Методологии проектирования

Мартынов А.И. – директор департамента разработки ITECH.group

Ульяновск, 2015

Рассматриваемые вопросы

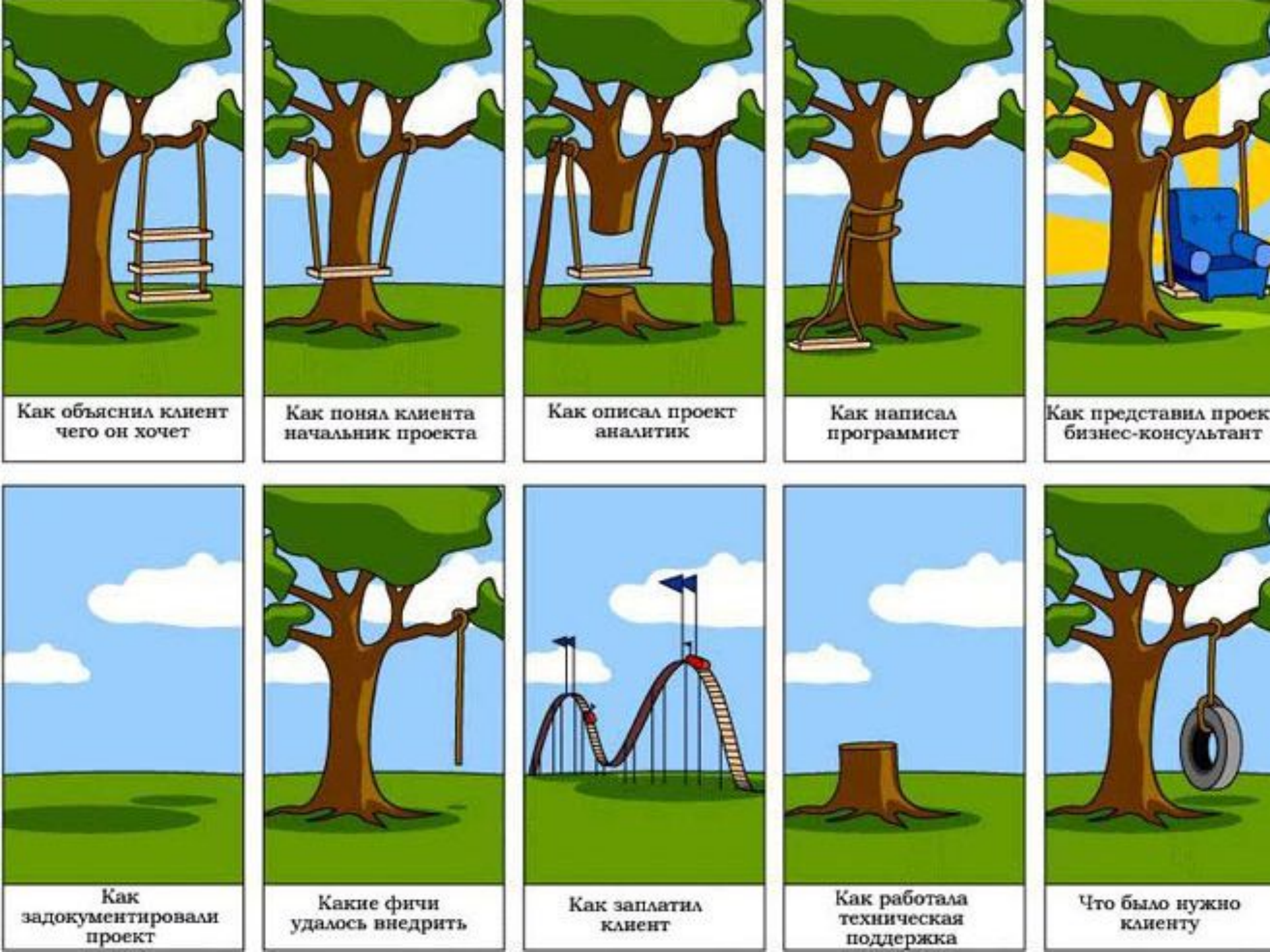
- Обзор методологий проектирования: классические методологии проектирования, гибкие методологии
- Достоинства и недостатки используемых методологий
- Методология ITECH.group
- Из опыта компании: где и когда следует применять различные методологии?

