Инфосистемы Джет
www.jet.su

Программное обеспечение «JET GALATEA»

ПРОЦЕССЫ, ОБЕСПЕЧИВАЮЩИЕ ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2019

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

Оглавление

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

1 Процесс управления циклом разработки

При реализации проектов в компании используются общепринятые методологии разработки информационных систем: итерационный процесс разработки – Rational Unified Process (RUP) и гибкие методологии – Agile, SCRUM.

1.1 Гибкие методологии

Такие методологии – альтернатива «тяжеловесным» практикам разработки (например, водопадной модели разработки) с жестким прописыванием фаз, где в документах до буквы прописаны зоны ответственности, состав и свойства каждого артефакта каждого этапа. Для «тяжеловесных» практик любое изменение становится крайне болезненным, если не катастрофическим. Для гибких методологий изменения, напротив, приветствуются (отсюда и название: «гибкие»).

Все гибкие методологии строятся на четырех постулатах (Agile-манифест):

1. Люди и взаимодействие важнее процессов и инструментов.
2. Важно создать эффективную команду, которая будет прекрасно взаимодействовать внутри и снаружи (например, с Заказчиком и/или производителем оборудования) и обеспечивать эффективное выполнение работ, а не следование каждой букве устава или другого проектного документа.
3. Работающий продукт важнее исчерпывающей документации.
4. Работающий продукт более важен, чем документация, но это не значит, что документации не должно быть либо она должна быть не качественной.
5. Сотрудничество с Заказчиком важнее согласования условий контракта.
6. Важно обеспечить удобную и понятную для Заказчика среду исполнения контракта (экосистему), даже если эта экосистема требует дополнительных трудозатрат и не прописана явно в контракте. Например, одним из элементов экосистемы может являться короткий цикл обратной связи, который позволяет оперативно реагировать на замечания и создавать именно тот продукт, который нужен Заказчику, а не тот, который изначально прописан в контракте.
7. Готовность к изменениям важнее следования первоначальному плану.
8. Согласование и жесткое регламентирование плана работ может привести к возникновению ситуации, когда требуются изменения, но они лежат за рамками исходного плана. Agile-подход позволяет вносить изменения без опасений.

Важное дополнение: гибкие методологии не отрицают важность постулатов справа, но считают более ценными постулаты слева.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

С учетом вариативности тематик проектов и использования методов машинного обучения видится, что гибкие методологии позволят обеспечить максимально эффективный путь реализации проекта, основываясь на следующих подходах:

1. Ориентир на результат, а не на процесс:
   - Обязательный цикл обратной связи с Заказчиком по итогам итерации разработки (спринта) и корректировка пути разработки по ее итогам.
   - Любая находящаяся в работе задача должна иметь измеримую бизнес-ценность. Нет бизнес-ценности – не надо разрабатывать.
   - Обеспечение минимального времени выхода функциональности в промышленную среду.
   - Максимально тесное вовлечение Заказчика в процесс разработки – продукт делается для него и под его нужды. Это приведет к улучшению взаимодействия Заказчика и Исполнителя и получению именного того продукта, который нужен Заказчику.
2. Работа в кросс-функциональных, самоорганизующихся и саморазвивающихся командах, нацеленных на максимальную эффективность:
   - Кросс-функциональная команда, в которой есть заранее определенные роли и границы ответственности, но с целью достижения наилучшего результата они изменяются. Таким образом, команда состоит из инженеров, которые вносят свой вклад в общий успех проекта в соответствии со своими способностями и проектной необходимостью. Команда самоорганизуется для выполнения конкретных задач в проекте, что позволяет ей гибко реагировать на любые возможные изменения.
   - Не менеджер, а вся команда берет на себя обязательства, и каждый член команды согласен, что обязательства выполнимы.
   - Все члены команды осознают и разделяют ответственность как за проект / продукт, так и друг за друга. Всегда готовы подстраховать и помочь.
   - Команда, находящаяся в режиме постоянного улучшения, а не застывшая на определенном уровне развития / производительности.
   - Команда, работающая в атмосфере доверия – когда ее члены не боятся признать проблемы и начать их решать, а не запихивают их «под ковер».
   - Команда, работающая в комфортной среде (физически и психологически), что позволяет достигать максимальной эффективности.
   - Команда, не боящаяся ошибаться. Только путем проб и ошибок можно найти оптимальное решение, а бездействие может принести даже больше вреда. Более того,

чем раньше будет совершена ошибка, тем дешевле будет ее исправление (принцип fail fast).

3. Максимальное вовлечение Заказчика в проект на всех фазах создания продукта – только постоянное и эффективное взаимодействие с представителями Заказчика (включая конечных пользователей) способно обеспечить создание жизнеспособного и полезного продукта.
4. Использование механизмов приоритизации – в работу берутся задачи, в порядке снижения приоритета (сначала самые важные). Причём приоритеты определяются только совместно с Заказчиком.
5. Использование метрик эффективности производственного процесса, с тщательным анализом причин, которые привели к изменению значений и разработкой мер по устранению негативных и сохранению позитивных трендов.
6. Получение стабильного, наглядного и управляемого процесса даже в условиях нестабильного окружения.

При реализации проекта и с учетом использования SCRUM обязательно использование следующих, не озвученных ранее, артефактов:

7. Скрам-мастер:
   - Член команды, который помогает правильно использовать SCRUM. Agile-лидер команды.
   - Направляет команду к становлению самоуправляемой и самоорганизующейся.
   - Следит за тем, чтобы команда выполняла принятые решения, соблюдала agile- и иные установленные практики.
   - Ведёт командные встречи/митинги (планирование, демонстрация, ретроспектива).
   - Устраняет проблемы, мешающие команде эффективно работать (самостоятельно или с привлечением управленческого персонала).
   - Следит за климатом и «пульсом» команды, создает атмосферу доверия. При возникновении конфликтов не «гасит» их, а наоборот: вытаскивает наружу и помогает команде найти быстрое и конструктивное их решение.
   - В ряде случаев может и должен приносить дискомфорт членам команды. Например, внося улучшения, которые не очень удобны конкретным членам, но полезны для процесса/проекта. Это позволит не отсеять полезные улучшения, приносящие дискомфорт на начальном этапе.
8. Владелец продукта:

- Единый, ответственный за продукт и список задач продукта сотрудник. Эксперт в предметной области, понимающий, каким должен стать продукт, какие задачи он должен решать, как выглядеть и как работать (с точки зрения бизнеса).
- Единая точка принятия окончательных решений для команды в проекте. Это всегда один человек, а не группа или комитет. Такой подход позволяет избежать проблемы множественности заказчиков, противоречивости и неоднородности требований.
- Управляет ожиданиями заинтересованных лиц и доносит до них причины принятия того или иного решения.
- Четко понимает, какие задачи должны быть сделаны, какую ценность они несут и каковы их приоритеты.
- Ставит задачи всей команде, а не конкретному члену команды.
- Принимает результаты в конце каждой итерации/спринта.
- Не контролирующий орган, требующий выполнения задач в сроки, а полноценный член команды, который берет на себя весомую часть аналитических активностей.

9. Итерация/Спринт:

- Фиксированный промежуток времени, в течение которого реализуется бизнес-ценностная поставка Заказчику. Предлагаемая длительность – две недели.
- В течение спринта команда:
    - собирает и уточняет требования;
    - реализует требования;
    - проверяет реализацию;
    - проводит поставку и демонстрацию Заказчику. Владелец продукта фиксирует предложения по его улучшению и включает их в планы последующих спринтов.
    - Состав задач спринта формируется с учётом приоритетов от владельца продукта и производительности команды (в спринт должны браться только наиболее приоритетные задачи, которые команда способна реализовать).
- Задачи спринта должны реализовываться в спринте, а не переноситься на следующий.
- При обнаружении проблем команда оперативно информирует заинтересованных лиц (обязательно владельца продукта) и вырабатывает решение по их устранению.

10. Список задач продукта:

- Приоритизированный список имеющихся на данный момент бизнес- и технических требований к продукту.
- Состоит из следующих видов проблем/вопросов:

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- эпики (контейнеры для пользовательских историй);
- пользовательские истории;
- запросы на изменения;
- задачи членам команды;
- ошибки;
- запросы на консультации.
- Список постоянно пересматривается, приоритизируется и дополняется – в него включаются новые требования, удаляются ненужные, пересматриваются приоритеты.

11. Список задач спринта:

- Приоритизированный список бизнес- и технических требований к продукту, которые должны быть реализованы в спринте.
- Производительность команды:
    - одна из основных метрик процесса разработки;
    - рассчитывается как сумма оценок задач, реализованных за спринт;
    - при планировании спринта берется усредненное значение по итогам прошедших спринтов;
    - разбросы производительности от спринта к спринту должны тщательно анализироваться, чтобы определить и устранить причины нестабильной работы команды.

12. Планирование спринта:

- Встреча, на которой команда и владелец продукта планируют, что будет реализовано в спринте.
- Владелец продукта ставит цели и предоставляет пользовательские истории, запросы на изменения и дефекты к реализации.
- Команда проводит оценку представленных задач.
- Полученные оценки сопоставляются с производительностью команды в предшествующих спринтах. В спринт включаются только те задачи, которые команда сможет успеть выполнить (хватит производительности). При этом спринт наполняется в чётком соответствии с приоритетами: недопустимо, когда из состава спринта исключаются задачи с приоритетом более высоким, чем включенные в спринт.
- По итогам планирования спринта все члены команды согласны с тем, что спринт можно реализовать.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

13. Доска задач:

- Физическая или виртуальная доска отражающая процесс реализации продукта. Разделена на столбцы, соответствующие статусам реализации задач.
- Доска доступна всем членам команды.
- В общем случае должна состоять минимально из трёх столбцов (**Необходимо выполнить**, **Выполняется**, **Закрыто**), между которыми перемещаются задачи.
- Задача команды – обеспечивать актуальное состояние доски. Например, в случае перевода задачи в работу, ее нужно передвинуть на доске из «Необходимо выполнить» в «Выполняется».
- Для обеспечения максимально эффективного производственного цикла используется следующая модель статусов:
- Backlog (Регистрация) – задача зарегистрирована и включена в список задач продукта;
- To do (Необходимо выполнить) – задача отобрана для реализации в текущем спринте;
- Analysis (Анализ) – над задачей работает аналитик, происходит уточнение требований;
- Ready for development (Готово к реализации) – работа аналитика и согласование требований завершены, issue готова к реализации;
- Development (Разработка) – ведется разработка задачи;
- Ready for review (Готово к проверке) – разработка завершена, решение задачи готово к проверке программного кода;
- Review (Проверка) – ведется проверка кода;
- Resolved (Решено) – решение задачи готово к тестированию;
- QA (Тестирование) – ведется тестирование;
- Reopened (Открыто повторно) – задача повторно открыта, требуется доработка;
- Accepting by customer (Приёмка) – приемка реализации Заказчиком;
- Paused (Приостановлено) – работа с задачей приостановлена;
- Closed (Закрыто) – задача решена в полном объеме.

14. Стендап:

- Не «отчетная встреча перед менеджером/владельцем продукта», а ежедневное оперативное обсуждение прогресса и инструмент дальнейшей координации действий команды с целью достижения результата.
- Следствием Стендапа является:

- улучшение горизонтальных коммуникаций в команде;
- публикация информации по статусу выполнения работ и специфических знаний;
- возможность оперативно вносить корректировки (при необходимости).

- При правильной организации процесса каждому члену команды приходится перед стендапом заниматься микропланированием и план-фактным анализом своей деятельности. Это дисциплинирует и даёт дополнительную практику в оценке трудозатрат.
- Проводит Скрам-мастер. Каждый член команды отвечает на три вопроса:
  - Что сделано вчера – подтверждение выполнения взятых ранее обязательств;
  - Что будет сделано сегодня – взятие новых обязательств;
  - С какими проблемами столкнулся – просьба о помощи у коллег.
- Это не рабочая, а статусная встреча. Поэтому все рабочие вопросы выносятся за её рамки. Это позволит избежать нерационального расходования ресурсов, когда два члена команды обсуждают задачу, а остальные ждут и не участвуют в обсуждении.
- Проводится в одно и то же время и в одном и том же месте.
- Длительность не более 15 минут.
- Все члены команды участвуют от начала и до конца.

15. Демонстрация:

- Все заинтересованные лица приглашаются и приходят на демонстрацию.
- Инструмент получения обратной связи и эффективных коммуникаций с заинтересованными лицами.
- Проводится в конце каждой итерации. Команда показывает результаты своей работы. Например: отчитывается о выполненных задачах и собирает обратную связь.
- Демонстрируются только выполненные задачи.
- По итогам демонстрации владелец продукта корректирует Бэклог продукта в соответствии с полученной обратной связью.

