска.

Хотя этапы реализации функциональности в конвейере описываются последовательно, конвейер работает в итерационном режиме, который позволяет Командам разработки формировать одну или несколько гипотез, строить решения для проверки каждой гипотезы, проводить исследование гипотез и извлекать уроки из реализации гипотез.

5.1 Метрики улучшения производственного процесса

Команды разработки, как правило, имеют конвейер разработки — в противном случае они не смогли бы выпустить какую-либо ценность вообще.

Первым шагом к улучшению потока ценностей является изучение текущего конвейера. Со временем существующий процесс должен быть расширен таким образом, чтобы обеспечить изменения в производственном цикле – от реализации новых функций до обслуживания Системы в ходе ее эксплуатации.

Для улучшения производственного процесса используют метрики:

45) Время процесса — это время, необходимое для выполнения работы за один шаг.

46) Время выполнения — это время, которое требуется с момента выполнения работы на предыдущем шаге до ее выполнения на текущем шаге.

47) Время задержки — это время, когда не происходит никакой работы. Понимание и устранение ненужных задержек имеет решающее значение для улучшения потока ценностей.

48) Процент завершенности и точности представляет собой процент работы, которую следующий шаг может обработать без необходимости доработки. Часто задержки вызваны низким качеством в восходящих (предыдущих) шагах. Процентная полная и точная метрика помогает определить шаги, где может происходить низкое качество и приводить к увеличению времени выполнения заказов, что приводит к задержкам доставки стоимости.

5.2 Определение возможностей для улучшения производственного процесса

Команды ищут возможность повысить эффективность каждого шага, следовательно, сокращая общее время выполнения работ. Это включает в себя определение времени процесса, а также качества (в процентах завершения и точности) каждого шага. Чем выше это число, тем меньше требуется доработок, и тем быстрее работа движется по системе.

Время задержки (время между шагами) часто является наиболее значительным начальным фактором. Время задержки представляет собой передачу, ожидание и другие отходы, не связанные с добавленной стоимостью. Этот процесс имеет две значительные задержки и значительный объем доработок на первом этапе процесса развертывания. Сокращение задержек, как правило, является самым быстрым и простым способом снижения общего времени выполнения заказа. Еще одной приоритетной областью для улучшения является любой шаг с низкими показателями затрат времени и ресурсов, поскольку сокращение доработки позволяет Командам разработки сосредоточиться на создании ценности (например, для программного решения, вместо исправления ошибок команда может сосредоточиться на новых функциях).

5.3 Визуализация производственного процесса

Чтобы заинтересованные стороны в Рабочей группе могли визуализировать и отслеживать текущую работу Команд разработки, предлагается использовать систему Канбан (Рисунок 32).

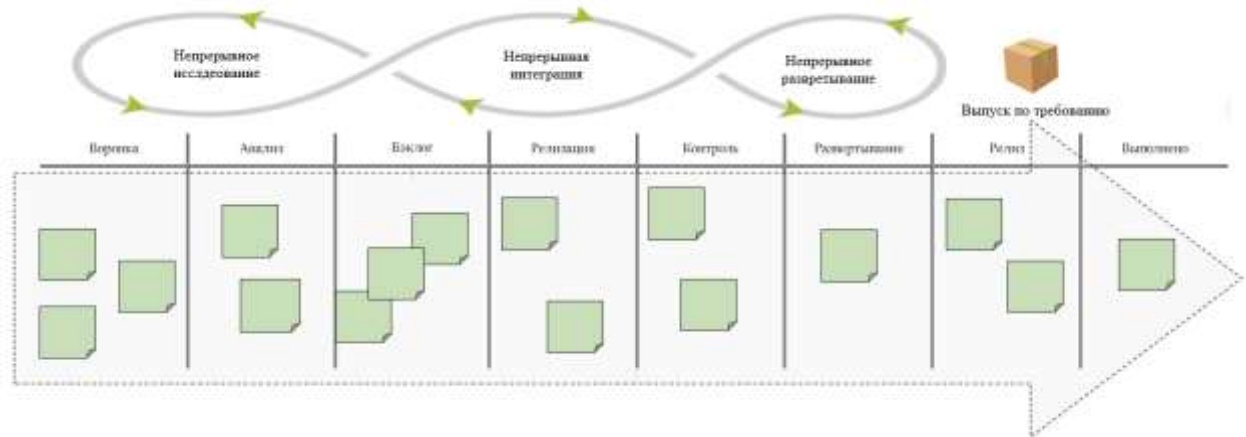


Рисунок 32. Отслеживание непрерывной поставки с помощью Канбан

Информационная доска Канбан, при этом, может состоять из следующих основных состояний:

49) Воронка — определение новых функций или улучшений существующей функциональности.

50) Анализ — уточнение функций с ключевыми атрибутами, включая гипотезу ценности для клиентов и критериев их реализации.

51) Бэклог — после анализа функции ранжируются по приоритету их реализации.

52) Реализация — основные функции из Бэклога разрабатываются и интегрируются в существующую функциональность Системы.

53) Контроль — тестирование реализованной функциональности.

54) Развертывание — по мере завершения работ реализованные компоненты развертываются в контуре эксплуатации, где они ожидают выпуска.

55) Релиз — развертывание в контуре эксплуатации и проверка гипотез ценности реализованной функциональности для клиента.

56) Готово — завершение работ при необходимости, предоставление реализованной функциональности клиентам.

6. ВСТРОЕННОЕ КАЧЕСТВО

Встроенное качество является основным принципом бережливого производства и направлено на снижение задержек выполнения работ, связанных с переделками и исправлением дефектов.

Встроенное качество включает 5 основных аспектов:

- 57) Создание потока функциональности;
- 58) Обеспечение качества архитектуры;
- 59) Обеспечение качества кода;
- 60) Обеспечение системного качества;
- 61) Обеспечение качественного выпуска версий Системы.



Рисунок 33. Основные аспекты встроенного качества

6.1 Создание потока функциональности

Создание потока функциональности имеет основополагающее значение для всех команд разработки, поскольку оно описывает, как избежать ошибок, переделок и других причин, которые замедляют разработку.

Поток в процессе создания (развития) Системы обеспечивается использованием с помощью:

- 62) практики разработки на основе тестирования,
- 63) практики раннего тестирования,
- 64) технологии конвейера непрерывной разработки.

6.1.1 Практика раннего тестирования

Вместо того, чтобы выполнять большое количество тестов в конце разработки, команды должны выполнять многие тесты на ранних стадиях разработки кода. Раннее тестирование применяется как к функциональным требованиям, так и к нефункциональным требованиям, к производительности, надежности и другим требованиям к качеству Системы, указанных в ТЗ.

Тесты должны выполняться быстро, и команды должны стремиться автоматизировать их. Поскольку более крупные, основанные на пользовательском интерфейсе, сквозные тесты выполняются намного медленнее, чем небольшие автоматизированные тесты, необходимо формировать сбалансированный набор тестов с большим количеством небольших быстрых тестов и меньшим количеством больших медленных тестов.

Поскольку тесты могут расти с течением времени (по мере роста объема кода), они начинают вносить задержки в работу команд. Поскольку настройка и выполнение полных наборов тестов может занять значительное время, команды могут создавать сокращенные наборы тестов и тестовых данных («дымовой тест»). Кроме этого, повышение скорости тестирования может быть обеспечено использованием более производительного оборудования в тестовых контурах. Для этого команды должны взаимодействовать с системной командой, чтобы сбалансировать скорость и качество тестирования (Рисунок 34).