Причины неуспешного завершения

- Неправильная оценка на начальном этапе
- Неполные требования, искаженные требования, непонимание заказчика
- Низкая степень вовлечения заказчика и конечных пользователей в процесс разработки
- Недостаточное обеспечение ресурсами
- Неэффективное планирование работ

Примеры самых долгих проектов

- Mac OS X была продемонстрирована на презентации в 1997 году, а выпущена в 2011 году
- Windows Vista – планировалась к выпуску в 2003 году, была выпущена в 2006 году
- Проект Xanadu – был начат в 1960 и завершен в 2014 году (общий срок проекта 54 года)


Как объяснил клиент чего он хочет
Как понял клиента начальник проекта
Как описал проект аналитик
Как написал программист
Как представил проек бизнес-консультант
Как задокументировали проект
Какие фичи удалось внедрить
Как заплатил клиент
Как работала техническая поддержка
Что было нужно клиенту

Основные этапы жизненного цикла системы

- **Определение требований** (подготовка документации, расчет экономической эффективности, заключение и подписание договоров)
- **Разработка спецификаций** (детальная проработка требований, детализация смет, расширенная постановка задачи)
- **Проектирование** (написание технического задания, разработка прототипа)
- **Реализация** (различные работы в зависимости от спецификации проекта)
- **Тестирование** (различные виды тестирования: функциональное, usability, нагрузочное)
- **Сопровождение** (устранение недочетов, ошибок, замечаний, адаптация к новым условиям)
- **Развитие проекта** (модификация, добавление новой функциональности, рефакторинг)
- **Вывод проекта из эксплуатации**

Идеальная модель ЖЦ

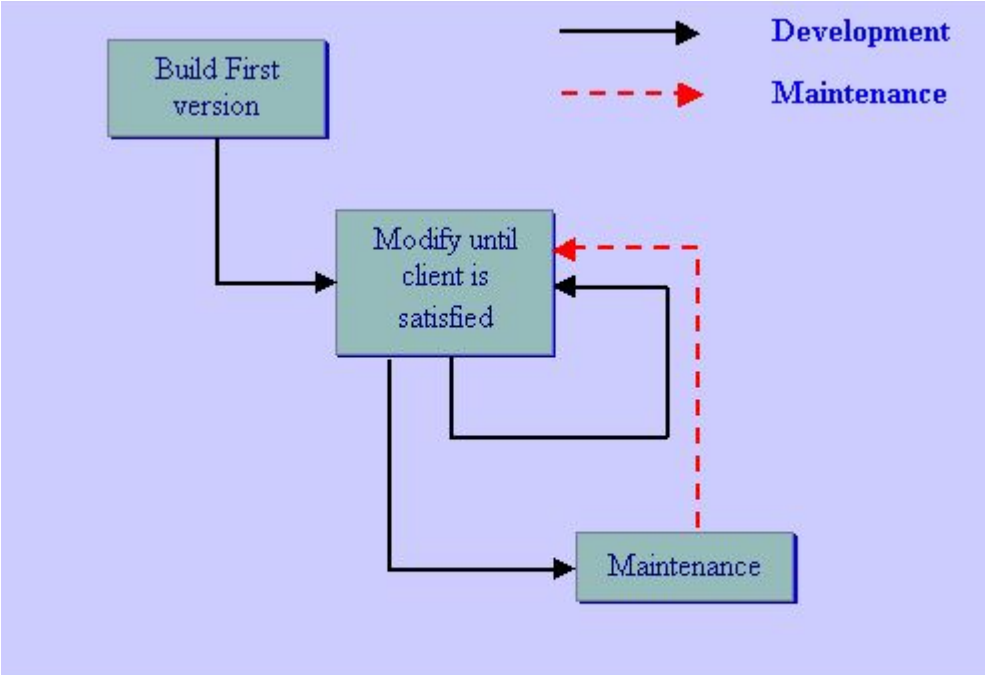
Классификация методологий

- Классические методологии проектирования
  - Модель Build&Fix
  - Водопадная (каскадная) модель проектирования
  - Итеративная модель
  - Спиральная модель
- Гибкие методологии проектирования
  - Методология XP
  - Манифест Agile/Методология SCRUM