16. Ретроспектива:

- Инструмент улучшения производственного процесса из четырёх составляющих:
  - найти проблемы и выработать пути их устранения (запланировать мероприятия);
  - выполнить мероприятия;
  - проверить, к чему привело выполнение мероприятия (позитив или негатив);
  - понять, что делать с результатами:

- если результаты позитивные, то включить мероприятия в производственную практику, и перейти к поиску иных проблем;
- если результаты негативные, то отказаться от принятых мер и вернуться на фазу поиска других решений проблемы.

- Проводится по итогам спринта.
- Обычно обсуждаются две группы событий, возникших в спринте:
  - позитивные. Если они могут исчезнуть – разрабатываются меры, которые помогут их сохранить;
  - негативные. Если они сами не исчезнут (например, ответственный сотрудник вернется из отпуска, и проблема решится сама собой), то разрабатываются меры, которые их устранят.
- Определяется, кто готов взять на себя реализацию и их сроки. В идеальном случае сотрудник должен сам вызваться выполнить работу. При этом тот, кому досталась та или иная мера, должен быть готов её выполнить, разделить полезность этой меры и приложить максимум усилий для ее выполнения. На следующей ретроспективе происходит синхронизация реализованных мер и их результатов.

17. Критерии готовности: согласованное с Владельцем продукта и заинтересованными сторонами определение готовности задач в виде чек-листа.

Составление критериев готовности позволяет избежать проблем уровня «а мы думали иначе».

Соблюдение критериев готовности позволяет систематизировать производственный процесс в части полноты и качества реализации задачи.

Одним из столпов для создания эффективных продуктов является дисциплина управления требованиями. В противном случае велика вероятность создания продукта, который только частично удовлетворяет Заказчика. Управление требованиями начинается со сбора информации о бизнес-процессах автоматизируемой или оптимизируемой предметной области. При сборе исходной информации работа проводится совместно с сотрудниками Заказчика. Такая коллективная работа предусматривает разнообразные методы сбора информации: анализ документов, анкетирование, интервьюирование, наблюдение, мозговой штурм, совещание.

- Анализ документов – исследуются документы, предоставляемые Заказчиком, и документы из открытых источников.
- Интервью – информация собирается путем очного опроса сотрудников Заказчика. Для проведения интервью мы предварительно составляем подробные списки вопросов. Сами интервью проводятся в несколько раундов для уточнения возникающих в процессе обследования вопросов.

- Анкетирование – заочный опрос экспертов Заказчика, при котором сотрудники заполняют подготовленные нами вопросники. Исполнитель рассматривает анкетирование как вспомогательный метод обследования и использует его опционально. Анкеты помогают подтвердить первоначальное видение характеристик бизнес-процессов.
- Наблюдение – применяется, когда есть несколько толкований полученных сведений и требуется проверка на месте, чтобы выработать однозначное их понимание.
- Мозговой штурм – используется для решения сложных нестандартных задач. Цель такого неформального общения – обсуждение разных вариантов решения проектных задач и принятие коллегиального оптимального решения.
- Совещание – проводится в целях:
    - достижения соглашения в спорных или невыясненных вопросах за предельно короткий промежуток времени;
    - быстрого принятия решение о том, в каком направлении действовать;
    - получения новых предложений;
    - фиксации решения.

При сборе информации Исполнитель анализирует различные аспекты и результаты деятельности компании, включая следующие:

- Численность, организационная структура, подчиненность подразделений.
- Цели и задачи подразделений.
- Функциональность подразделений и типовых рабочих мест специалистов.
- Структура и функциональность информационных систем, недостатки в информационном обеспечении.
- Используемые документы, информация, нормативно-справочная информация, справочники, классификаторы.
- Достаточность информации для принятия решений, перечень недостающей информации.
- Взаимодействие с подразделениями внутри компании, существующие проблемы во взаимодействии с подразделениями.
- Получаемые документы (планы, отчеты, приказы, распоряжения) и входящая информация.
- Формируемые документы и выходящая информация.
- Перечень и характер проблем на языке бизнеса.

- Детали рассматриваемых организационных, производственных, технологических процессов.
- Как решается задача сейчас.
- Какие уже были рассмотрены/ применены модели, методы решения.
- Какие итоги рассмотрения/ применения моделей и методов решения.
- Каким видится идеальное решение.
- Какие требования к решению.
- Какие имеются ограничения.
- Что важно учитывать при математическом моделировании, машинном обучении.
- Предполагаемое дальнейшее организационное развитие подразделений.
- Сведения по имеющимся данным:
- Имеющиеся наборы данных, полнота, корректность.
- За какой исторический период доступны данные.
- Имеются ли существенные изменения записи/ хранения за рассматриваемый период.
- Какие данные считать выбросами.
- Какие признаки процесса являются наиболее значимыми с точки зрения экспертов области. Какие являются менее значимыми.
- Способы обогащения и актуализации данных.
- Метрики качества моделей:
    - какими будут бизнес-метрики качества;
    - какие измерения бизнес-метрик значимы для бизнеса.
    - правила/ методики оценки эффективности моделей (включая экономический эффект).

При описании бизнес- / технологических/ производственных процессов (далее – Процессов) применяется системный подход к их формализации. Это означает, что каждый Процесс рассматривается в аспекте влияния на него внутренних процессов компании (взаимосвязанных и взаимодействующих) и внешнего окружения.

Такая предпосылка определяет основные правила, которыми Исполнитель руководствуется при проведении системного анализа:

- рассмотрение организации как бизнес-системы;
- декомпозиция организации на ее составляющие части (подсистемы, элементы);

- выбор и использование наиболее подходящих специальных методов для решения отдельных задач;
- объединение частных решений, направленных на достижение основной цели анализа.

Обычно процесс системного анализа мы разбиваем на следующие этапы:

18. Систематика информации о направлениях деятельности и общей структуре организации.
19. Анализ внешней среды организации:
    - классификация выявленных проблем, связанных с влиянием внешней среды;
    - определение тенденций и динамики внешних факторов.
20. Анализ деловой среды организации:
    - к деловой среде организации относятся также производственные процессы, технологические, производственные и организационные ограничения;
    - анализ схемы деятельности;
    - систематизация выявленных проблем;
    - определение тенденций воздействия факторов внешней среды.
21. Анализ и формализованное описание структуры организации:
    - анализ организационной структуры;
    - анализ информационной инфраструктуры;
    - выделение основных и вспомогательных процессов для детализированного анализа;
    - определение основных направлений взаимодействия.
22. Формализация существующего положения организации:
    - формализация процессов управления и информационных потоков;
    - анализ процессов на соответствие целям и задачам организации;
    - систематизация выявленных проблем в процессах и информационном обеспечении;
    - анализ причинно-следственных связей внутренних проблем с проблемами взаимодействия с внешним окружением.
23. Анализ стратегии и целей организации.

При обследовании и анализе Процессов применяется системно-аналитический подход, который регламентирует выполнение следующих работ:

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- выявление всех аспектов обследуемых Процессов;
- исследование каждого аспекта принятыми методами анализа;
- унификация полученных результатов обследования;
- обработка результатов и выработка рекомендаций по оптимизации и автоматизации Процессов.

Такая методика предусматривает проведение следующих работ:

- сбор информации об обследуемых бизнес-процессах;
- формализация и структурирование полученной информации;
- документирование имеющейся информации;
- подготовка рекомендаций по оптимизации и автоматизации бизнес-процессов.

При системно-аналитическом обследовании Процессы проходят идентификацию и декомпозицию. При этом аналитики:

- выделяют Процесс из других процессов;
- декомпозируют Процесс на этапы;
- декомпозируют этапы Процесса на локальные бизнес-функции;
- описывают локальные функции с точки зрения их организационно-ролевого, информационного и событийного окружения.

Сама технология системно-аналитического обследования Процессов регламентируется принятыми методиками сбора информации о бизнес-процессах и моделирования бизнеса.

Обследованию Процесса предшествует его выделение из множества других процессов компании. К характеристикам Процесса отнесены:

- входы и выходы Процесса;
- поставщики Процесса;
- клиенты Процесса;
- владелец Процесса;
- взаимодействие (соприкосновение) Процесса с другими Процессами (внешними и внутренними);
- качественные и/или количественные характеристики, определяющие качество деятельности (производительность, продолжительность, результативность и пр.);
- функциональная / предметная область;

- организационные, технологические и производственные ограничения Процесса;

Каждый выделенный для обследования Процесс документально оформляется и согласовывается с Заказчиком.

При декомпозиции Процесс разбивается на этапы – подпроцессы (включая процедуры, функции и работы). Для каждого подпроцесса определяются:

- входы и выходы;
- владелец;
- ресурсы, обеспечивающие подпроцесс;
- регламенты, нормирующие подпроцесс.

При этом проводится дифференциация основных и вспомогательных Процессов. Декомпозиция подпроцесса на локальные функции выполняется аналогично декомпозиции Процесса на его этапы.

Этап моделирования наступает после анализа, структурирования и формализации информации, собранной в процессе обследования.

Моделирование позволяет ответить на следующие вопросы:

- кто выполняет те или иные функции;
- как взаимосвязаны функции;
- какими ресурсами поддерживаются функции;
- каким законам/ правилам подчиняются функции;
- как регламентировано управление функциями;
- как регламентированы связи между исполнителями взаимосвязанных функций;
- какие возможны корреляции между функциями/ данными.

Моделирование позволяет получить интегрированное однозначное представление о Процессах.

При моделировании выполняется:

- детализация Процессов до требуемого уровня;
- привязка выполняемых бизнес- и технологических функций к выполняющим их организационно-ролевым единицам;
- привязка информационных объектов (исходящих, входящих и внутренних документов, информационных потоков и пр.) к бизнес- и технологическим функциям и операциям, в которых они появляются, используются или уничтожаются;

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- привязка выполняемых бизнес- и технологических функций к поддерживающим их информационным системам и оборудованию;
- формализованное описание Процессов в графическом формате;
- математическое моделирование и проверка корректности оптимизационных гипотез.

В процессе моделирования составляется библиотека взаимосвязанных моделей, которые описывают функциональную структуру бизнес- и технологических процессов, ее организационное, ролевое и информационное окружение.

Целью создания такой библиотеки является формирование единой базы, которая отображается в едином стандарте и содержит следующую информацию:

- выполняемые Процессы;
- функции каждого Процесса;
- действия персонала при выполнении функций Процесса;
- разграничение функций Процесса между исполнителями;
- разграничение функций Процесса между подразделениями;
- документы, регламентирующие Процесс;
- программное и аппаратное обеспечение Процесса;
- Модели, обеспечивающие оптимизацию процесса (в том числе прогнозные и рекомендательные).

При графическом описании Процессов обычно применяются стандартные нотации, такие как IDEF и ARIS eEPC.

Сначала строится модель «как есть». Она отражает существующее положение Процесса со всеми его недостатками и достоинствами.

При моделировании ситуации «как есть» применяется следующий подход: целесообразно выделять проблемные зоны и описывать только их. Таким образом, целью анализа «как есть» является создание понятного непротиворечивого списка слабых мест и определение возможностей по оптимизации затронутых процессов.

В модели «как есть» отражаются:

- имя Процесса;
- цель/предмет Процесса;
- входы Процесса;
- выходы Процесса;

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- владелец Процесса;
- используемые прикладные системы;
- управляющие воздействия;
- ресурсы, обеспечивающие Процесс;
- документация по процессу и ее актуальность;
- функции, выполняемые Процессом;
- частота выполнения Процесса и частота возникновения ошибок в Процессе;
- показатели эффективности существующего Процесса.

При составлении модели «как есть» Исполнитель максимально взаимодействует со специалистами Заказчика. Практика показывает, что проще всего внедряются те решения, которые были выработаны совместно. В свою очередь, сотрудники Заказчика обретают знания и компетенции в части технологий описания, оптимизации и регламентации бизнес-процессов. В дальнейшем это позволяет специалистам Заказчика самостоятельно поддерживать документацию Процессов в актуальном состоянии.

Анализ выявленных недостатков является основой для формирования предложений по оптимизации существующего положения. Эти предложения оформляются в соответствии с требованиями Заказчика и обязательно иллюстрируется моделью «как должно быть» и каким образом «как должно быть» будет достигаться.