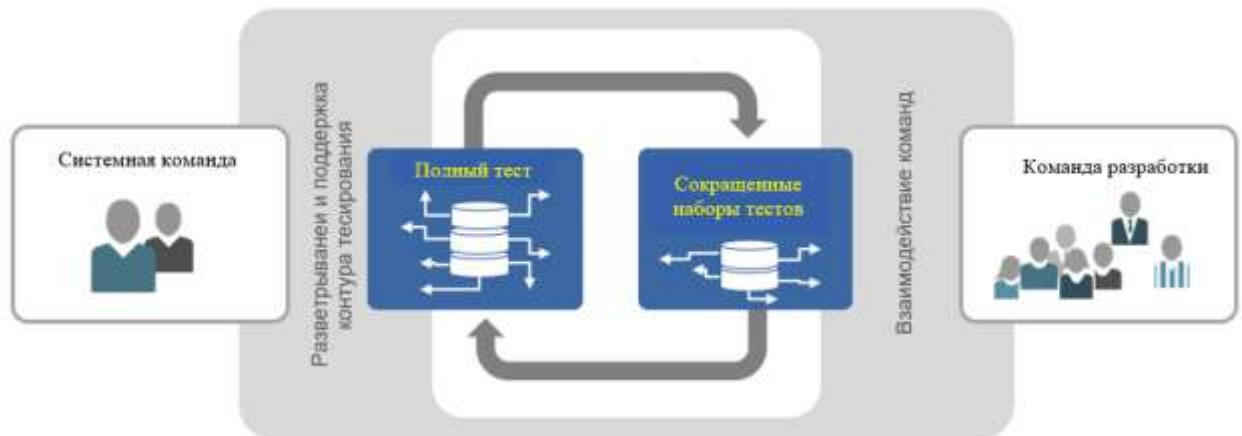


Рисунок 34. Оптимизация набора тестов

6.1.2 Разработка на основе тестирования

Разработка на основе тестирования (РНТ) — это философия и практика, которая включает в себя построение и выполнение тестов перед реализацией кода или компонента Системы (Рисунок 35). Проверяя их на соответствие серии согласованных тестов, РНТ — практика гибкого тестирования — улучшает результаты системы, гарантируя, что реализация системы соответствует ее требованиям. РНТ, наряду с поведенческой разработкой, является частью

подхода «сначала тест» для достижения встроенного качества. Написание тестов сначала создает более сбалансированный портфель тестов с множеством быстрых автоматизированных тестов разработки и меньшим количеством медленных, ручных, сквозных тестов.



Рисунок 35. Разработка на основе тестирования

РНТ определяется следующими правилами:

65) Сначала напишите тест, убедившись, что разработчик понимает требуемое поведение. Это может быть новый тест или модификация существующего теста.

66) Запустите тест и наблюдайте, как он завершается ошибкой. Поскольку кода еще нет, это может показаться бессмысленным, но оно выполняет две полезные цели: проверяет работу теста, включая любые тестовые программы, и демонстрирует, как система будет вести себя, если код неправильный.

67) Напишите минимальный объем кода, необходимый для прохождения теста. В случае сбоя переработайте код или тест до тех пор, пока он не пройдет регулярно.

68) Продолжайте реализацию нового кода до тех пор, пока не пройдут все тесты. Этот шаг дает разработчику уверенность в том, что его изменения соответствуют текущим требованиям и не создали ошибку в другой части системы.

69) Рефакторинг по мере необходимости для обеспечения соответствия проекта изменяющимся требованиям (например, возникающий дизайн). Разработчики постоянно обновляют свои

проекты, чтобы гарантировать, что меняющиеся требования и растущая кодовая база не приведут к низкому качеству кода.

6.2 Обеспечение качества архитектуры

Архитектура Системы в конечном итоге определяют, насколько хорошо Система может поддерживать текущие и будущие потребности конечных пользователей (клиентов). Качество архитектуры облегчает реализацию как функциональных, так и нефункциональных требований, а также упрощает тестирование Системы.

Поскольку требования к Системе со временем могут изменяться, проектные и архитектурные решения заложенные на начальном этапе создания Системы также могут изменяться. Традиционные процессы, которые заставляют не изменять принятые проектные решения, могут привести к снижению качества разработки и конечной версии Системы в условиях изменяющихся требований. Определение наилучшего проектного решения требует знаний, полученных в ходе экспериментов, моделирования, прототипирования и других исследований. Это также требует подхода к проектированию на основе формирования и исследования альтернативных вариантов архитектуры, чтобы прийти к наилучшему решению.

Архитектура также влияет тестируемость системы. Модульные компоненты, которые взаимодействуют через четко определенные программные интерфейсы, создают «швы», которые позволяют тестировщикам и разработчикам заменять дорогие или медленные компоненты тестовыми «заглушками».

6.3 Достижение качества кода

Все возможности Системы в конечном итоге реализуются программным кодом (или компонентами) Системы. При этом, скорость и простота добавления новых возможностей зависят от того, насколько быстро и надежно разработчики смогут его модифицировать. Рекомендуется использовать практики модульного тестирования, разработки на основе встроенных тестов, парной работы и стандарты коллективной ответственности и кодирования.

6.3.1 Модульное тестирование и разработка на основе тестов

Практика модульного тестирования разбивает код на части и гарантирует, что каждая часть имеет автоматизированные тесты для его выполнения. Эти тесты запускаются автоматически после каждого изменения программного кода и позволяют разработчикам вносить быстрые изменения, будучи уверенными,

что модификация не сломает другую часть Системы. Тесты также служат документацией и являются исполняемыми примерами взаимодействия с интерфейсом компонента, чтобы показать, как этот компонент должен использоваться. Разработка на основе тестирования направляет создание модульных тестов, указывая тест для изменения перед его созданием. Это заставляет разработчиков мыслить более широко о проблеме, включая крайние случаи и граничные условия перед реализацией. Лучшее понимание приводит к более быстрой разработке с меньшим количеством ошибок и меньшим количеством доработок.

6.3.2 Парная работа

При парной работе два разработчика работают с одинаковыми изменениями на одной и той же рабочей станции. Один – пишет код, а другой – обеспечивает обзор и обратную связь в режиме реального времени. При этом разработчики часто меняют роли. Такое сопряжение создает и поддерживает качество, поскольку код будет содержать общие знания, перспективы и лучшие практики от каждого участника. Это также повышает и расширяет набор навыков для всей команды, поскольку товарищи по команде учатся друг у друга.

6.3.3 Стандарты коллективной собственности и кодирования

Коллективная собственность уменьшает зависимости между командами и гарантирует, что любой отдельный разработчик или команда будет работать эффективно. Любой член команды может добавлять функциональность, исправлять ошибки, улучшать дизайн или рефакторить программный код. Поскольку код не принадлежит одной команде или отдельному человеку, поддержка стандартов кодирования способствует согласованности, чтобы каждый сотрудник мог понять и поддерживать качество каждого программного компонента.

6.4 Достижение качества системы

В то время как качество программного кода и архитектуры гарантирует, что системные артефакты могут быть легко реализованы и изменены, качество системы подтверждает, что Система работает так, как ожидалось.

6.5 Использование практик развития Системы, основанного на поведении

Развитие, основанное на поведении, требует согласование действий всех заинтересованных сторон в создании (развитии) Системы:

29) заинтересованные стороны, ориентированные на конечного пользователя, понимают их потребности, а также относительную важность реализуемых командами требований к Системе;

30) заинтересованные стороны, ориентированные на развитие, понимают альтернативные решения и их технологическую осуществимость;

31) заинтересованные стороны, ориентированные на тестирование, рассматривают исключения, граничные случаи и граничные условия для реализуемого командами разработки поведения.

6.6 Модельно-ориентированная системная инженерия (МОСИ)

Модельно-ориентированная системная инженерия— это практика разработки набора связанных системных моделей, которые помогают определять, проектировать и документировать разрабатываемую Систему. Эти модели обеспечивают эффективный способ изучения, обновления и передачи системных аспектов заинтересованным сторонам, при этом значительно уменьшая или устраняя зависимость от традиционных документов.

Также МОСИ обеспечивает высокоуровневое, полное представление обо всех требуемых функциях Системы и о том, как проект создания Системы реализует ее.

Хотя модели не являются идеальным представлением системы, они обеспечивают знания и обратную связь быстрее и более экономически эффективно, чем только реализация. И они позволяют моделировать сложные взаимодействия в создаваемой Системе с соответствующей точностью для обучения. На практике инженеры используют модели для получения знаний и служат руководством для внедрения системы.

6.7 Непрерывная интеграция комплексного решения

Масштабирование Системы приводит к тому, что многие инженеры вносят множество небольших изменений, которые необходимо постоянно проверять на наличие конфликтов и ошибок. Методы непрерывной интеграции и непрерывной поставки обеспечивают разработчикам быструю обратную связь. Каждое изменение быстро создается, а затем интегрируется и тестируется на нескольких уровнях, включая среду развертывания. Непрерывная интеграция и непрерывная поставка автоматизирует процесс перемещения изменений на всех этапах и знает, как реагировать в случае сбоя теста. Тесты качества для нефункциональных требований также должны быть автоматизированы. При этом некоторые функциональные и нефункциональные тесты могут быть выполнены только вручную.