Модель Build&Fix

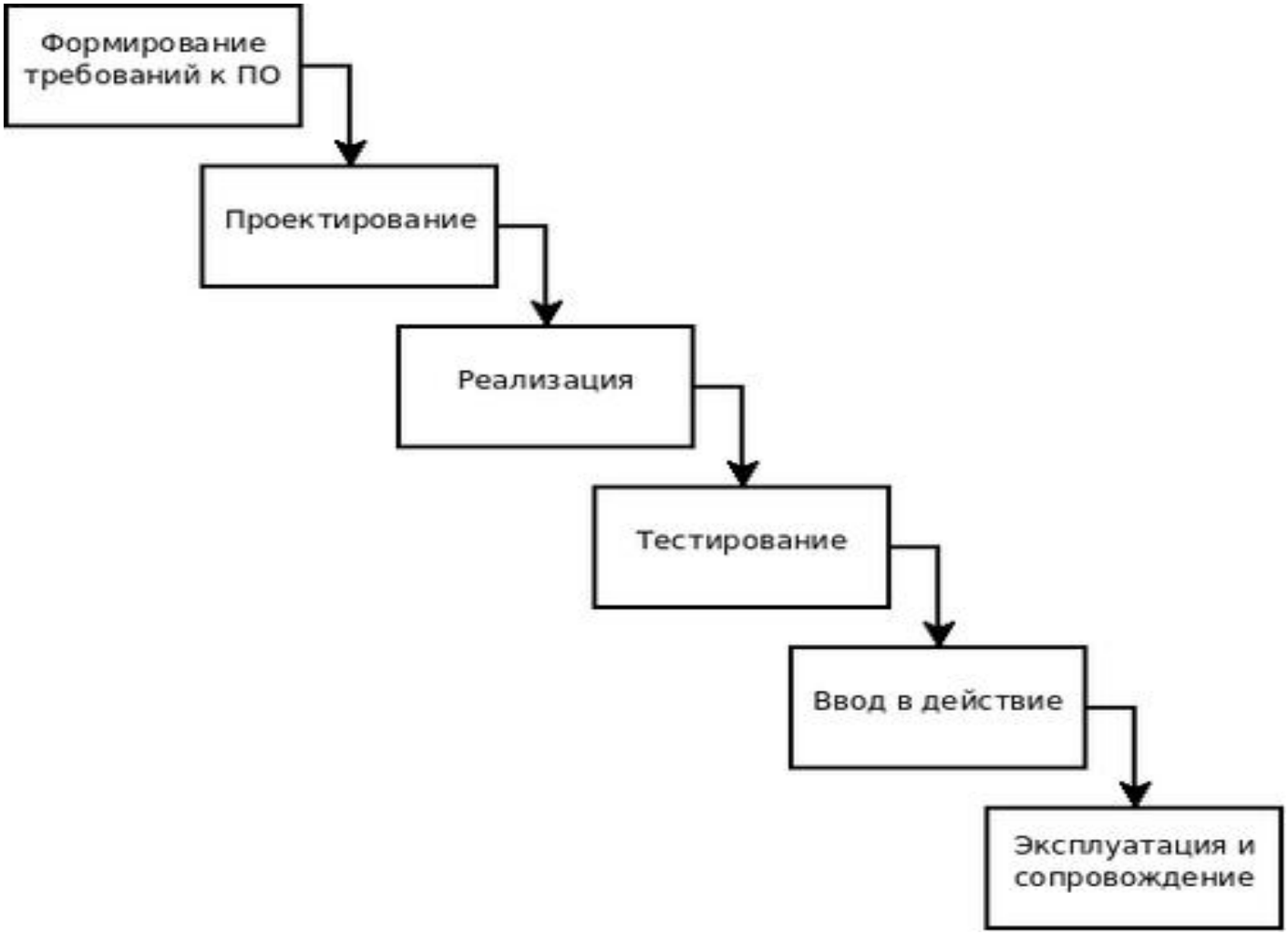
Данная модель имеет следующий алгоритм:

- Постановка задачи
- Выполнение
- Проверка результата
- При необходимости переход к первому пункту