Задача моделирования «как должно быть» заключается в конкретизации довольно абстрактных первоначальных представлений сотрудника Исполнителя о том, как будет оптимизирован Процесс. Отправной точкой являются первичные цели оптимизации, на базе которых формулируются цели использования моделей, а также требования к ним.

В практике Исполнителя используются два метода моделирования: нисходящее (в направлении сверху вниз) и восходящее (в направлении «снизу вверх»). Как часто бывает, оба метода имеют свои преимущества и недостатки.

Моделирование «сверху вниз» применяется, когда во главу угла поставлены новые цели бизнеса Заказчика в целом. В этом случае бизнес-процесс самого верхнего уровня детализируется до бизнес-процессов нижнего уровня. При этом каждый шаг детализации (декомпозиции) в модели представляет собой функцию бизнес-процесса более низкого уровня. Преимущество этого метода – точное соответствие модели требованиям новой стратегии бизнеса, ведь именно эта стратегия определила модель самого верхнего уровня. Недостаток – возможные упущения при определении глубины декомпозиции и потери связей между процессами нижних уровней

Моделирование «снизу вверх» применяется, когда очевидна каждая бизнес-операция, но для достижения поставленной цели надо оптимизировать всю их совокупность (например, для снижения временных затрат или затрат на ресурсы и т. п.). Преимущество этого метода – созданную модель легко воплотить в жизнь, так как она будет учитывать все взаимосвязанные процессы. Недостаток – большие временны́е затраты на моделирование из-за большого количества бизнес-операций нижнего уровня.

Независимо от выбранной степени детализации, созданные модели «как должно быть» в обязательном порядке определяют те же параметры, что и модель «как есть» (имя процесса, его цель и т. п.). Ввиду того, что чаще всего реализуют проекты по разработке или модернизации информационных систем, модели «как должно быть» традиционно отражают изменения процессов, которые происходят за счёт внедрения системы автоматизации.

Построение модели «как должно быть» является логическим завершением проекта под общим названием «Обследование бизнес-процессов». Модель наглядно показывает, как будет работать компания Заказчика, если она примет решение реорганизовать неэффективные методы ведения бизнеса.

1.1.1 Методологии, стандарты и подходы как инструменты достижения целей

Достижение целей Заказчика обеспечивается за счёт:

1. Комплекса факторов «зашитых» в SCRUM по умолчанию, основными из которых являются:

- Активное вовлечение Заказчика в разработку продукта (в том числе за счёт циклов обратной связи). Позволяет избежать проблем, когда продукт формально соответствует исходным требованиям, но пользоваться им неудобно, сложно и т. д.
- Создание прозрачной и комфортной для Заказчика среды реализации продукта, позволяющей:
    - вносить изменения и уточнения в требования к целевому продукту;
    - активно участвовать Заказчику в поиске и выборе оптимальных методов решения поставленных задач;
    - иметь высокую степень контроля проекта/продукта. В любой момент времени Заказчик понимает, какие задачи и каким образом решает Исполнитель, какие имеются трудности и т. п.
- Тщательный анализ поступающей информации не только с точки зрения «что нужно сделать», но и с точки зрения «а какую ценность это принесет».
- Нацеленность команды на результат интересный Заказчику, а не формальное выполнение обязательств.

- Нацеленность команды на максимальную производительность, а значит и возможность сделать больше, чем того требуют формальные документы.

2. Использование дополнительных артефактов/ принципов, с помощью которых можно упростить взаимодействие и реализацию проекта:

- Система учета задач – Atlassian Jira, позволяющая:
  - гибко настраивать производственные процессы и встраивать в них представителей Заказчика (например, за счёт статуса «Accepting by Customer/Приемка Заказчиком»);
  - использовать автоматически обновляемую виртуальную доску задач, доступ к которой будут иметь и представители Заказчика;
  - упростить процедуры подготовки поставок и выпуска релизов за счёт интеграции с репозиториями кода и инструментами непрерывной интеграции и непрерывного развёртывания (CI/CD).
- Wiki-система – Atlassian Confluence, позволяющая:
  - хранить все текстовые и визуальные артефакты в одном месте;
  - обеспечивать одновременную работу пользователей над артефактами с минимизацией конфликтов слияний.
- Решение для контроля качества – Adaptavist's Test Management for Jira позволяющее:
  - создать и поддерживать в актуальном состоянии модели тестирования продукта;
  - фиксировать результаты прохождения различных видов тестов.
- Использование шаблона описания пользовательских историй и запросов на изменение вида.
- Постоянный мониторинг требований на предмет соответствия критериям качества:

Критерий качества	Пояснение
Единичность	Требование описывает одну и только одну реализуемую сущность
Завершённость	Требование полностью определено в одном месте, вся необходимая информация присутствует
Последовательность/ Непротиворечивость	Требование не противоречит другим требованиям и полностью соответствует внешней документации
Атомарность	Требование «атомарно». Не может быть разбито на ряд более детальных требований без потери завершённости
Отслеживаемость	Требование полностью или частично соответствует деловым нуждам, как заявлено заинтересованными лицами, и задокументировано
Актуальность	Требование не стало устаревшим с течением времени
Выполнимость /достижимость	Требование может быть реализовано в пределах проектных работ
Недвусмысленность	Требование определено без использования технического жаргона, акронимов и других скрытых формулировок. Оно выражает

Критерий качества	Пояснение
	объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация. Определение не содержит нечётких фраз. Использование отрицательных утверждений и составных утверждений запрещено
Обязательность	Требование представляет определённую заинтересованным лицом характеристику, отсутствие которой приведёт к неполноценности решения, которая не может быть проигнорирована. Необязательное требование – противоречие самому понятию требования
Измеримость	Требование может быть измерено количественными либо качественными показателями

- Максимально глубокое погружение всех специалистов Исполнителя в предметную область и проблематику Заказчика, позволяющее обеспечить многоступенчатый контроль не только с точки зрения функционального качества, но и с точки зрения бизнес-потребностей.
- Привлечение к реализации проектов специалистов с опытом и знанием предметной области, позволяющее не только избежать типовые ошибки, не видимые при отсутствии опыта, но и существенно упростить коммуникации с представителями Заказчика.
- Использование машинного интеллекта, которое позволит с получить больший эффект при меньших трудозатратах по направлениям:
    - эффективность мониторинга и автокоррекция производственных процессов;
    - оптимизация цепочек поставок и производства;
    - увеличение степени автоматизации в производственных процессах;
    - интеллектуальная автоматизация планирования за счёт прогнозных и рекомендательных моделей;
    - сокращение времени принятия решений;
    - снижение количества задержек, дефектов и отклонений от спецификаций.
- При решении задач искусственного интеллекта специалисты Исполнителя тщательно подходят к выбору алгоритмов и методов решения задач и выбирают только наилучшим образом работающее решение. Как минимум анализируется возможность применения:
    - Алгоритмов, основанных на решающих деревьях:
        - Различных типов линейных регрессий.
        - Опорных векторов (SVM).
        - Байесовской оптимизации.
    - Нейросетевых алгоритмов (свёрточные (RESNET, VGG16, UNET) и рекуррентных нейросетей (LSTM, RNN), вариационных автоэнкодеров.

- Коллаборативной фильтрации.
- Алгоритмов кластеризации:
  - K-means;
  - DB-Scan и пр.

1.1.2 Успешное внедрение проектных решений и препятствующие факторы

Основными препятствиями для внедрения искусственного интеллекта являются:

- Недостаточные или разрозненные данные. Информация может быть «раскидана» по разным приложениям в разных форматах, включая форматы фотографий и видео. В итоге имеются проблемы со сбором, агрегацией и качеством данных.
- Сложные процессы анализа и прогнозирования. Искусственный интеллект позволяет делать прогнозы, но далеко не всегда может объяснить, как он пришел к конкретным выводам. Это может вызывать отторжение у лиц, принимающих решение.
- Оторванность «производства» от «ИТ-персонала». Недостаточный уровень навыков «производства» в области компьютерной науки, приводит к нежеланию работать с чем-то новым (зачем, если и старое в целом устраивает). Более того, сотрудники часто воспринимают инновации как попытки заменить их машинами.
- Отсутствие ключевого драйвера (владельца продукта) имеющего влияние как на «производство», так и на «ИТ-персонал». При отсутствии у владельца продукта влияния на все задействованные в цепочке автоматизации подразделения существует высокая вероятность получить продукт, который работает формально и несёт минимальную ценность.

Решение этих проблем – крайне важная задача, которую Исполнитель зачастую самостоятельно решить не может. В таких случаях требуется активное участие Заказчика. Например:

- приведение данных к формату, который может использован для обучения моделей искусственного интеллекта (включая обогащение и агрегацию);
- повышение уровня лояльности «производства» к изменениям (например, за счёт повышения уровня компьютерной грамотности и указания перспектив, которые несут изменения);
- формирование проактивной команды Заказчика, поддерживаемой вплоть до уровня Директоров;
- включение в команду Заказчика не только «производства», но «ИТ-специалистов» – владельцев целевых систем;
- детальная проработка критериев успешности проекта на его старте.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

1.1.3 Подход к обеспечению качества и его применение

При разработке программных продуктов АО «Инфосистемы Джет» руководствуется следующими целями:

1. Организация полноценного и всеобъемлющего контроля качества разрабатываемых программных продуктов.
2. Обеспечение полноты тестирования на соответствие требованиям к продукту.
3. Добавление управляемости и прозрачности в процессы разработки и тестирования.
4. Минимизация сроков выдачи изменений/ исправлений.

На диаграмме (Рисунок 1) крупными блоками представлены основные вехи работы команды от момента формулирования требований и до исправления дефектов в промышленной среде:

1. Формулировка бизнес-задачи.
2. Подготовка бизнес-требований.
3. Технический анализ.
4. Планирование релиза.
5. Тестирование «идеи» или ревью технических требований к продукту.
6. Разработка.
7. Настройка CI/CD.
8. Сборка.
9. Отладка в среде разработки.
10. Тестирование.
11. Приёмочное тестирование.
12. Развёртывание продукта на различные среды.
13. Вывод продукта в продуктивную (промышленную) среду.
14. Исправление дефектов, найденных на промышленном образце и постанализ для дальнейшего исправления пропущенной проблемы.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

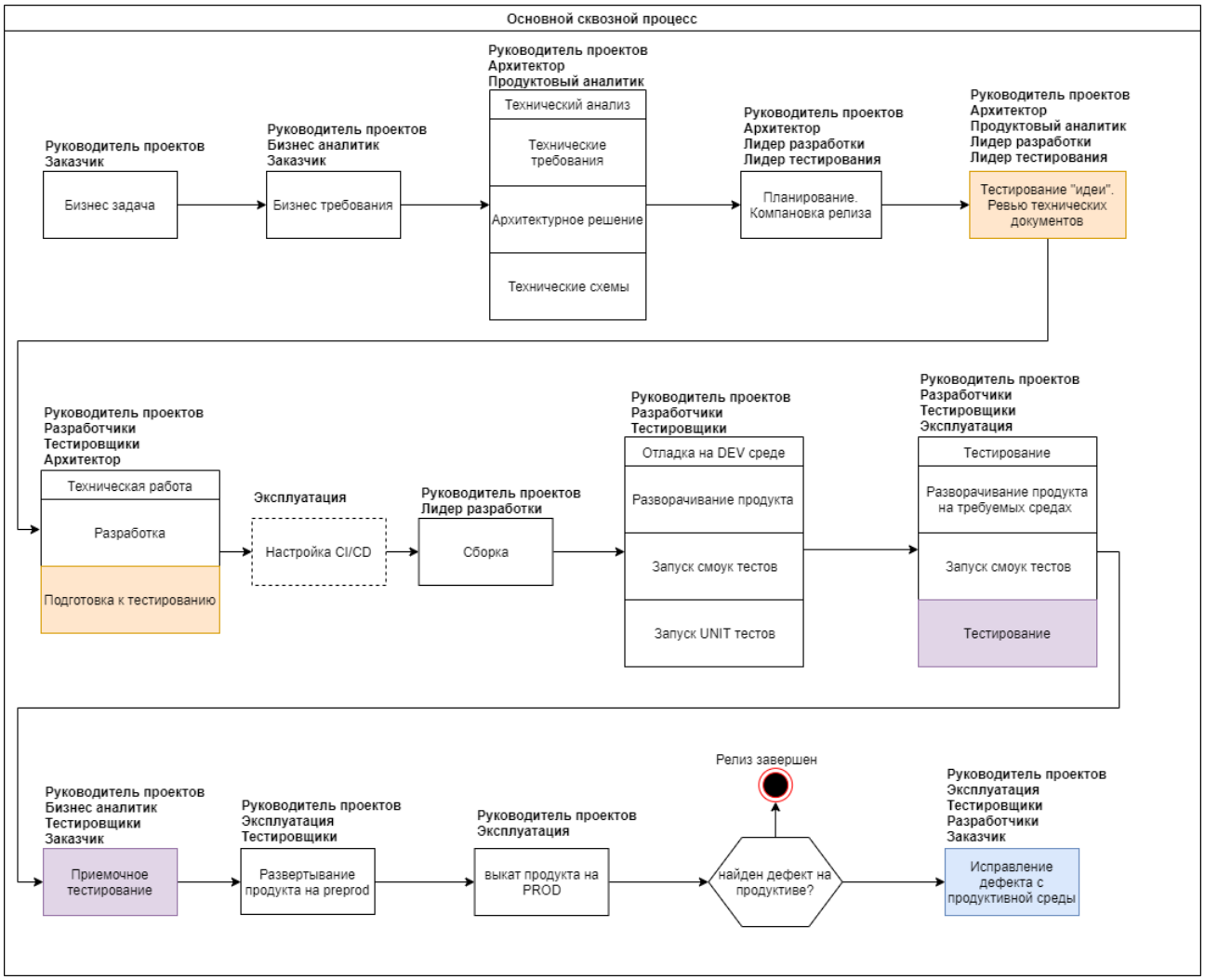

Рисунок 1 – Вехи работы команды

Обязательному контролю качества, регламентации и мониторингу соблюдения регламентов подлежат следующие процессы:

- Правила работы с изменениями – создание и ведение задач, работа с требованиями и базой знаний системы/проекта.
- Правила ведения веток кода.
- Контроль версий.
- Правила контроля качества на этапе кодирования.
- Частота и правила выпуска сборок.
- Правила разграничения доступов на различные среды.
- Автоматическая инсталляция продукта на различные среды.