6.8 Обеспечение качества выпуска версий

Выпуск релизов Системы позволяет получить обратную связь от конечных пользователей и/или заказчика Системы. Сервисная архитектура, определение стандартные интерфейсы между компонентами Системы, позволяют независимо выпускать более мелкие изменения на уровне компонентов. Меньшие изменения обеспечивают более быстрые, частые и менее рискованные выпуски, но для обеспечения качества требуется, как было сказано выше (раздел 5) автоматизированный конвейер (Рисунок 36).



Рисунок 36. Автоматизированный конвейер непрерывной разработки

6.8.1 Поддержка соответствия нормативным требованиям

Для систем, которые должны продемонстрировать объективные доказательства соответствия или аудита, выпуск версии имеет дополнительные условия. Исполнитель должны доказать, что система соответствует своему назначению и не имеет непреднамеренных, вредных последствий.

6.8.2 Масштабируемое определение выполненных работ

Определение выполненных работ является важным способом обеспечения того, чтобы разработку версии Системы можно было считать завершенной. Непрерывное развитие функциональности Системы требует масштабированного определения выполненной работы, чтобы гарантировать, что необходимая работа выполняется в нужное время разными командами и группой команд в целом.

7. ДИЗАЙН-МЫШЛЕНИЕ

7.1 Общие положения

Дизайн-мышление – это методология решения инженерных, деловых и прочих задач, основывающаяся на творческом подходе. Главной особенностью дизайн-мышления, в отличие от аналитического мышления, является не критический анализ, а творческий процесс, в котором порой самые неожиданные идеи ведут к лучшему решению проблемы.

Идею дизайн-мышления впервые сформулировал Герберт Саймон в 1969 году. Герберт Саймон выделяет следующие этапы в дизайн-мышлении (Рисунок 37):

- 70) определение проблемы,
- 71) исследование,
- 72) формирование идей,
- 73) прототипирование,
- 74) выбор лучшей идеи,
- 75) внедрение идеи.



Рисунок 37. Концепция дизайн-мышления

Дизайн-мышление - ориентированный на клиента процесс разработки, который создает прибыльные и устойчивые на протяжении всего жизненного цикла продукты. Подход к проектированию с использованием дизайн-мышления выходит за рамки традиционного акцента на особенностях и функциях предлагаемого продукта. Вместо этого он подчеркивает понимание проблемы, которую необходимо решить, контекста, в котором это решение будет использоваться, а также эволюции решения.

Традиционные каскадные подходы к разработке продукта являются последовательными: сначала определяются требования; затем решения проектируются, создаются и поставляются на рынок. Основное внимание, как правило, уделяется наиболее очевидным проблемам. Часто успех определяется внедрением решения, отвечающего требованиям, а не потребностям пользователя, в результате чего продукты и услуги с непригодными для использования или игнорируемыми функциями не соответствуют запросам пользователей.

Дизайн-мышление представляет собой подход к разработке продуктов, в котором применяются дивергентные¹⁶ и конвергентные¹⁷ методы для понимания проблемы, создания продукта и предоставления его пользователям.

Дизайн-мышление сфокусировано на следующих основных свойствах продукта:

76) **Полезность** – Нужен ли клиентам и пользователям продукт?

77) **Осуществимость** – Можем ли мы создать продукт?

78) **Жизнеспособность** - создает ли то, как мы создаем и предлагаем продукт пользователям, больше ценности, чем затрат на его создания? Например, на коммерческом предприятии мы прибыльны?

79) **Устойчивость в развитии** – можем ли мы развивать наш продукт так, чтобы он соответствовал изменяющимся ожиданиям пользователей?

В процессе прохождения этих этапов формулируются проблемы, задаются правильные вопросы, придумываются идеи и выбираются лучшие решения. При этом данные этапы не являются линейными – разные этапы можно проходить одновременно и возвращаться к определенным этапам при необходимости.

7.2 Определение проблемы

Определение проблемы – это самый первый и самый важный этап дизайн-мышления, поскольку ошибочное определение проблемы может привести к тому, что в результате будет получено решение не той проблемы, которую нужно было решить.

Также для определения проблемы нужно сформировать понимание потребностей клиентов, а также сформулировать критерий достижения результата (например: ответить на вопрос, что является успешным результатом создания продукта).

7.3 Исследование

Второй этап в дизайн-мышлении – исследование – начинается с обзора истории проблемы: какие решения создавались ранее и какие результаты при этом были получены. Обзор истории поможет избежать «изобретения велосипеда» и ошибок, которые были совершены прежде.

На этом этапе важно взаимодействие с конечными пользователями – например интервьюирование или анкетирование с целью получение понимания

16 Дивергентный метод- поиск множества альтернативных решений проблемы

17 Конвергентный метод - точное использование инструкций по решению задачи

проблемы конечными пользователями (клиентами) и, возможно, идей о путях ее решения. Иногда наиболее эффективным способом получить информацию о проблеме будет наблюдение, поскольку в реальности конечные пользователи могут вести себя иначе, как они рассказывают при интервьюировании.

7.4 Формирование идей

На этом этапе необходимо собрать полученную на предыдущих стадиях информацию, чтобы формально зафиксировать потребности ваших пользователей, после чего как правило проводится процесс мозгового штурма. Главная задача мозгового штурма – придумать как можно больше самых разных идей, решающих проблему. При этом недопустимо останавливаться на одной идее, даже если она представляется релевантной.

Практика мозгового штурма предполагает участие нескольких человек, работающих совместно ориентировочно в течение одного дня. При этом высказываемые во время мозгового штурма идеи нельзя критиковать.

7.5 Прототипирование

После формирования перечня альтернатив решения проблемы необходимо провести их анализ с целью улучшения или объединения близких по смыслу решения.

Рекомендуется получение обратной связи с конечными пользователями (клиентами) для того, чтобы внести соответствующие изменения в ваш решения, после чего создать несколько работающих прототипов продукта, обеспечивающего решение проблемы клиента.

7.6 Выбор лучшего решения проблемы

Для выбора наилучшего решения возможно использование обратной связи от конечных пользователей или применение формальных методов на базе экспертных оценок, например описанных в разделе 8.1.

7.7 Внедрение работающего продукта

На этом этапе осуществляется создание и внедрение работающего продукта.

7.8 Оценка результатов

Создание и внедрение продукта – это не последний этап в процессе дизайн-мышления. Далее требуется организация процедур мониторинга с целью оценки качества решения первоначально определенной проблемы в соответствии с

выбранным критерием и/или степени удовлетворенности клиента. На основе мониторинга принимаются решения о доработке или изменении продукта.

8. ОПРЕДЕЛЕНИЕ ПРИОРИТЕТОВ ЗАДАЧ

8.1 Метод анализа иерархий

Метод анализа иерархий предложен Саати и предполагает определение цели выполнимых работ, критериев оценки достижения цели и набора возможных альтернатив с последующим построением оценок вкладов альтернатив в достижение цели на основе парных сравнений (альтернативы сравниваются попарно, затем результаты обрабатываются и формируется вектор приоритетов):

80) На первом этапе методом экспертных оценок определяют веса критериев относительно цели.

81) На втором этапе методом экспертных оценок определяют веса альтернатив относительно каждого критерия.

82) На третьем этапе методом определяют приоритеты реализации функций относительно цели.

Метод является универсальным (фактически на нем базируются другие, более «легкие» методы определения приоритетов), но для большого числа критериев и альтернатив требует проведения ресурсоемких матричных вычислений, что затрудняет его использование без автоматизации.

8.2 Более Ценная и Короткая Работа Сначала

Более Ценная и Короткая Работа Сначала (WSJF) – это метод приоритизации задач, предложенный Доном Рейнертсоном в книге 2009 года (<https://www.amazon.com/Principles-Product-Development-Flow-Generation/dp/1935401009>) на основе «Самая короткая работа сначала» SJF используемый в фреймворке SAFe (scaledagileframework.com).

Смысл метода в том, чтобы наиболее ценные для пользователей и быстровыполнимые задачи брать с более высоким приоритетом.