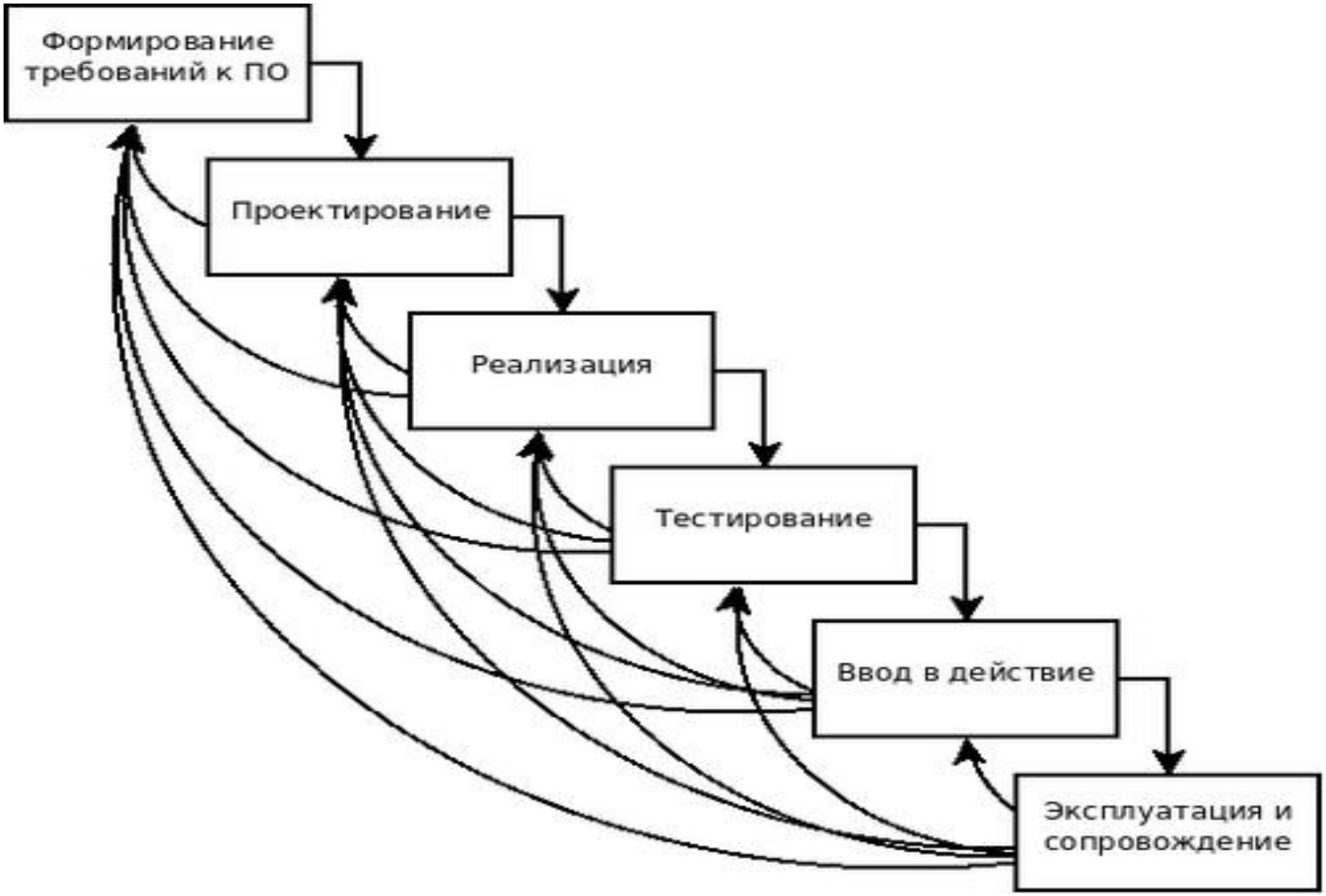


Область применения: очень простые проекты, сроком до 1 месяца, с четкими требованиями, известными еще на начальном этапе работ, команда – не более 3-х человек

Водопадная модель (идеальный вариант)