- Автоматизация процессов разработки.
- Сопроводительные артефакты для каждого нового релиза/спринта.
- Применение/ прогон автотестов.
- Тестирование и работа с исправлениями.
- Правила работы с дефектами.
- Регулярная отчётность о результатах тестирования.

1.2 Итерационные методологии

Целью итераций является последовательное осмысление стоящих проблем, наращивание эффективности решений и снижение риска потенциальных ошибок. На выходе каждой итерации создается законченная версия работающего программного продукта.

При таком подходе можно гибко учитывать новые требования или проводить тактические изменения в деловых целях. Это позволяет на самых ранних этапах разработки выявлять и разрешать проблемы и, тем самым, снижать затраты.

Rational Unified Process предполагает разработку, реализацию и тестирование архитектуры на самых ранних стадиях выполнения проекта, устраняя самые опасные риски, связанные с архитектурой. Благодаря этому удается избежать существенных переработок на последних стадиях, если вдруг выяснится, что выбранное решение не обеспечивает, выполнение каких-либо требований.

Разработка информационных систем в компании ведется с использованием современных технологий и подходов, предназначенных как для быстрой реализации прототипов, так и для создания больших многоуровневых компонентных и масштабируемых решений корпоративного уровня. В своих проектах компания успешно применяет методологию создания прикладных программных систем, разработанную на базе RUP.

При создании приложений применяются инструментальные средства и технологии таких известных компаний как Oracle, IBM, Microsoft, Borland, а также собственные оригинальные разработки и технологии.

Для разработки больших и сложных информационных систем, создания компонентных и масштабируемых решений в компании применяются технологии JAVA и .NET. Использование собственных разработок позволяет не начинать реализацию проекта «с нуля» – имеется банк шаблонов стандартных проектов. Это дает возможность заказчику получить реально работающую и экономичную систему в максимально короткие сроки.

Укрупнённо проект по созданию программного обеспечения можно разделить на три фазы:

- Инициация;
- Исполнение;

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- Финал.

1.2.1 Инициация проекта

Предпроектное обследование инициируется запросом от Заказчика на предоставление технико-коммерческого предложения и проводится на основе информации, которая содержится в запросе. При необходимости Исполнитель проводит одно-два интервью с экспертами Заказчика или запрашивает дополнительные материалы. По итогам предпроектного обследования разрабатывается Коммерческое предложение, в котором представлены следующие артефакты:

- Концепция системы (функциональная и программно-аппаратная архитектура);
- План проекта;
- Оценка трудозатрат;
- Ценовое предложение.

Исполнитель использует в повседневной работе модель расчета трудоемкости и стоимости работ/услуг. Модель базируется на технологии ведения проектов, которая предполагает следующие этапы проекта: Техническое задание, Проектирование, Разработка, Тестирование, Документирование.

Такая модель оценки предназначена для расчета трудоемкости проекта на основании зависимости трудоемкости этапов проекта от трудоемкости этапа разработки и с учётом факторов влияния на различные этапы проекта.

Исходными данными для расчета трудоемкости проекта являются перечень предварительных требований к разрабатываемой системе и верхнеуровневое описание архитектуры приложения.

Расчет состоит из двух основных шагов:

- определение базовой трудоемкости этапа разработки;
- определение трудоемкости проекта.

Трудоемкость разработки является базой для определения трудоемкости остальных этапов проекта. При расчете базовой трудоемкости учитываются различные аспекты разработки, наличие которых корректирует трудоемкость данного этапа.

Для всех остальных этапов проекта определены нормативы зависимости от трудоемкости этапа Разработки (в процентах). Различные этапы могут входить или не входить в расчет трудоемкости того или иного проекта. По каждому этапу определяются факторы влияния, наличие которых увеличивает нормативную трудоемкость данного этапа.

Модель определяет три типа нормативов:

- нормативы трудоемкости разработки компонентов;
- нормативы коэффициентов для аспектов разработки;
- нормативы коэффициентов для этапов и факторов влияния.

Для расчета трудоемкости разработки определены следующие типы технических компонентов:

- Пользовательский интерфейс (экранные формы, веб-страницы, отчеты, печатные формы и пр.);
- Бизнес-логика (компоненты, реализующие логику работы приложения, алгоритмы обработки данных и пр.);
- Компоненты хранения данных (таблицы БД, серверные объекты, хранимые процедуры и пр.);
- Внешний интерфейс (компоненты, реализующие логику интеграции с внешними системами).

Для каждого типа технических компонентов определены градации сложности (от **простого** до **очень сложного**) и определены нормы трудозатрат на разработку (в человеко-часах), основанные на экспертной оценке.

Например, для разработки экранных форм классификация компонентов по сложности будет выглядеть так:

- **Простая** – окно авторизации или окно ввода данных с проверкой вводимых данных по формату (без логических связей между различными параметрами);
- **Средняя** – форма, реализующая поиск и фильтрацию данных с постраничным выводом;
- **Сложная** – экранная форма, содержащая не более двух вкладок с проверкой ввода данных по формату и логической взаимосвязи данных;
- **Очень сложная** – экранная форма, содержащая более двух вкладок с проверкой ввода данных по формату и с учетом логических взаимосвязей.

Такая классификация является лишь иллюстрацией, окончательное отнесение компонента к той или иной категории сложности выполняет архитектор.

Нормативы **трудоёмкости разработки** определяются в человеко-часах в разрезе сложности компонентов.

Нормативы **коэффициентов для аспектов разработки** определяются в процентах относительно трудоемкости разработки технических компонентов и разработки в целом.

Нормативы **коэффициентов трудоемкости** определяются в процентном соотношении к Разработке или другим этапам и факторам влияния.

В таблице 1. приведен пример нормативов для этапов тестирования ПО и трудоемкости управления

Таблица 1 – Примеры коэффициентов для расчета трудоемкости этапов

Этап	Описание этапа	Детали расчета фактора/этапа
Интеграционное тестирование	Интеграционное тестирование (включает время на исправление найденных во время тестирования критичных дефектов)	Процент от Разработки внешних интерфейсов
Нагрузочное тестирование	Нагрузочное тестирование (включает время на исправление найденных во время тестирования критичных дефектов)	Процент от Разработки
Документирование	Документирование	Процент от Разработки
Интеграционное тестирование	Интеграционное тестирование (включает время на исправление найденных во время тестирования критичных дефектов)	Процент от Разработки внешних интерфейсов

Фактор влияния – это обстоятельство или условие, увеличивающее трудоемкость этапа или вида работы. Примеры факторов влияния и нормативы соответствующих коэффициентов приведены в таблице 2.

Таблица 2 – Примеры коэффициентов для расчета факторов влияния

Этап	Фактор влияния	Детали расчета фактора/этапа
Техническое задание	Комплексно сложный продукт (необходимо написание ЧТЗ на различные компоненты/модули системы)	Процент от Технического задания
Проектирование	Использование специфической методологии, нотации, специальные требования заказчика и т. д	Процент от Проектирования
Разработка	Специфическая среда разработки (например отсутствие средств рефакторинга)	Процент от Разработки
Разработка	Обеспечение качества: код-ревью. Этот фактор может не учитываться в случае немасштабного проекта, разработки макета или прототипа	Процент от Разработки

Этап	Фактор влияния	Детали расчета фактора/этапа
Разработка	Обеспечение качества: юнит-тестирование. Этот фактор может не учитываться для немасштабного проекта, разработки макета или прототипа	Процент от Разработки
Функциональное тестирование	Необходимость развёртывания тестовой среды на площадке Заказчика	Процент от Функционального тестирования
Функциональное тестирование	Необходимость разработки тестовой платформы (автотесты, эмуляторы и пр.). Наличие этого фактора зависит от масштаба, критичности, планов поддержки разрабатываемой системы и т. п.	Процент от Разработки
Интеграционное тестирование	Отсутствие технических специалистов на стороне внешних систем	Процент от Интеграционного тестирования
Документирование	Написание документации на иностранном языке (перевод). За каждый дополнительный язык устанавливается свой норматив. Консультирование переводчика	Процент от Документирования
Документирование	Заказчик – Госструктура (требование оформления документации по ГОСТу или использование отраслевых или иных специализированных стандартов)	Процент от Документирования
Документирование	Необходимость подготовки учебных курсов	Процент от Документирования
Документирование	Необходимость сертификации (ФСБ, ФСТЭК, Минкомсвязь и пр.) – полный пакет документов	Процент от Документирования
Управление	Требование ведения работ по разработке продукта на площадке Заказчика	Процент от Управления

1.2.2 Исполнение проекта

Общая схема технологии разработки ПО представлена на рисунке 2.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