Формула расчёта приоритета выглядит так:

$$\text{WSJF} = \text{Стоимость задержки} / \text{Продолжительность}$$

$$\text{WSJF} = \text{Стоимость задержки} / \text{Сложность задачи}$$

Стоимость задержки определяется следующим образом:

$$\text{Стоимость задержки} = \text{Ценность для пользователей} + \text{Временная критичность} + (\text{Снижение рисков или Новые возможности})$$

Оценка показателей, составляющих Стоимость задержки осуществляется методом экспертных оценок, при этом эксперты, оценивая показатели отвечают на вопросы:

83) Ценность для пользователей (User Value): насколько сильно просят об это пользователи? Какой потенциально негативный эффект будет, если это выполнить позже, а не раньше?

84) Временная критичность (Time Criticality): задерживает ли функция реализацию других функций? Нужно ли это выпустить к определенной дате? Есть ли риск того, что опоздание с реализацией функции обесценит ранее проделанную работу?

85) Снижение рисков (Risk Reduction): снижает ли реализация функции какие-то риски? Будет ли это позитивно влиять на качество в других областях? Будет ли эффект сиюминутным или долгосрочным?

86) Новые возможности (Opportunity Enablement): откроет ли реализация функции новые возможности для продукта? Поможет ли расширить пользовательскую аудиторию?

Оценки проставляются путём сравнения задач относительно друг друга, от 1 до 21 из последовательности Фибоначчи (1, 3, 5, 8, 13, 21...). Полученные оценки суммируются и делятся на размер работы. Размер работы тоже оценивается числами Фибоначчи методом экспертных оценок.

	Ценность для пользователей	Временная критичность	Снижение риска/ Новые возможности	Стоимость задержки	Сложность задачи	WSJF	Приоритет
Задача 1	1	3	13	17	2	8,50	1
Задача 2	13	5	13	31	5	6,20	2
Задача 3	5	1	8	14	3	4,67	3
Задача 4	13	5	3	21	5	4,20	4
Задача 5	13	8	13	34	13	2,62	5

Правила проставления оценок:

87) Оценивать задачи по одному столбцу за раз, указать единицу (минимальный размер) для наименьшего элемента, а затем оцените остальные задачи относительно наименьшей.

88) Следствие: должна быть минимум одна единица в каждой колонке.

89) Самое большое число по WSJF означает наиболее высокий приоритет.

Суммируя показатели в трех первых столбцах и деля на размер задачи, мы получаем значения показателя, сортируя которые от большего к меньшему, мы получаем приоритет задач.

ПЕРЕЧЕНЬ ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

Термин	Определение	Источник
DevOps	Набор практик и средств автоматизации разработки, нацеленных на активное взаимодействие разработчиков программного обеспечения со специалистами служб эксплуатации и поддержки, с тем чтобы обеспечить взаимную интеграцию их рабочих процессов. Основу составляет идея о тесной взаимозависимости процессов разработки и эксплуатации программного обеспечения (ПО).	Описание производственного процесса по созданию сервисов на цифровой платформе «ГосТех»
DevSecOps	Интегрированный набор практик и средств автоматизации разработки, обеспечивающий управление безопасностью в рамках DevOps и автоматизирующий основные процессы, связанные с обеспечением безопасности, в том числе – за счет использования автотестов функций безопасности	Инструменты DevSecOps, обеспечивающие безопасность рабочих процессов DevOps. Кев Зеттлер. Atlassian.

Архитектура Системы	Совокупность основных понятий или свойств системы в окружающей среде, воплощенной в ее элементах, отношениях и конкретных принципах ее проектирования и развития. Архитектура системы состоит из шести архитектурных представлений: а) архитектура деятельности; б) архитектура программных средств (программная архитектура); в) архитектура данных; г) интеграционная архитектура; д) инфраструктура информационной безопасности; е) технологическая архитектура.	ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011 «Национальный стандарт Российской Федерации. Системная и программная инженерия. Описание архитектуры». Типовое Соглашение о моделировании архитектуры государственной информационной системы
Архитектурная поддержка	Специальное программное обеспечение и техническая инфраструктура, которые в каждый момент времени позволяют реализовывать высокоприоритетную функциональность продукта без задержек и перепроектирования.	Типовое Соглашение о моделировании архитектуры государственной информационной системы
Бережливое производство	Концепция организации бизнеса, ориентированная на создание привлекательной ценности для потребителя путем формирования непрерывного потока создания ценности с охватом всех процессов организации и их постоянного совершенствования через вовлечение персонала и устранение всех видов потерь	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь

Ведомственная программа цифровой трансформации (ВПЦТ)	Документ, содержащий мероприятия, направленные на поэтапную реализацию цифровой трансформации государственного органа, цели и соответствующие им показатели (включая их значения) результативности и эффективности системы госуправления, которые планируется достигнуть посредством цифровой трансформации государственного органа в очередном плановом периоде, включая сведения об источниках и объемах необходимого для этого финансового обеспечения	Постановление Правительства РФ от 10 октября 2020г. № 1646 «О мерах по обеспечению эффективности мероприятий по использованию информационно-коммуникационных технологий в деятельности федеральных органов исполнительной власти и органов управления государственными внебюджетными фондами»
Визуализация (visualization)	Расположение всех инструментов, деталей, производственных стадий и информации о результативности работы производственной системы таким образом, чтобы они были четко видимы и чтобы каждый участник производственного процесса моментально мог оценить состояние системы.	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь
Встроенное качество	Методология приведения возможностей процессов и систем измерения в соответствие с требованиями потребителя к качеству продукции, в том числе предоставление доказательств выполнения данных требований.	ГОСТ 57 522 - 2017 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО Руководство по интегрированной системе

	<p>Термин "встроенное качество" предусматривает оценку исполнителем показателей возможностей процессов с целью снижения рисков несоответствий на приемлемо низком уровне, а также специальную организацию операций контроля и аудита производственных процессов для поддержания установленной скорости потока создания ценности. Деятельность по встроенному качеству предусматривает, что ответственность за качество на этапе производства (оказания услуги) переходит к исполнителям.</p> <p>Деятельность по встроенному качеству необходимо проводить в первую очередь на этапах проектирования продукции и производственных процессов.</p> <p>Обеспечение встроенного качества существенно уменьшает необходимость в процедурах контроля качества, позволяет передать функцию контроля качества непосредственно исполнителям и реализовать принцип "3 НЕ" - не делай, не передавай, не принимай дефектную продукцию (результат услуги).</p>	менеджмента качества и бережливого производства
Государственная информационная система	Система, создаваемая в целях реализации полномочий государственных органов и обеспечения обмена информацией между этими органами, а также в иных установленных федеральными законами целях.	№ 149-ФЗ «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г.

	<p>Государственные информационные системы создаются, модернизируются и эксплуатируются с учетом требований, предусмотренных законодательством Российской Федерации о контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд либо законодательством Российской Федерации о государственно-частном партнерстве, о муниципально-частном партнерстве, законодательством о концессионных соглашениях, а в случаях, если эксплуатация государственных информационных систем осуществляется без привлечения средств бюджетов бюджетной системы Российской Федерации, в соответствии с иными федеральными законами.</p>	
<p>Государственная услуга, предоставляемая федеральным органом исполнительной власти, органом государственного внебюджетного фонда, исполнительным органом государственной власти субъекта Российской Федерации, а также органом местного самоуправления при осуществлении отдельных государственных полномочий, переданных федеральными</p>	<p>Деятельность по реализации функций соответственно федерального органа исполнительной власти, государственного внебюджетного фонда, исполнительного органа государственной власти субъекта Российской Федерации, а также органа местного самоуправления при осуществлении отдельных государственных полномочий, переданных федеральными законами и законами субъектов Российской Федерации, которая осуществляется по запросам заявителей в пределах установленных нормативными правовыми актами Российской</p>	<p>ФЗ от 27 июля 2010г. № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг»</p>