Водопадная модель в реальном применении



Достоинства и недостатки водопадной модели

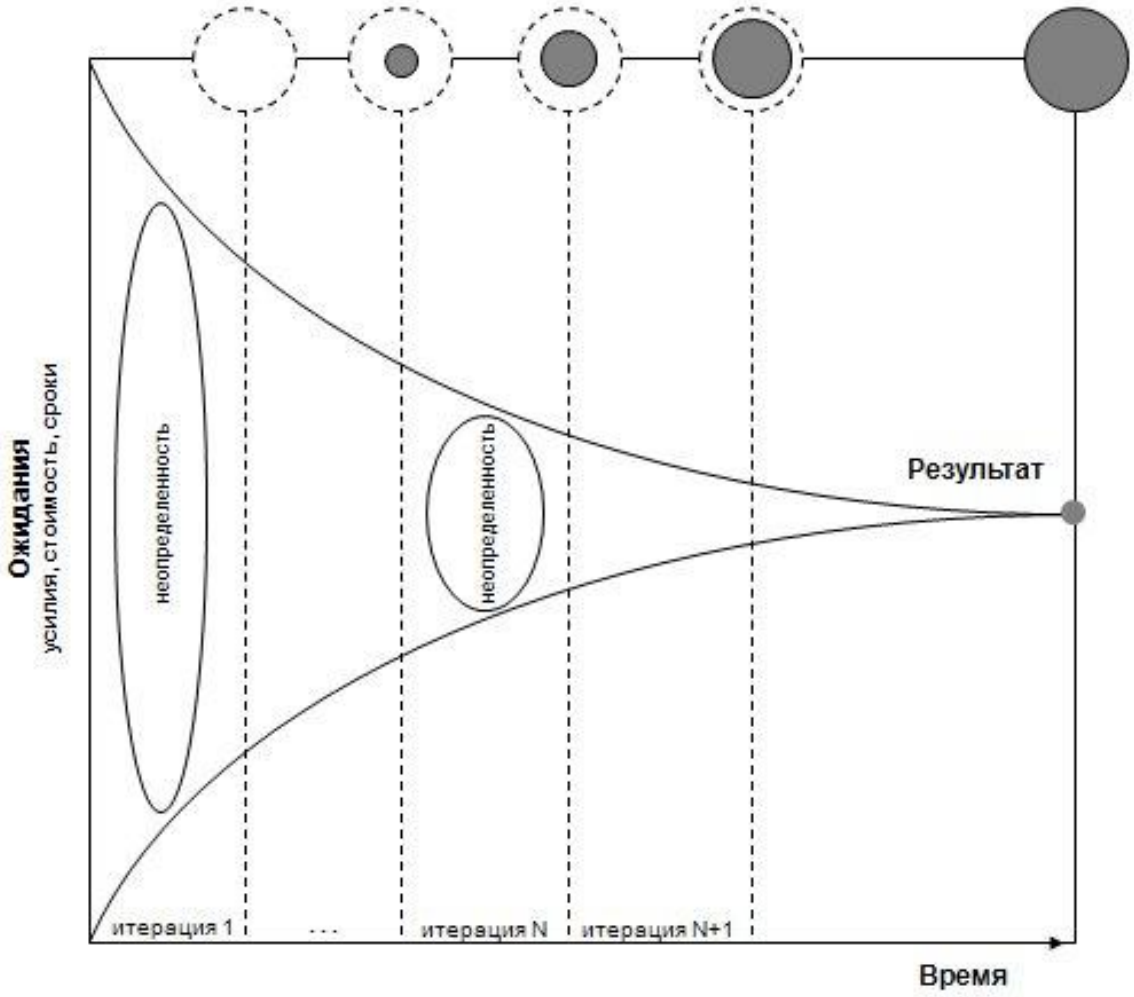
Достоинства:

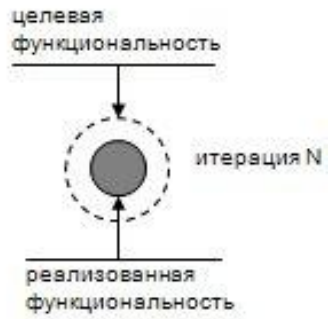
- Последовательное выполнение этапов проекта в строгом фиксированном порядке
- Позволяет оценивать качество продукта на каждом этапе

Недостатки:

- Жесткость модели
- «Запаздывание» и «Бесполезность»