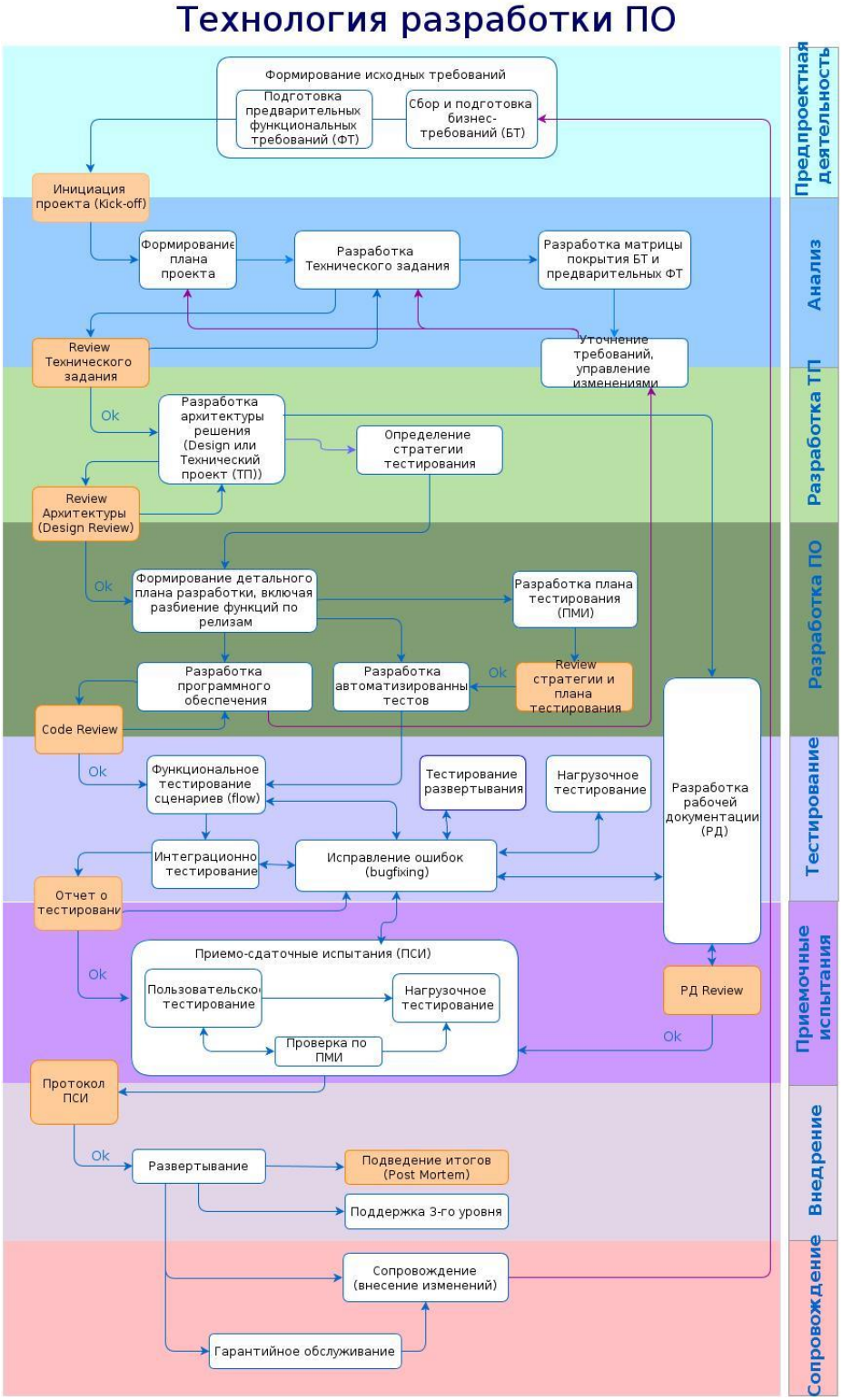

Рисунок 2 – Технология разработки программного обеспечения

Разбиение на этапы приведено с учётом времени выполнения той или иной работы и не всегда совпадает с логическим разбиением работ по типам. Например, работы «Определение стратегии тестирования» и «Разработка программы и методики испытаний»

относятся к внутреннему тестированию, а на схеме отнесены к этапам «Разработка технического проекта» и «Разработка программного обеспечения» соответственно.

На первом этапе, в начале проекта формируются бизнес-требования и предварительные функциональные требования, после чего инициируется проект, который переходит в этап анализа. На этапе анализа формируется детальный план проекта, проводится бизнес-анализ, разрабатываются техническое задание и матрица соответствия. Этап подразделяется на работы:

- Обследование (проведение серий интервью), детализация и конкретизация бизнес-требований;
- Разработка технического задания;
- Описание функциональности системы и бизнес-процессов «как будет». Здесь же определяется логическая структура объектов и их атрибутный состав.

По итогам этапа разрабатываются документы:

- Отчет об обследовании (необязательный документ, необходимость разработки определяется в зависимости от особенностей проекта и согласовывается с Заказчиком)
- Бизнес-требования (как правило, содержатся в запросе на технико-коммерческое предложение, в некоторых случаях требуется их детализация до разработки Технического задания);
- Техническое задание (обязательный документ).

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

Этап технического проектирования подразделяется на работы:

- Проектирование (детализация) архитектуры разрабатываемого программного комплекса;
- Проектирование физических структур объектов хранения;
- Детальное определение интерфейсов взаимодействия с внешними системами;
- Проектирование пользовательского интерфейса.

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

Review Технического проекта происходит и на этапе разработки программного обеспечения.

На этапе разработки программного обеспечения формируется детальный план разработки, включая разбиение функций по релизам, и разработка программного обеспечения. Кроме того, на этом этапе разрабатываются автоматизированные тесты и план тестирования, которые входят и в этап внутреннего тестирования, как и review code. Детализация этапа разработки совпадает с детализацией, определяемой архитектором при расчете трудоемкости.

Внутреннее тестирование подразделяется на работы:

- Определение стратегии тестирования (на схеме отнесено к этапу «Разработка ТП»);
- Разработка программы и методики испытаний (отнесено к этапу «Разработка ПО»). Для проектов, не требующих оформления по ГОСТ, разрабатываются сценарии тестирования;
- Проведение тестирования;
- Исправление дефектов по результатам тестирования.

Отдельно, в зависимости от специфики проекта, определяется необходимость нагрузочного тестирования и автоматизированного и (или) регрессионного тестирования. Одновременно определяется необходимость разработки соответствующих скриптов.

Документирование подразумевает разработку пользовательской и эксплуатационной документации, сопровождающей поставку программного продукта, а также оформление проектной документации, разработанной в ходе проектирования в соответствии со стандартами, используемыми в проекте (ГОСТ, RUP и т. д.). Точный перечень документов, разработка которых требуется при выполнении проекта, определяется в контракте и Техническом задании. В обязательном порядке разрабатываются документы:

- Руководство пользователя;
- Руководство администратора.

Документацию можно разделить на релизную, эксплуатационную и сертификационную. Наиболее сложная и объемная документация – сертификационная.

На этапе приемочного тестирования выполняется различные виды тестирования.

1.2.2.1 Управление требованиями

При реализации проектов используется систематический подход к управлению требованиями и отслеживанию их изменений. В рамках процесса управления требованиями выполняется: выявление, документирование, верификация и утверждение требований, планирование реализации и отслеживание изменений требований.

Типовой состав участников проектной группы в рамках процесса управления требованиями выглядит следующим образом: системный аналитик; специалист по требованиям; рецензент

требований; архитектор. Кроме того, в процессе задействован эксперт предметной области. В большинстве проектов роль эксперта, как правило, играет представитель заказчика, но в ряде случаев эксперт предметной области входит в состав проектной группы со стороны разработчика.

Для систематизации сбора и структуризации общей базы знаний используется Wiki-система Confluence.

Процесс управления требованиями изображен на рисунке 3.

1.2.2.2 Политика релизов

Политика релизов (выпуск версий информационных систем) фиксируется до начала проектных работ и может отличаться в проектах, выполняемых для различных заказчиков. Поэтому, в начале проекта, прежде всего, определяется и согласовывается с заказчиком классификация выдаваемых ему релизов. Например, одной из возможных классификаций является разбиение релизов на Major/Minor-релизы с точки зрения функционала и Candidate/Final-релизы с точки зрения процесса разработки.

К Major-релизам относятся все релизы с изменениями в функциональности и релизы, содержащие пакеты исправлений выявленных критичных дефектов в функциональности промышленной версии продукта.

К Minor-релизам относятся релизы, содержащие пакеты исправлений дефектов, обнаруженных в выданном ранее Major-релизе, не запущенном пока в промышленную эксплуатацию.

К Candidate-релизу относится готовая сборка всех компонентов продукта, содержащая реализованные функциональные требования, которые предъявляются к данному релизу, но не прошедшая этап тестирования.

После успешного прохождения этапа внутреннего тестирования релиза и выдачи его заказчику Candidate-релиз переходит в разряд Final-релизов.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

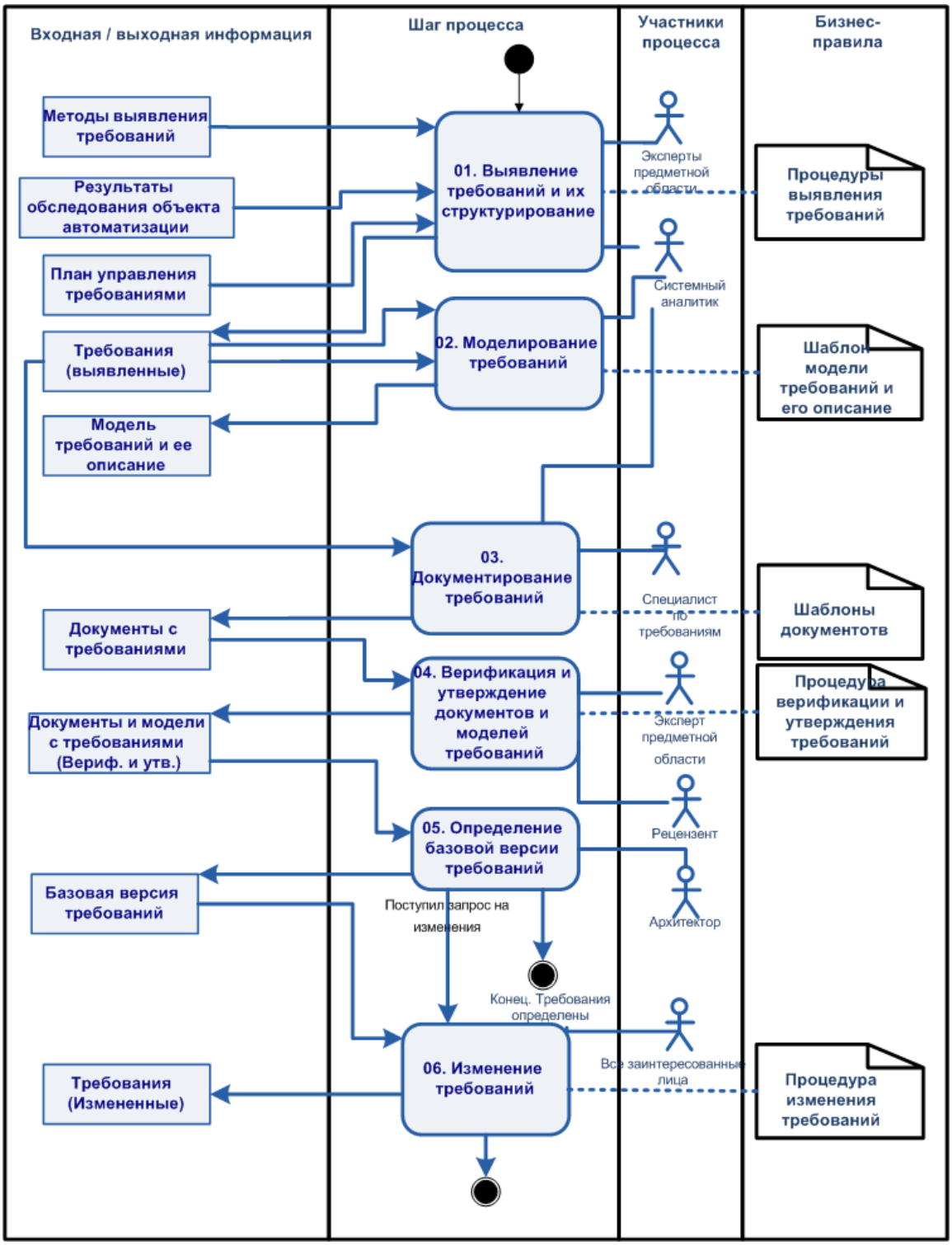

Рисунок 3 – Процесс управления требованиями

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

1.2.2.3 Порядок именования релизов

Название любого релиза состоит из четырёх цифр X.Y.W.Z,

где:

X – изменение платформы;

Y – плановые релизы с новой функциональностью;

W – внеплановые релизы с новой функциональностью, исправления продуктивной версии;

Z – порядковый номер выданного Minor-релиза.

Если в ходе приёмочного тестирования (до установки Major-релиза в продуктивную среду) были выявлены критичные дефекты, их исправление проводится в рамках Minor-релиза. Этот релиз получает метку X.Y.W, совпадающую с меткой Major-релиза, а индекс Z инкрементируется на единицу.

Если ведётся разработка продукта с адаптацией под различные типы платформ, в метку названия релиза вносится дополнительный параметр Platform. Соответственно, метка релиза меняется на X.Y.W.Z-Platform, где Platform – тип рабочей платформы продукта.

Если продукт состоит из отдельных модулей, подразумевающих возможность их отдельного использования или функционирования, политика их именования строго совпадает с общей принятой политикой именования продукта. Все используемые в продукте компоненты при выдаче очередной Major-версии продукта заказчику имеют одинаковые метки X.Y.W, которые гарантированно определяют принадлежность данных версий компонентов версии Major-релиза продукта, которая поступает Заказчику.

1.2.2.4 Система контроля версий

Для работы с исходными кодами и документацией используется система управления версиями Subversion (SVN). В качестве системы управления репозиториями кода для Git используется GitLab.

SVN имеет древовидную структуру хранения данных проекта, которая позволяет:

- отслеживать все изменения;
- организовывать одновременную работу группы разработчиков над общим программным кодом продукта;
- организовывать одновременную работу над несколькими версиями продукта, восстановление измененных и удаленных частей исходных кодов продукта.

Для каждой информационной системы в системе контроля версий создаётся отдельный, соответствующий этой системе проект.