законами и законами субъектов Российской Федерации (Государственная услуга)	Федерации и нормативными правовыми актами субъектов Российской Федерации полномочий органов, предоставляющих государственные услуги	
Государственный контракт	Договор, заключенный от имени Российской Федерации, субъекта Российской Федерации государственным заказчиком для обеспечения соответственно государственных нужд, муниципальных нужд	п. 8 ст. 3 Федерального закона от 05.04.2013 № 44-ФЗ «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд»
Государственный контроль (надзор), муниципальный контроль в Российской Федерации	Государственный контроль (надзор), муниципальный контроль осуществляются на основе управления рисками причинения вреда (ущерба), определяющего выбор профилактических мероприятий и контрольных (надзорных) мероприятий, их содержание (в том числе объем проверяемых обязательных требований), интенсивность и результаты.	Федеральный закон от 31 июля 2020 г. № 248-ФЗ «О государственном контроле (надзоре) и муниципальном контроле в Российской Федерации»
Группа команд разработки	Несколько Команд разработки, совместно работающих над созданием и развитием Системы итерациями фиксированной длительности (рекомендуется – 12 недель) - Инкрементами	Термин определен настоящим документом
Группа требований	Часть полного объема требований к функциям Системы, каждая из которых обеспечивает отдельную самостоятельную ценность для	Постановление Правительства РФ от 6 июля 2015 г. N 676 "О требованиях к порядку

	клиентов. Для Групп требований должны быть установлены конечные сроки их реализации и критерии приёмки	создания, развития, ввода в эксплуатацию, эксплуатации и вывода из эксплуатации государственных информационных систем и дальнейшего хранения содержащейся в их базах данных информации" (с изменениями и дополнениями)
Демонстрация системы	Комплексное представление заинтересованным сторонам новых функций Системы, реализованных в рамках Инкремента	Термин определён настоящим документом
Жизненный цикл продукта	<p>1. Совокупность взаимосвязанных процессов, объединяемых в этапы, изменения состояния информационной системы от принятия решения о ее создании, формирования исходных требований к ней и до окончания ее эксплуатации (вывода из эксплуатации).</p> <p>2. Цикл, охватывающий четыре основных этапа продукта — «запуск», «рост», «зрелость» и «упадок» — и связанный с активным маркетингом продукта на рынке. (Иногда в основу жизненного цикла продукта закладывают процесс, состоящий из пяти этапов - запуска, роста, зрелости, насыщения и упадка).</p>	<p>Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации»</p> <p>ГОСТ Р 58537- 2019. УПРАВЛЕНИЕ ПРОДУКЦИЕЙ. Основные положения</p>

Заинтересованная сторона	Индивидуум, команда, организация или их группы, имеющие интерес в Системе	Типовое Соглашение о моделировании архитектуры государственной информационной системы
Инкремент	Выполнение видов работ по реализации группы требований, в том числе повторное выполнение в целях устранения выявленных недостатков	Постановление № 676
Инструмент	Средство осуществления действий, направленных на решение определенных задач или достижение определенной цели	ГОСТ 56407 - 2015 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО Основные методы и инструменты
Информационная система	Совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств	№ 149-ФЗ «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г.
Инфраструктура взаимодействия	Единый комплекс информационно-технологических и телекоммуникационных элементов, включающий: информационные элементы в составе следующих информационных систем и входящих в них подсистем: а) федеральная государственная информационная система "Единый портал государственных и муниципальных услуг (функций)";	Постановление Правительства РФ от 8 июня 2011 г. № 451 «Об инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для

	<p>б) федеральная государственная информационная система "Федеральный реестр государственных и муниципальных услуг (функций)";</p> <p>в) информационная система головного удостоверяющего центра, функции которого осуществляет уполномоченный федеральный орган исполнительной власти;</p> <p>г) федеральная государственная информационная система "Единая система идентификации и аутентификации в инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг в электронной форме";</p> <p>д) единая система межведомственного электронного взаимодействия;</p> <p>е) федеральная государственная информационная система "Единая система нормативной справочной информации";</p> <p>ж) федеральная государственная информационная система, обеспечивающая процесс досудебного (внесудебного) обжалования решений и действий (бездействия), совершенных при предоставлении государственных и муниципальных услуг;</p> <p>з) федеральная государственная информационная система "Единая информационная платформа национальной системы управления данными";</p>	<p>предоставления государственных и муниципальных услуг и исполнения государственных и муниципальных функций в электронной форме»</p>
--	---	---

	<p>и) федеральная государственная информационная система "Единая информационная система управления кадровым составом государственной гражданской службы Российской Федерации" и интегрированная с ней цифровая кадровая платформа органов публичной власти (платформа "Государственные кадры");</p> <p>к) информационная система обеспечения внутриведомственного и межведомственного документооборота и контроля исполнения поручений, в том числе с использованием облачных сервисов.</p>	
Итерационный подход к разработке	Выполнение последовательности итераций, каждая из которых включает в себя реализацию группы требований к Системе, predetermined набор целей, критерии приемки каждого требования, плановые сроки реализации итераций, критерии готовности к вводу в эксплуатацию	Постановление № 676
Канбан (kanban):	<p>Средство информирования, с помощью которого дается разрешение или указание на производство или передачу изделий в производстве, организованном по принципу вытягивания.</p> <p>Примечание - В переводе с японского языка Канбан означает "бирка" или "значок". Наиболее известным и распространенным примером таких средств</p>	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь

	<p>коммуникации служат карточки Канбан. Во многих случаях они представляют собой листки бумаги, иногда помещенные в прозрачные пластиковые конверты, на которых указана следующая информация: наименование детали, номер детали, внешний поставщик или внутренний процесс-поставщик, число изделий в упаковке, местоположение склада и процесса-потребителя. На карточке может быть помещен штрих-код для считывания или автоматического выставления счета.</p>	
Клиент	<p>Физическое или юридическое лицо, индивидуальный предприниматель, иные лица, получающие государственные услуги и (или) результаты осуществления государственных функций, а также лица, деятельность которых является объектом государственного контроля (надзора)</p>	<p>Материалы 32-го Всемирного Конгресса IPMA, сессия «Клиентоцентричное управление государственными проектами».</p>
Клиентоцентричность	<p>Определение и максимальное удовлетворение явных и скрытых потребностей пользователей с учетом разумных издержек и соблюдения интересов заинтересованных сторон</p>	<p>Материалы 32-го Всемирного Конгресса IPMA, сессия «Клиентоцентричное управление государственными проектами».</p>

Команда разработки	Кросс-функциональная команда специалистов размером от 7 до 12 человек, которая наделена полномочиями и способна определить, создать, протестировать и внедрить определенную функциональность Системы, работающая в рамках Инкремента короткими фиксированными циклами	Термин определен настоящим документом
Конвейер непрерывной доставки	Автоматизированные рабочие процессы, обеспечивающие реализацию новой функциональности Системы от проектирования до развертывания в составе релиза	Термин определен настоящим документом
Контур разработки	Часть платформы разработки, обеспечивающая разработку и/или адаптацию программного обеспечения Системы командами разработки	Термин определен настоящим документом
Контур тестирования	Часть платформы разработки, обеспечивающая проведение тестирования разработанного программного обеспечения Системы	Термин определен настоящим документом
Контур эксплуатации	Часть платформы разработки, обеспечивающая эксплуатацию Системы	Термин определен настоящим документом
Метод	Систематизированная совокупность шагов, действий, которые необходимо предпринять, чтобы решить определенную задачу или достичь определенной цели.	ГОСТ 56407 - 2015 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО

		Основные методы и инструменты
Минимально достаточная версия Системы (МДВ)	Версия Системы, обладающая минимальными, но достаточными функциями для удовлетворения одной или нескольких потребностей клиентов.	Термин определён настоящим документом
Непрерывная интеграция	Процесс обеспечения согласованного функционирования разработанных функций Системы, их проверки Системы в контуре тестирования, по результатам которого обеспечивается готовность релиза к развертыванию и выпуску.	Термин определён настоящим документом
Непрерывное исследование	Процесс, который поддерживает инновации в интересах реализации новой функциональности Системы путем постоянного изучения потребностей клиентов, и определения набора функций Системы, которые отвечают этим потребностям.	Термин определён настоящим документом
Непрерывное развертывание	Процесс, который обеспечивает автоматизированный перенос проверенных функции в контуре тестирования в контур эксплуатации, где они могут быть предоставлены клиентам	Термин определён настоящим документом

Очередь (автоматизированной системы)	Часть автоматизированной системы, для которой в техническом задании на создание в целом установлены отдельные сроки ввода и набор реализуемых функций	ГОСТ Р59853 - 2021 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения
План-график создания и развития Системы	Документ, определяющий количество и очередность итераций по разработке Системы (с учетом сформированных в техническом задании групп требований к Системе) и плановые сроки окончания каждой итерации.	Постановление № 676
Планирование Инкремента	Регулярно повторяющееся мероприятие по планированию работ в рамках предстоящего Инкремента, которое является основополагающим для Группы команд разработки и объединяет все Команды разработки для согласования общих целей выполнения работ	Термин определен настоящим документом
Платформа разработки	Программно-аппаратная среда, предоставляющая функционально полный набор информационно-технологических сервисов, позволяющих эффективно создавать, развивать и эксплуатировать прикладное программное обеспечение государственных информационных систем и их компонентов.	Постановление Правительства РФ от 12 октября 2020 г. № 1674 «О проведении эксперимента по созданию, переводу и развитию государственных информационных систем и их компонентов на единой

		цифровой платформе Российской Федерации «ГосТех»
Поток создания ценности (value stream)	<p>Все действия, как создающие, так и не создающие ценность, которые позволяют продукции пройти все процессы - от разработки концепции до запуска в производство и от принятия заказа до доставки потребителю</p> <p>Используется как интегральное понятие, включающее в себя материальные потоки (сырья, материалов, комплектующих, деталей и сборочных единиц, готовой продукции), информационные и финансовые потоки, направленные на создание и доставку готовой продукции потребителю в установленное время, в установленном месте, с установленной стоимостью, с последующим ее обслуживанием в процессе эксплуатации и утилизации.</p>	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь
Поток (stream)	Совокупность элементарных действий, которые управляются как целое, характеризуемое скоростью перемещения основной характеристики объекта.	ГОСТ Р 57524-2017 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО Поток создания ценности
Пред- и пост- планирование инкремента	Мероприятия по планированию до и после планирования инкремента, используются для подготовки к планированию планирования	Термин определён настоящим документом

	инкремента для Группы команд разработки и поставщиков, а также последующих действий после планирования.	
Программно-аппаратный комплекс (ПАК)	Интегрированная программно-аппаратная среда, имеющая компонентную структуру и предназначенная для развертывания и функционирования в соответствии с его назначением	Отчетные материалы по Гос заданию по государственному заданию от 19 октября 2020 № 071-00002-20-05
Продукт	Изделие и/или услуга, которые в настоящее время и в будущем будут предлагаться производителем и обладают определенной ценностью для существующих или потенциальных клиентов	ГОСТ Р 58537- 2019. УПРАВЛЕНИЕ ПРОДУКЦИЕЙ. Основные положения
Пул команд разработки	Несколько совместно работающих Групп команд разработки, каждая из которых обеспечивает разработку отдельных функциональных подсистем Системы	Термин определен настоящим документом
Рабочая группа по созданию и развитию Системы	Группа, включающая должностных лиц Ведомства и привлекаемых Ведомством исполнителей, обеспечивающая создание, развитие и эксплуатацию Системы на всех этапах ее жизненного цикла	Термин определен настоящим документом
Результат оказания государственной услуги	Сведения в электронной форме, формируемые по результатам оказания государственных услуг,	Постановление Правительства Российской Федерации от 26 марта 2016 г. № 236 «О

	внесенные в государственные и муниципальные информационные системы	требованиях к предоставлению в электронной форме государственных и муниципальных услуг»
Релиз	Набор новых и/или измененных элементов конфигурации, которые были совместно протестированы и внедрены в рабочую среду	1. ГОСТ Р ИСО/МЭК 19770-1-2014 Информационные технологии. Менеджмент программных активов. Часть 1. Процессы и оценка соответствия по уровням
Риск	<p>Следствие влияния неопределенности на достижение поставленных целей.</p> <p>Риск часто характеризуют путем описания возможного события и его последствий или их сочетания.</p> <p>Под следствием влияния неопределенности необходимо понимать отклонение от ожидаемого результата или события (позитивное и/или негативное).</p>	ГОСТ Р 57306-2016 Инжиниринг. Терминология и основные понятия в области инжиниринга

	Цели могут быть различными по содержанию (в области экономики, здоровья, экологии и т.п.) и назначению (стратегические, общеорганизационные, относящиеся к разработке проекта, конкретной продукции и процессу).	
Синергетический эффект	Повышение результативности и эффективности деятельности в результате соединения, интеграции или слияния отдельных частей в единую систему.	ГОСТ 57 522 - 2017 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО Руководство по интегрированной системе менеджмента качества и бережливого производства
Системная команда	Команда для решения специальных задач по созданию инфраструктуры разработки, включающей необходимые программно-аппаратные средства для интеграции разработанной функциональности в составе Системы, проведения всех видов тестирования, демонстрации Систем и решений, развертывания новых версий Системы, а также предоставления доступа к инфраструктуре разработки для Рабочей группы	Термин определен настоящим документом
Системный подход	Методология изучения (рассмотрения) системы как целостного комплекса взаимосвязанных и	Садовский В. Н. Системный подход и общая теория систем: статус, основные проблемы и

	взаимодействующих элементов для достижения поставленных целей	перспективы развития. — М.: Наука, 1980.
Стадия создания (автоматизированной системы, АС)	Одна из частей процесса создания АС, установленная нормативными документами и заканчивающаяся выпуском документации на АС, содержащей описание полной, в рамках заданных требований, модели АС на заданном для данной стадии уровне, или изготовлением несерийных компонентов АС, или приемкой АС в промышленную эксплуатацию	ГОСТ Р59853 - 2021 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы
Управление потоком создания ценности	Деятельность по планированию, реализации, проверке и улучшению ПСЦ как системы процессов, направленную на удовлетворение требований потребителей и других заинтересованных сторон.	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь
Управление продуктами	Управление продуктами отвечает за определение и поддержку создания желательных, выполнимых, жизнеспособных и устойчивых продуктов, которые удовлетворяют потребности клиентов в течение жизненного цикла продукта на рынке.	ГОСТ Р 58537- 2019. УПРАВЛЕНИЕ ПРОДУКЦИЕЙ. Основные положения
Функциональность	Способность продукта или услуги обеспечивать функции, которые соответствуют установленным и	ГОСТ Р 57306-2016

	предполагаемым потребностям, при их использовании в определенных условиях	Инжиниринг. Терминология и основные понятия в области инжиниринга
Функция (автоматизированной системы)	Совокупность действий, направленная на достижение определенной цели	ГОСТ Р 59853 - 2021 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения
Функция федеральных органов исполнительной власти	Нормативно установленный вид властной деятельности указанного органа государства, постоянно осуществляемый им в масштабах Российской Федерации. Функций федеральных органов исполнительной власти: а) функции по принятию нормативных правовых актов; б) функции по контролю и надзору; г) функции по управлению государственным имуществом; д) функции по оказанию государственных услуг	Указ Президента РФ от 9 марта 2004 г. № 314 «О системе и структуре федеральных органов исполнительной власти»
Ценность (value)	Полезность, присущая продукции с точки зрения потребителя и находящая отражение в цене продаж и рыночном спросе.	ГОСТ Р 56020-2020 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО. Основные положения и словарь

<p>Цикл "Планируй - Делай - Проверь - Действуй" (PDCA: Plan - Do - Check - Act)</p>	<p>Цикл, позволяющий организации обеспечивать ее процессы необходимыми ресурсами, осуществлять их менеджмент, определять и реализовывать возможности для улучшения.</p> <p>Цикл PDCA может быть применен ко всем процессам и к системе менеджмента в целом.</p> <p>Описание цикла PDCA:</p> <ul style="list-style-type: none"> - планируй - разработка целей системы и ее процессов, определение ресурсов, необходимых для достижения результатов в соответствии с требованиями потребителей и политикой организации, определение и рассмотрение рисков и возможностей; - делай - выполнение того, что запланировано; - проверь - мониторинг и (там, где это применимо) измерение процессов, продукции в сравнении с политикой, целями, требованиями и запланированными действиями и сообщение о результатах; - действуй - принятие мер по улучшению результатов деятельности в той степени, насколько это необходимо. 	<p>ГОСТ Р 57524-2017 БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО Поток создания ценности</p>
<p>Цифровой продукт</p>	<p>Отдельная программа для ЭВМ (приложение) для выполнения некоего конечного процесса</p>	<p>Приказ Минкомсвязи РФ от 1 августа 2019 г. № 428 «Об утверждении Разъяснений (методических рекомендаций) по разработке региональных</p>