При работе с исходными кодами участники проекта придерживаются следующих правил:

- В момент фиксации Candidate-релиза на его основе создается ветка проекта, в которой идет дальнейшая работа в рамках исправления критичных ошибок, выявленных в ходе тестирования.
- Название ветки однозначно идентифицирует разрабатываемую версию программного продукта и имеет вид: <имя проекта/компонент>X.Y.W.
- Для каждой версии Major-релиза создается ветка документации, содержащая весь перечень сопроводительных документов системы.
- Все исправления для данной ветки дерева проекта ведутся в рамках создания Minor версий релизов продукта. При этом отдельные ветки для Minor-версий не создаются.
- Когда ветка проекта создана, ведется работа над следующей Major-версией релиза продукта.
- Все исправления, сделанные в рамках разработки Minor-версий продукта в соответствующей ветке, должны быть проанализированы на предмет их переноса в другие ветки/ствол дерева проекта — Major-версии релизов программного продукта.

Соблюдение этих правил в работе с деревом проекта, а также особенности архитектуры систем контроля версий позволяют в любой момент времени получать актуальные версии различных релизов информационных систем.

1.2.2.5 Резервирование релизов

Каждый выдаваемый заказчику релиз, включая исходные коды и документацию всех компонентов этого релиза, в обязательном порядке сохраняется в отдельном хранилище данных. Кроме того, при необходимости, создаются дополнительные резервные копии на оптических или иных носителях.

1.2.2.6 Тестирование программного обеспечения

Для тестирования ПО используется широкий набор программного инструментария: начиная со свободно распространяемых средств тестирования и заканчивая внутренними разработками. Так, для решения задач по автоматизации процесса тестирования, который бы позволял быстро и точно адаптироваться к специфике различных программных решений, была разработана своя тестовая платформа.

В основе этой платформы лежит технология Java, которая позволяет избавиться от платформенных ограничений. Это, в свою очередь, дает возможность отделить тестовую среду от среды разработки без потери функциональности.

При реализации компонентов платформы были учтены жесткие требования к большим нагрузкам, поэтому платформа может использоваться и для функциональной, и нагрузочной серии испытаний.

Тестовая платформа, разработанная в компании «Инфосистемы Джет», имеет следующие основные преимущества:

1. Типизация – написание шаблонов, упрощающих написание серии тестов.
2. Поддержка большинства используемых технологий с помощью реализованных интерфейсов и адаптеров.
3. Возможность добавления различных модулей с поддержкой дополнительных сервисов.
4. Запуск тестов полностью в автоматическом режиме.
5. Поддержка регрессивного тестирования с расширенным протоколированием результатов.
6. Поддержка нагрузочного тестирования.
7. Автоматическая генерация отчётов о тестировании.

Для обеспечения качества тестирования специалисты Компании разработали комбинации тестов новой функциональности и регрессионных тестов. Кроме функционального тестирования, периодически проводятся тесты производительности (performance), тестирование общей надежности (stability), тестирование поведения при перегрузке (stress-test), тесты пользовательского интерфейса (GUI), автотесты.

1.2.2.7 Автоматизация процессов тестирования

Для автоматизации тестирования применяются различные программные продукты. Их описание приведено в таблице 3.

Таблица 3 – ПО для автоматизации процессов тестирования

Этап цикла ПО	Инструмент	Преимущество
Формирование требований, Планирование	Confluence, MS Project	Требования ведутся централизованно; есть возможность отслеживать историю изменений; актуальная информация доступна в режиме онлайн, планирование ведётся прозрачно
Аналитика	Confluence, Jira	Требования ведутся централизованно; есть возможность отслеживать историю изменений; актуальная информация доступна в режиме онлайн; процессы аналитики, разработки и тестирования могут идти параллельно по мере готовности одного требования и занесения его в виде задачи в Jira, за

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

Этап цикла ПО	Инструмент	Преимущество
		счет чего ускоряются процессы разработки и тестирования. Ошибки выявляются на более ранних стадиях разработки, за счёт этого экономятся ресурсы, затраченные на исправление одной ошибки. Повышается качество выдаваемого продукта
Кодирование (разработка)	Confluence, Jira, Git, GitHub, Intellj Idea, Ant, Maven, Gradle, Sonar Qube, Jenkins	Введение правил разработки, их включение в интегрированных средах разработки, анализ кода, автосборка и интеграция с инструментами CI/CD позволяет сократить время на подготовку сборки для тестовой и промышленной среды. Ошибки выявляются на более ранних стадиях разработки, за счёт этого экономятся ресурсы, затраченные на исправление одной ошибки. За счёт юнит-тестов, разрабатываемых совместно с новым кодом, повышается качество выдаваемого продукта
Разработка	Jira, Git, Jenkins,Docker, AWS, Ansible, Kubernetes, Allure, Maven, Gradle, Sonar Qube, Python, Zabbix, ELK	Построение полноценного CI/CD позволяет сократить время: на подготовку сборки для тестовой и промышленной среды; подготовку тестовых данных; время, затраченное на тестирование в целом. Ошибки выявляются на более ранних стадиях разработки, за счёт этого экономятся ресурсы, затраченные на исправление одной ошибки. Повышается качество выдаваемого продукта
Тестирование	Confluence, Jira, Git, Java, Junit, TestNG, Selenium, Selenide, Ant, Maven, Gradle, RestAssure, Allure, Testrail, TestManagmentTool	Автоматизация тестирования позволяет сократить время, затраченное на тестирование в целом. Повышается качество выдаваемого продукта. Процесс тестирования и его результаты прозрачны для каждого прогона и имеют свою историчность. Дашборды отображают качество продукта в режиме онлайн
Мониторинг сред и сбор системных и продуктовых метрик	ELK, Grafana, Zabbix	Мониторинг позволяет: заблаговременно предотвратить инфраструктурные проблемы; вовремя реагировать на возникновение блокирующих и критичных ошибок на различных средах; учесть возникшие проблемы в тестовой модели для недопущения повторения подобной ошибки. Продуктовые метрики позволяют оперативно оценивать продукт с точки зрения бизнеса и своевременно подстраиваться под потребности конечного пользователя

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

1.2.2.8 Процесс управления дефектами

Для регистрации дефектов ПО и их исправления применяется формализованная методика.

В качестве инструментального средства управления дефектами используется открытое программное обеспечение Bugzilla.

Процесс управления дефектами изображен на рисунке 4.

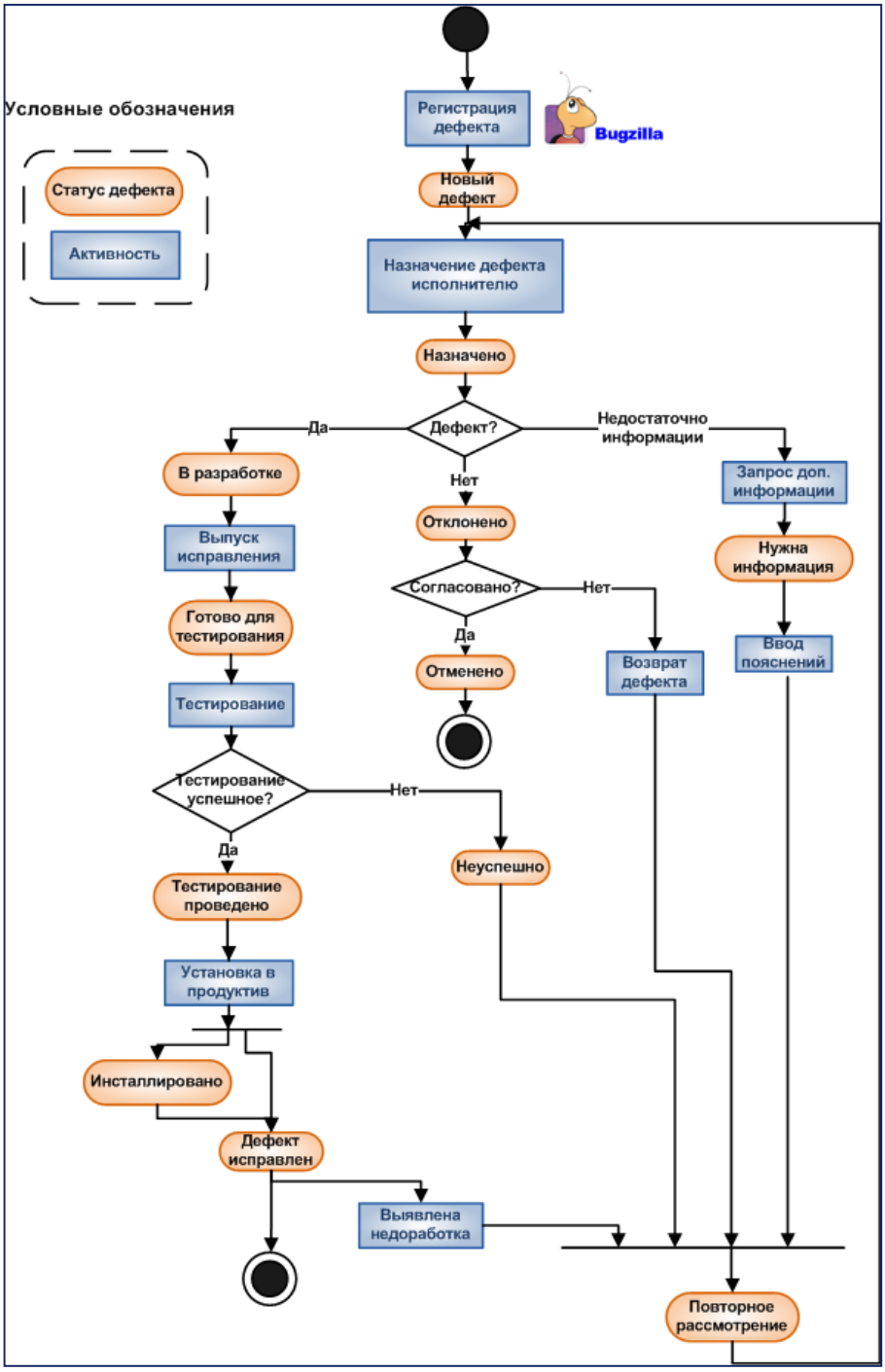

Рисунок 4 – Процесс управления дефектами

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

1.2.3 Финальная фаза проекта

Финальная фаза включает в себя сдачу ПО Заказчику и его гарантийное обслуживание.

Сдачу-приёмку готового продукта принимает комиссия, состав которой утверждает Заказчик. Члены комиссии проводят испытания в соответствии с положениями и проверками, включенными в утвержденную Программу и методику испытаний. Сдача ПО оформляется протоколом приемочных испытаний и актом.

Приёмо-сдаточные испытания могут проходить в два этапа. Необходимость предварительных и окончательных испытаний устанавливается по согласованию с Заказчиком.

На находящиеся в эксплуатации программные продукты распространяются гарантийные обязательства Исполнителя. Гарантийный срок сопровождения составляет 12 месяцев начиная с даты приёмки работ. По соглашению с Заказчиком этот срок может изменяться. В течение гарантийного периода все выявленные дефекты Исполнитель устраняет бесплатно.

Постгарантийное обслуживание регламентируется отдельным договором. Такой договор предусматривает заключение двухстороннего соглашения об уровне сервиса. Дополнительная информация о технической поддержке и Сервисном центре компании «Инфосистемы Джет» приведена в разделах 2 и 3.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

2 Сопровождение программных продуктов. Сервисный центр компании «Инфосистемы Джет»

Сервисный центр образован в 1993 году для обеспечения технического обслуживания и поддержки сложных информационных систем корпоративных заказчиков – от ремонта вышедшего из строя оборудования до полнофункционального аутсорсинга информационных систем.

Для выполнения работ по запросам (заявкам) в Сервисном центре предусмотрены следующие функциональные роли:

- Руководитель процесса управления инцидентами – отвечает за решение стратегических и тактических задач процесса, осуществляет общий контроль процесса.
- Руководитель Диспетчерской службы (Менеджер процесса управления инцидентами) – отвечает за решение тактических и операционных задач приема и обработки заявок, осуществляет оперативный контроль процесса.
- Инициатор заявки – подаёт в Диспетчерскую службу заявку на устранение инцидента или предоставление обслуживания, предоставляет информацию, необходимую для классификации заявки, диагностики и разрешения инцидента Исполнителем, оценивает качество выполнения заявки при ее закрытии.
- Диспетчер (Диспетчерской службы) – отвечает за прием и фильтрацию обращений, регистрацию заявок, правильную классификацию и назначение на Инженера первой линии поддержки или Технического ответственного, выделение исполнителей работ по заявке (Инженеров), контроль статуса обработки заявки, соблюдение сроков и закрытие заявки.
- Технический ответственный – отвечает за определение состава работ по заявке, выполнение и/или организацию выполнения работ по заявке и контроль качества выполнения работ.
- Инженер 1-й линии поддержки – отвечает за первичную диагностику зарегистрированного инцидента, определение способа его разрешения и правильность применения инженером-исполнителем предложенного решения.
- Исполнитель (Инженер) – отвечает за выполнение и фиксацию результатов выполнения назначенных ему заявок и заданий, инициирует открытие заявок при обнаружении сбоев в функционировании оборудования и ПО заказчика.
- Эксперт – выполняет экспертную поддержку, консультирование Инженера 1-й линии, Технического ответственного за заявку или Исполнителя по техническим вопросам, связанным с разрешением инцидентов или внесением изменений в конфигурации и настройки поддерживаемого оборудования и ПО.