		проектов в рамках федеральных проектов национальной программы «Цифровая экономика Российской Федерации»
Цифровой сервис	Комплексное решение на базе цифровых продуктов, направленное на значимое качественное улучшение или ускорение процессов жизнедеятельности, организационных или бизнес-процессов, в том числе производственных процессов	Приказ Минкомсвязи РФ от 1 августа 2019 г. № 428 «Об утверждении Разъяснений (методических рекомендаций) по разработке региональных проектов в рамках федеральных проектов национальной программы «Цифровая экономика Российской Федерации»
Эмерджентность	Эмерджентность в теории систем — свойство систем, обуславливающее появление новых свойств и качеств, не присущих элементам, входящим в состав системы	ГОСТ Р 43.0.30-2022 Информационное обеспечение техники и операторской деятельности. Системность.
Этап создания (автоматизированной системы, АС)	Часть стадии создания АС, выделенная по соображениям единства характера работ и (или) завершающего результата, и (или) специализации исполнителей	ГОСТ Р 59853 - 2021 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы

ОБЯЗАННОСТИ РАБОЧЕЙ ГРУППЫ СОЗДАНИЯ И РАЗВИТИЯ ГОСУДАРСТВЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

1. ВЛАДЕЛЕЦ СИСТЕМЫ

Владелец Системы — должностное лицо Ведомства, которое несет основную, в том числе финансовую, ответственность за результаты создания (развития) Системы, за надзор и соответствие Системы нормативно-правовым актам, за ввод в эксплуатацию и эксплуатацию Системы в течение всего ее жизненного цикла.

Владелец Системы назначается внутренним распорядительным документом Ведомства (приказом).

Владельцу Системы непосредственно подчинена Группа управления Системой.

Владелец Системы:

- 32) осуществляет управление созданием и развитием Системы;
- 33) согласовывает основные положения Концепции создания (развития) Системы, утверждает или представляет на утверждение вышестоящим должностным лицам Концепцию;
- 34) организывает разработку Технического задания (далее – ТЗ) на создание (развитие) Системы, содержащее Группы требований, утверждает ТЗ или представляет его на утверждение вышестоящим должностным лицам;
- 35) организывает разработку План-графика итерационной разработки по их реализации, утверждают План-график итерационной разработки;
- 36) формируют состав (назначают, при наличии полномочий) Группы управления Системой;
- 37) организывает привлечение исполнителей либо внутри Ведомства, либо внешних - на контрактной или иной основе в соответствии законодательством российской Федерации;
- 38) организывает подготовку и непосредственно участвуют в планировании Инкрементов, определяют ценность командных целей и утверждают окончательных план Команд разработки на Инкремент;

39) принимает участие в мероприятиях по синхронизации Команд разработки;

40) принимает участие в подведения итогов Инкремента (ретроспективы), оценивают фактически достигнутый результат, организывает корректировку План-графика итеративной разработки на следующий Инкремент с учетом выявленных проблем;

41) контролирует ежеквартальный выпуск релиза Системы, включая проведение демонстрации системы для оценки прогресса и предоставления Командам разработки обратной связи, предварительных испытаний, опытной эксплуатации, согласовывают (утверждают/представляют на утверждение) соответствующие документы;

42) организывает приемочное тестирование;

43) принимает решение о вводе в эксплуатацию релиза, согласовывают (утверждают/представляют на утверждение) соответствующие документы;

44) при необходимости организует взаимодействие с другими Ведомствами (например, ФСТЭК, ФСБ) и вышестоящими Органами власти;

45) принимает решения по решению возникающих проблем создания (развития) Системы.

2. ГРУППА ОТВЕТСТВЕННЫХ ЗА СИСТЕМУ

Группа ответственных за Систему – должностные лица и специалисты Ведомства, отвечающие за реализацию требований к Системе, соответствующих потребностям конечных пользователей Системы (потребителей государственных услуг), зафиксированных в ТЗ на Систему.

Группа ответственных за Систему контролирует работу Исполнителя, а также иных специалистов и команд, привлечение которых необходимо для выполнения работ по созданию (развитию) Системы.

Состав Группы ответственных за Систему утверждается внутренним распорядительным документом Ведомства (приказом) и, в общем случае, включает руководителя группы, его заместителя, должностных лиц различных подразделений Ведомства, в том числе отвечающих за предоставление требуемой функциональности Системы клиентам, информационной безопасности и организации эксплуатации.

При выполнении работ на расширенном уровне Группа ответственных за Систему, включает Ответственных за каждую из подсистем, назначаемых из

должностных лиц Ведомства и отвечающих за предоставление клиентам требуемой функциональности соответствующей подсистемы.

Группа ответственных за Систему (Ответственные за подсистему – в части касающейся):

46) принимает участие в разработке проектов документов планирования создания и развития Системы, а также проводит обоснование необходимости и целесообразности ее создания (развития);

47) разрабатывает основные положения Концепции создания (развития) Системы, согласовывают ее с должностными лицами Ведомства и, при необходимости - другими Ведомствами;

48) разрабатывает Техническое задание на создание (развитие) Системы, включая Группы требований, согласовывают его с должностными лицами Ведомства и другими Ведомствами;

49) разрабатывает План-график итеративной разработки, согласовывает его с должностными лицами Ведомства и, при необходимости - другими Ведомствами, после чего представляет План-график на утверждение Владельцу Системы;

50) разрабатывает документы необходимые для привлечения исполнителей на контрактной или иной основе (в соответствии с законодательством Российской Федерации);

51) организывает совместно с Менеджером процессов Группы команд подготовку и проведение планирования Инкремента, координируют совместно с Ответственными за реализацию и Менеджером процессов Группы команд взаимодействие Команд разработки в ходе планирования;

52) участвует в мероприятиях по синхронизации Команд разработки;

53) организывает совместно с Менеджером процессов Группы команд и Ответственными за реализацию подведение итогов Инкремента, оценивает под руководством Владельца Системы фактически достигнутый результат, готовит и представляет Владельцу Системы предложения по корректировке План-графика итеративной разработки на следующий Инкремент с учетом выявленных проблем;

54) обеспечивает подготовку и проведение мероприятий выпуска релизов Системы, включая демонстрацию системы, проведение предварительных испытаний, опытной эксплуатации, разрабатывает и представляют на утверждение Владельцу Системы соответствующие документы;

55) участвует в приемке релизов и Системы в целом в эксплуатацию, разрабатывает и представляют на утверждение Владельцу Системы соответствующие документы;

- 56) обеспечивает юридическое сопровождение выполняемых работ,
- 57) организует взаимодействие с поставщиками в интересах обеспечения команд разработки необходимым оборудованием и программным обеспечением;
- 58) участвует при необходимости в совместной работе с другими Ведомствами (МинЦифры, ФСТЭК, ФСБ).

3. ОТВЕСВЕННЫЙ ЗА РЕАЛИЗАЦИЮ

Ответственный за реализацию - член Команды разработки, ответственный за определение приоритетов и управление реализацией функциональности Системы посредством итераций при сохранении концептуальной и технической целостности компонентов Системы и достижении целей Инкремента.

Ответственный за реализацию имеет следующие обязанности.

При подготовке к планированию Инкремента Ответственный за реализацию:

59) участвует в формировании и уточнении задач Команды разработки на предстоящий Инкремент, а также в определении критериев достижения целей команды в ходе Инкремента;

60) разрабатывает предложения по внесению изменений в План-график итеративной разработки.

Во время мероприятия планирования итерации Ответственный за реализацию:

61) участвует в определении планируемых к реализации функций;

62) формирует детали реализации функций, таким образом, чтобы Команда разработки к концу планирования согласовала окончательный план команды;

63) координирует и согласовывает зависимости между работами Команд с другими Ответственными за реализацию;

64) если Команда разработки в период планирования Инкремента берет на себя риски осуществляет планирование и работу по парированию рисков или уменьшения их влияния.