- Сервис-менеджер СЦ – отвечает за обеспечение выполнения контрактных и гарантийных обязательств перед VIP-заказчиками, наделен специальными полномочиями по контролю хода обработки заявки VIP-заказчика.
- Руководитель структурного подразделения СЦ – привлекается при возникновении сложностей с определением исполнителей работ по заявке (выделением ресурсов), а также наделен специальными полномочиями по координации работ для разрешения инцидентов с признаком «Критический сбой».

Прием и обработка заявок в Сервисном центре проводится в соответствии со следующими правилами:

1. Диспетчерская служба СЦ является единой точкой контакта для всех инициаторов заявок на выполнение работ в СЦ.
2. Диспетчерская служба принимает и регистрирует заявки, связанные с предоставлением услуг по обслуживанию и поддержке оборудования и ПО, в том числе поступающие от заказчиков, не имеющих действующих сервисных контрактов с СЦ (например, запрос заказчика на предоставление информации об условиях заключения сервисного контракта). Режим работы Диспетчерской службы – официальные рабочие дни с 9.00 до 20.00.
3. Для выполнения обязательств СЦ по сервисным контрактам с режимом обслуживания 24x7 в нерабочие часы Диспетчерской службы прием обращений заказчиков СЦ осуществляет инженер Дежурной смены. При этом инженер пишет отчет о результатах обработки заявок, принятых во время дежурства.
4. Для обработки заявок в СЦ выделены 1-я и 2-я линия обработки заявок.
5. Задачи 1-й линии:
   - Уточнение характера проблем у заказчика (уточнение постановки задачи);
   - Сбор первичной информации о сбое;
   - Локализация сбоя с точностью до технической области;
   - Формирование предварительных выводов о возможных причинах сбоя/отказа с последующей эскалацией (переводом) на 2-ю линию или производителя;
   - Определение компетенции инженера 2-й линии, который в состоянии далее работать с заявкой;
   - Формирование плана работ по устранению сбоя и восстановительных работ;
   - Выявление «типовых» отказов, диагностика, подготовка плана решения;
   - Устранение сбоя (с привлечением инженера 2-й линии):

- Контроль хода проведения работ по устранению сбоя инженером 2-й линии;
- Приемка и тестирование работоспособности оборудования и ПО после восстановления.

6. Задачи 2-й линии:
   - Устранение «нетиповых» сбоев;
   - Выполнение плановых работ;
   - Выполнение запросов на изменение;
   - Выполнение проектных работ;
   - Выполнение других работ по заявкам.
7. Все заявки регистрируются в единой базе данных системы автоматизации Remedy Help Desk.
8. Учётная форма заявки регистрируется в системе на основании обращения представителя заказчика или сотрудника компании «Инфосистемы Джет».
9. Учетная форма задания регистрируется в системе Техническим ответственным за заявку в случае, если ему необходимо привлечь (выдать задание) другого инженера для выполнения работ по заявке.
10. Существуют нормативные сроки обработки заявки в привязке к временным параметрам сервисного контракта (время реакции, время восстановления), статусу и приоритету заявки.
11. У каждой заявки есть ответственный Диспетчер СЦ, который отвечает за организацию выполнения работ по заявке в соответствии с контрактными обязательствами.
12. У каждой заявки есть Технический ответственный (инженер 1-й линии, инженер 2-й линии, эксперт), который отвечает за качество решения, примененного им для выполнения заявки (устранения сбоя или выполнения плановой задачи), и соблюдение установленных Диспетчером СЦ сроков обработки заявки.
13. У каждого задания есть Исполнитель, который отвечает за качественное выполнение задания в соответствии с установленными плановыми сроками. Ход работ и результат выполнения работ по заданию контролирует и принимает Технический ответственный, который сформировал это задание.
14. При возникновении любых сложностей с выполнением заявки, по содержанию работ или срокам, Технический ответственный за заявку обязан оперативно сообщить об этом Диспетчеру и руководителю своего подразделения (отдела СЦ); Диспетчер и руководитель подразделения СЦ принимают меры по выполнению заявки в

соответствии с установленными нормативными сроками или эскалируют ситуацию на руководство СЦ.

15. Если при назначении Технического ответственного за заявку (или исполнителя задания) Диспетчер выявил «конфликт ресурсов», Диспетчер привлекает Руководителя процесса управления инцидентами для разрешения данного конфликта.
16. Переназначение Технического ответственного за заявку может выполнять Диспетчер СЦ.
17. Все поступившие в заявки закрываются только после получения подтверждения от заказчика и оценки заказчиком качества выполнения работ.

3-й уровень поддержки предусматривает устранение проблем путём доработки или изменения ПО и выпуска новой версии Системы. При этом Исполнитель обеспечивает устранение дефектов, ошибок и их результатов, эскалированных с первого и второго уровня поддержки Заказчика. Кроме того, Исполнитель ищет обходные решения для запросов со статусом «инцидент», которые поступили Исполнителю с первого и второго уровня поддержки.

1. Исполнитель обеспечивает гарантированный уровень качества работы эксплуатируемой Системы в рамках своих обязательств.
2. Исполнитель обеспечивает согласованное с Заказчиком время работы диспетчеров и консультантов 3-й линии поддержки по локализации возникающих проблем и выработки временного решения в соответствии со сроками, определенными в соглашении об обслуживании.
3. Исполнитель обеспечивает ведение и поддержку в актуальном состоянии эксплуатационной документации – с учетом выявленной информации по проблемам, ошибкам, сбоям, путям выхода из них, а также с учётом особенностей работы и настройки Системы.
4. Исполнитель обеспечивает тестирование и всесторонний контроль новых версий Системы перед ее установкой в промышленную среду.
5. Исполнитель предоставляет Заказчику объективную информацию о качестве Системы в виде регулярных отчетов об обработке инцидентов, проблем и дефектов.

Иллюстрация процессов поддержки

Ниже на рисунках приведены схемы процессов полного цикла поддержки:

- Рисунок 5 – Схема процесса обработки запросов.

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- Рисунок 6 – Схема обработки инцидентов, проблем и дефектов.
- Рисунок 7 – Схема взаимодействия между вторым и третьим уровнем поддержки при исправлении дефектов.
- Рисунок 8 – Схема взаимодействия между вторым и третьим уровнем поддержки при консультировании.
- Рисунок 9 – Схема взаимодействия между вторым и третьим уровнем поддержки при решении инцидентов.