В рамках выполнения Инкремента Ответственный за реализацию:

65) осуществляет контроль реализации функциональности Системы;

66) обеспечивает скоординированную работу Команды для реализации запланированной функциональности точно в срок — за время Инкремента, с требуемым качеством, уделяя особое внимание в первую очередь пригодности для использования реализованной функциональности;

67) участвует (совместно с Системным архитектором / инженером) в принятии решений по созданию критически важной технологической инфраструктуры, которая будет обеспечивать создание новой функциональности;

68) взаимодействует с Командой разработки, чтобы обеспечить соответствие реализованной функциональности критериям приемлемости в ходе приемочных испытаний и опытной эксплуатации;

69) участвует во всех стандартных мероприятиях команды (планирование итераций, ежедневный стендап, демонстрация Системы, ретроспектива).

Ответственный за реализацию участвует в ретроспективе Инкремента, где Команды разработки собираются для улучшения своих процессов и активно участвуют в семинарах Группы команд разработки в рамках мероприятия «Инспекция и адаптация».

4. МЕНЕДЖЕР ПРОЦЕССОВ ГРУППЫ КОМАНД

Менеджер процессов Группы команд является лидером Группы команд разработки. Он отвечает за эффективность работы Группы команд разработки через соблюдение установленным процессам.

Менеджер процессов Группы команд поддерживает взаимодействие (коммуникации) со всеми заинтересованными сторонами Рабочей группы, решает возникающие проблемы и способствуют постоянному улучшению качества работы Группой команд за счет постоянного совершенствования процессов.

В интересах совершенствования процессов выпуска версий Менеджер процессов Группы команд должен иметь знания и практические навыки в области внедрения бережливых и гибких практик создания Системы, а также возможных вариантов масштабирования разработки Системы, в том числе, с учетом рекомендаций настоящего документа.

При выполнении работ на расширенном уровне назначается Менеджер процессов Пула разработки, координирующий действия Менеджеров процессов группы команд, выполняющих создание отдельных подсистем.

Менеджер процессов Группы команд разработки (пула команд разработки) обязан:

70) улучшать процессы создания Системы с помощью различных инструментов гибкого и бережливого управления, принятых в организации, ведущей разработку продукта (решения);

71) обеспечивать стандартизацию и документирование процессов выпуска версий Системы;

72) участвовать в подготовке и планировании Инкремента;

73) содействовать периодической синхронизации действий и результатов всех команд, входящих в состав Группы команд;

74) организовывать обучение Команд разработки методологии гибкого и бережливого создания продуктов, в том числе – в соответствии с рекомендациями настоящего документа;

75) оказывать помощь Командам разработки в обеспечении взаимодействия;

76) выявлять проблемы взаимодействия Команд разработки и эскалировать их при необходимости на уровень Группы ответственных за Систему и Владельца Системы;

77) предоставлять Группе ответственных за Систему информацию о необходимых ресурсах для реагирования на риски и устранения возникших проблем взаимодействия Команд разработки;

78) участвовать в демонстрации Системы с целью минимизации рисков несогласованной работы Команд разработки;

79) обеспечивать улучшение процессов работы Группы команд разработки с помощью мероприятий по проверке их соответствия установленным требованиям к процессам и внедрения (адаптации) улучшений процессов;

80) оценивать уровень соответствия команд рекомендованным процессам подготовки и выпуска версий и содействовать улучшению процессов.

Менеджер процессов Группы команд должен обладать лидерскими организационными навыками для выполнения своей роли:

81) проводить мониторинг социальных аспектов Команд разработки;

82) слушать и поддерживать команды в выявлении проблем и принятии решений;

83) быть открытым и ценить открытость в других;

84) создавать среду взаимного влияния;

85) понимать и сопереживать другим;

86) поощрять и поддерживать личностное развитие каждого сотрудника и развитие команд;

87) формировать у членов команд активную позицию и инициативу в обсуждении рабочих вопросов и (децентрализованному) принятию решений;

88) применять проактивное и системное мышление;

89) поддерживать обязательства команд.

5. МЕНЕДЖЕР ПРОЦЕССОВ КОМАНДЫ РАЗРАБОТКИ

Менеджер процессов команды разработки — это лидер Команды разработки, отвечающий за эффективность ее работы через выполнение установленных требований к процессу разработки. Эту роль, как правило, берет на себя член Команды разработки. Менеджер процессов команды тратит большую часть своего рабочего времени на поддержание согласованных процессов, помогая другим членам Команды разработки взаимодействовать и сотрудничать. Менеджер команды также помогает команде координировать свои действия с другими командами в группе команд.

Менеджер процессов команды обязан:

90) поддерживать принципы организации процессов итеративной разработки Системы;

91) способствовать прогрессу Команды разработки в достижении целей команды через повышение качества и производительности выполнения работ, а также предсказуемости и согласованности;

92) оказывать помощь Команде разработки в планировании и проведении итерации команды и достижении ежедневных целей в контексте создания Системы в целом;

93) участвовать совместно с Ответственным за реализацию в организации и проведении демонстрации результатов каждой итерации при участии всех заинтересованных лиц,

94) организовывать проведение ретроспективы при участии всех членов Команды разработки;

95) организовывать и проводить мероприятия, в том числе (где это применимо) ежедневное совещание на ногах, планирование, обзор итераций и ретроспективу итераций;

96) внедрять практику качества, в соответствии с настоящим документом, предоставлять рекомендации по улучшению качества результатов;

97) помогать развивать культуру технической дисциплины, которая является отличительной чертой эффективных команд гибкой разработки;

98) помогать команде совершенствоваться и брать на себя ответственность за свои действия;

99) помогать команде разрешать межличностные конфликты и проблемы, а также выявлять возможности для роста;

100) помогать координировать межкомандное сотрудничество в Группе команд, постоянно улучшают коммуникации и отношения с другими командами;

101) помогать команде строить эффективные отношения с системной командой, архитекторами/инженерами и общими службами;

102) взаимодействовать с Менеджерами процессов других команд, чтобы помочь Команде разработки внести свой вклад в создание и развитие Системы;

103) участвовать в оценке результатов итераций и разрабатывать рекомендации по улучшению работы на последующих итерациях;

104) проводить мониторинг социальных аспектов команды и поддержание командного духа.

6. АРХИТЕКТОР СИСТЕМЫ

Архитектор Системы - технический специалист (или команда специалистов), который разрабатывает и уточняет проектные (технические) решения, определяет архитектуру Системы и осуществляет архитектурный надзор в процессе создания и развития Системы.

При выполнении работ на расширенном уровне, создается команда архитекторов, включающая архитекторов подсистем под руководством Главного архитектора Системы.

Архитектор Системы (Архитектор подсистемы – в части касающейся) обязан:

105) разрабатывать решения по архитектуре Системы и документацию на Систему, в том числе - техническую документацию в объеме, необходимом для описания полной совокупности проектных решений (в том числе по защите информации) и достаточном для дальнейшего выполнения работ по созданию Системы, а также описания проектных (технических) решений, обеспечивающих реализацию одного или нескольких процессов предоставления государственных услуг, государственных функций, включая контрольно-надзорную деятельность, подлежащих автоматизации и/или цифровой трансформации и реализуемых посредством Системы;

106) уточнять проектные (архитектурные) решения, обеспечивающие реализацию и интеграцию функциональности, реализуемой различными Командами разработки в предстоящей итерации;

107) определять необходимость создания вспомогательных технических и программных средств, необходимых для создания новых функций и возможностей Системы;

108) определять основные технологии, планируемые к использованию при разработке Системы и нефункциональные требования;

109) разрабатывать архитектурные решения для реализации функциональности в предстоящем Инкременте, а также соответствующие изменения в методах разработки и в использовании практик DevSecOps, таких как непрерывная интеграция, непрерывное развертывание и автоматизация тестирования;

110) организовывать разработку и осуществлять архитектурный надзор за разработкой рабочей документации на Систему;

111) проводить изучение альтернативных вариантов архитектурных решений и определять рациональные проектные решения в соответствии с обоснованными критериями;

112) планировать и обеспечивать архитектурную поддержку реализации новых возможностей Системы;

113) разрабатывать и согласовывать технологические подходы к разработке функциональности Системы между Командами разработки, входящих в состав в Группе команд;

114) внедрять и использовать конвейер непрерывной разработки функциональности Системы;

115) принимать участие в подготовке и проведении планировании Инкремента, проведении демонстрации Системы, а также в подготовке релизов Системы и внедрении Системы в эксплуатацию;

116) обеспечивать надзор за качеством разработки, а также внедрения инструментов разработки в командах.