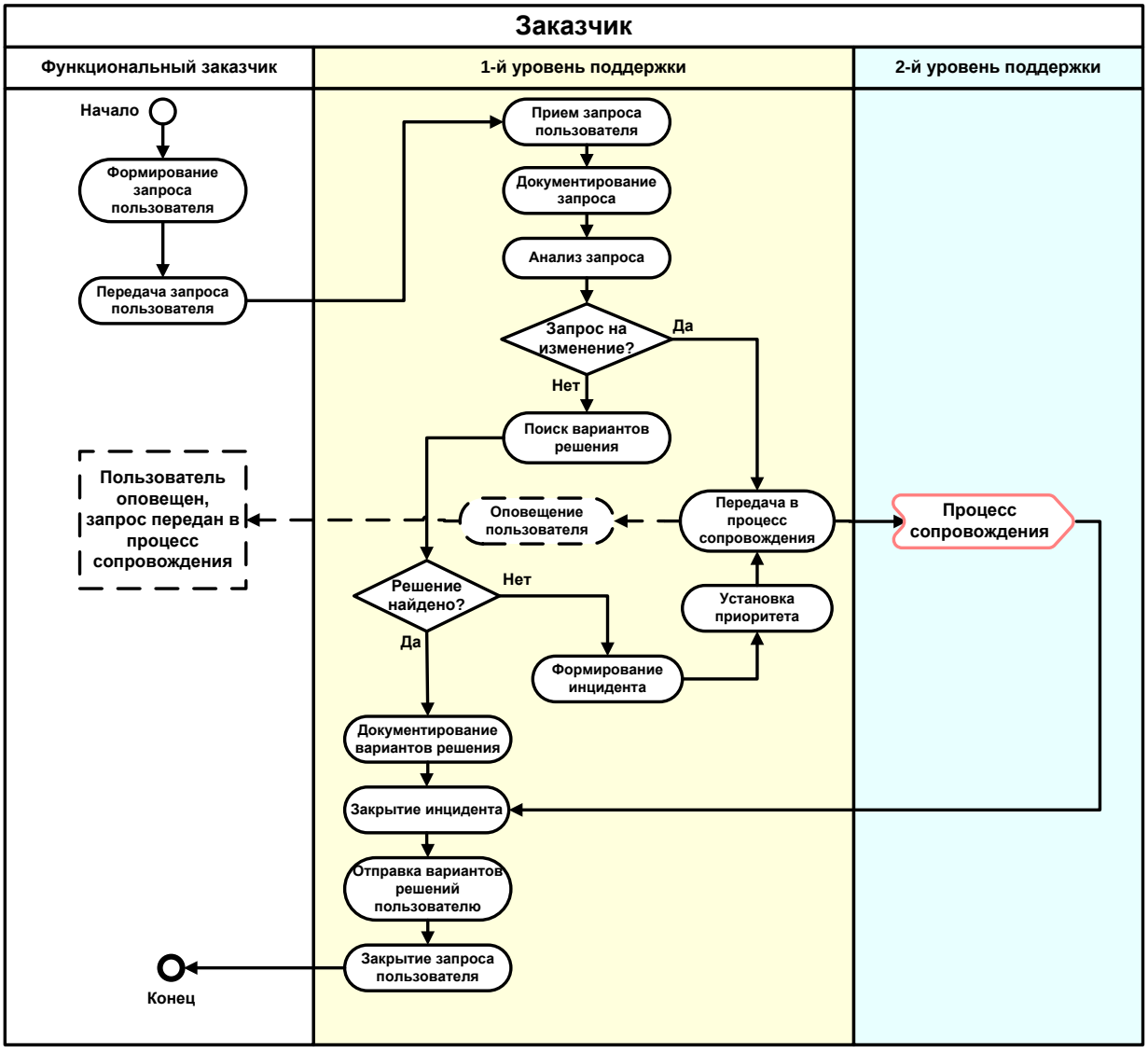

Рисунок 5 – Схема обработки запросов пользователей

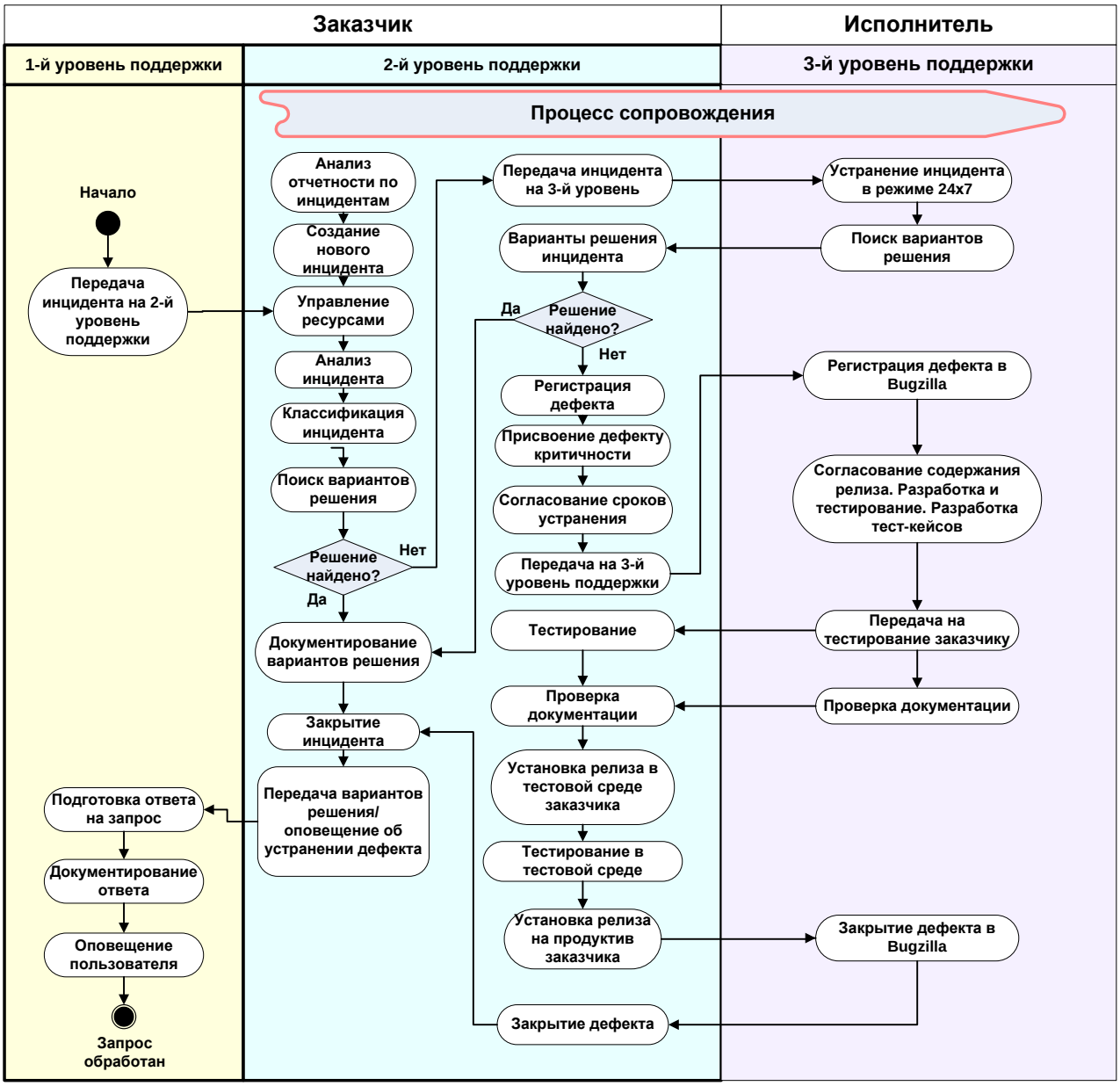

Рисунок 6 – Схема обработки инцидентов, проблем и дефектов

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

Рисунок 7 – Схема взаимодействия между вторым и третьим уровнем поддержки при исправлении дефектов

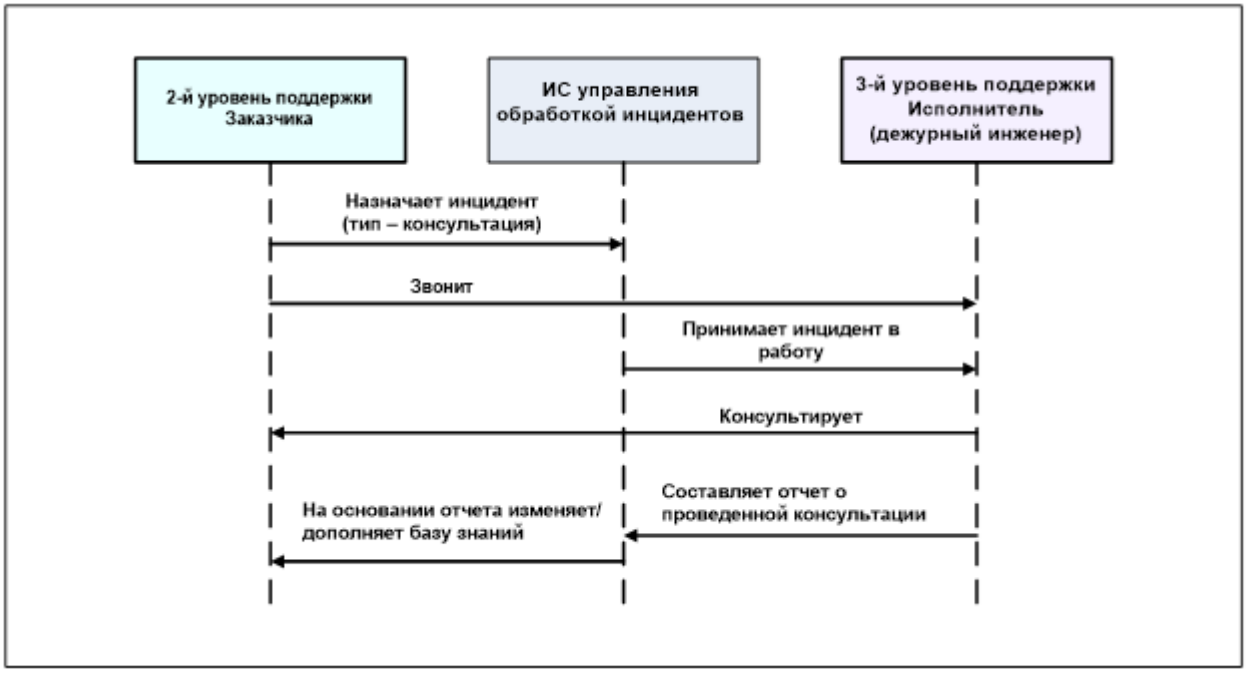

Рисунок 8 – Схема взаимодействия между вторым и третьим уровнем поддержки при консультировании

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

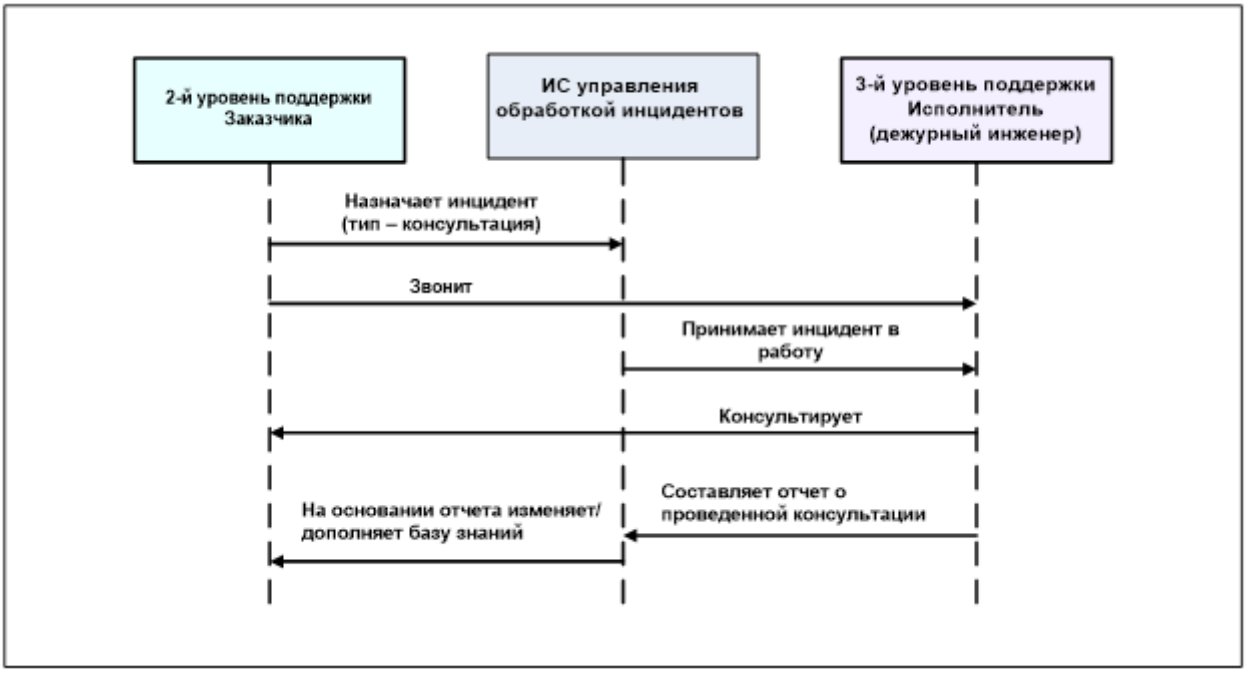

Рисунок 9 – Схема взаимодействия между вторым и третьим уровнем поддержки при решении инцидентов

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

3 Информация о Сервисном центре компании АО «Инфосистемы Джет»

Компания «Инфосистемы Джет» специализируется на обслуживании сложных вычислительных комплексов и информационных систем корпоративного уровня. Для этих целей в 1993 году в компании был образован собственный Сервисный центр. Сегодня Сервисный центр компании «Инфосистемы Джет» – это крупнейшее в России и СНГ сервисное предприятие рынка ИТ. В его составе – более 350 инженеров и администраторов, имеющих более 1500 сертификатов компаний-производителей оборудования и ПО. Кроме того, Сервисный центр располагает штатом квалифицированных менеджеров проектов и опытных диспетчеров.

Запросы на обслуживание круглосуточно принимает единая диспетчерская служба: в режиме 24×7, по нескольким каналам связи. Служба поддержки пользователей Сервисного центра сертифицирована по стандарту международной системы менеджмента качества ISO 9001:2000. Все поступающие в Сервисный центр запросы, а также процесс их выполнения в обязательном порядке регистрируются в автоматизированной системе учета и контроля исполнения заявок. Кроме того, в системе учета и контроля исполнения заявок создана и ведется база конфигураций обслуживаемых компонентов информационных систем заказчиков, что позволяет осуществлять консолидированное обслуживание разнородных ИТ-инфраструктур, учитывая все особенности их эксплуатации. На данный момент база конфигураций содержит более 140 000 элементов.

Сервисный центр обладает собственным Центром удаленного мониторинга информационных систем заказчиков. Автоматизированная система мониторинга построена на основе программных решений BMC Software и Hewlett Packard Enterprise. Услуги по мониторингу и администрированию предоставляются специально организованной круглосуточной службой, включающей инженеров и администраторов разных специализаций.

В распоряжении Сервисного центра имеется собственная техническая лаборатория для моделирования проблемных ситуаций, которые могут возникнуть на программно-аппаратных комплексах, установленных у заказчиков. Лаборатория имеет в своем составе серверное и сетевое оборудование масштаба предприятия, кластерные конфигурации серверов, специализированное управляющее и прикладное программное обеспечение.

На сегодняшний день на поддержке находится оборудование и ПО более 250 государственных и коммерческих организаций, расположенных в 80 городах России и СНГ, от Калининграда до Владивостока.

Обслуживается более:

- 8 800 корпоративных серверов;
- 2 500 дисковых массивов и библиотек;
- 4 100 единиц сетевого оборудования;

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

- 50 серверов и 50 массивов класса High-End;
- 50 кластерных комплексов.

Центральный офис СЦ находится в Москве. Кроме того, работают филиалы, имеющие собственный инженерный персонал: восемь – на территории Российской Федерации, а также в Республике Казахстан (г. Алматы), в Республике Азербайджан (г. Баку).

Потребности каждого заказчика в услугах технической поддержки и ИТ-аутсорсинга уникальны. В результате заказчик получает:

- возможность сосредоточиться на решении ключевых для бизнеса задач;
- сокращение объема собственных работ по управлению ИТ-инфраструктурой;
- снижение рисков при эксплуатации информационных систем;
- оптимизацию издержек на содержание собственной ИТ-службы;
- возможность прогнозирования эксплуатационных и финансовых расходов;
- обеспечение гибкости и быстрой реакции ИТ-подразделения компании на изменения требований бизнеса;
- гарантию высокого качества услуг от компании «Инфосистемы Джет».

127015, Россия, Москва, Б. Новодмитровская 14/1
тел. +7 495 411 76 01
факс +7 495 411 76 02
info@jet.su
www.jet.su

Термины и сокращения

Термин/сокращение	Описание
CI/CD	Continuous Integration/ Continuous Delivery. Непрерывная интеграция, доставка и развёртывание ПО. Часть концепции DevOps
DevOps	Development&Operations. Методология взаимодействия специалистов при разработке ИТ-решений
ISO	Международная организация по стандартизации
QA	Quality Assurance. Совокупность мероприятий, охватывающих все этапы разработки ПО. Применяется на разных стадиях жизненного цикла ПО
RUP	Rational Unified Process. Методология разработки информационных систем
SCRUM	Один из методов управления проектами
ИТ	Информационные технологии
ПО	Программное обеспечение
СЦ	Сервисный центр
ФСТЭК	Федеральная служба по техническому и экспортному контролю