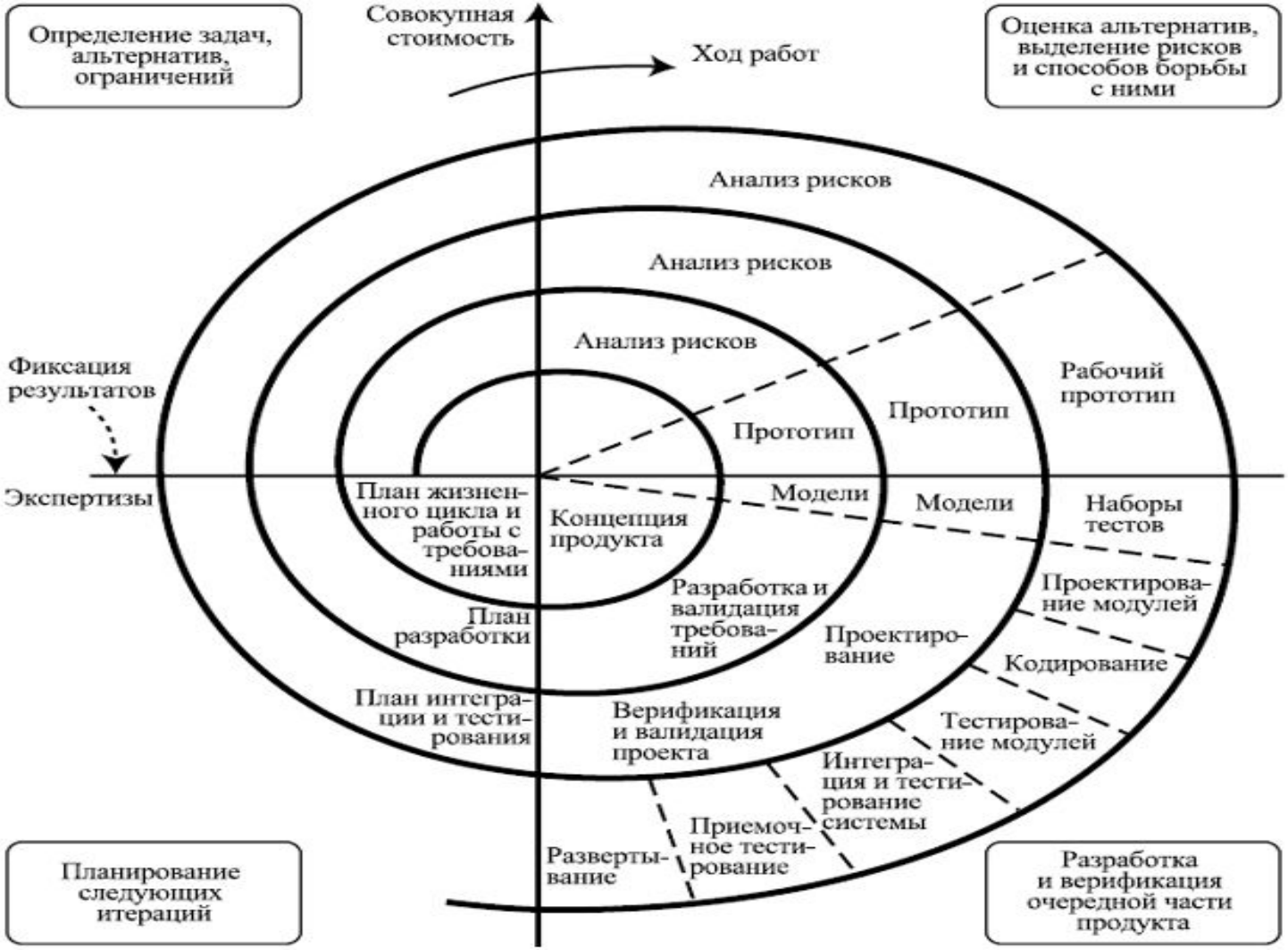
Итеративная модель





Организация работ при использовании итеративной модели

Спиральная модель



Цели решаемые в итеративной и спиральной модели

- **Сделать процесс более гибким** и дать возможность корректировать результаты полученные на предыдущем этапе
- **Вовлечь заказчика** в процесс разработки как можно раньше (в конце каждой итерации)
- **Сократить время** одной итерации до месяца или нескольких месяцев

Область применения классических методологий

- Фиксированный бюджет и сроки проекта
- Четкие требования, известные в самом начале использования проекта
- Сложное программное обеспечение (встроенные системы, системы реального времени, промышленные и военные системы)
- Конвейерный подход к разработке

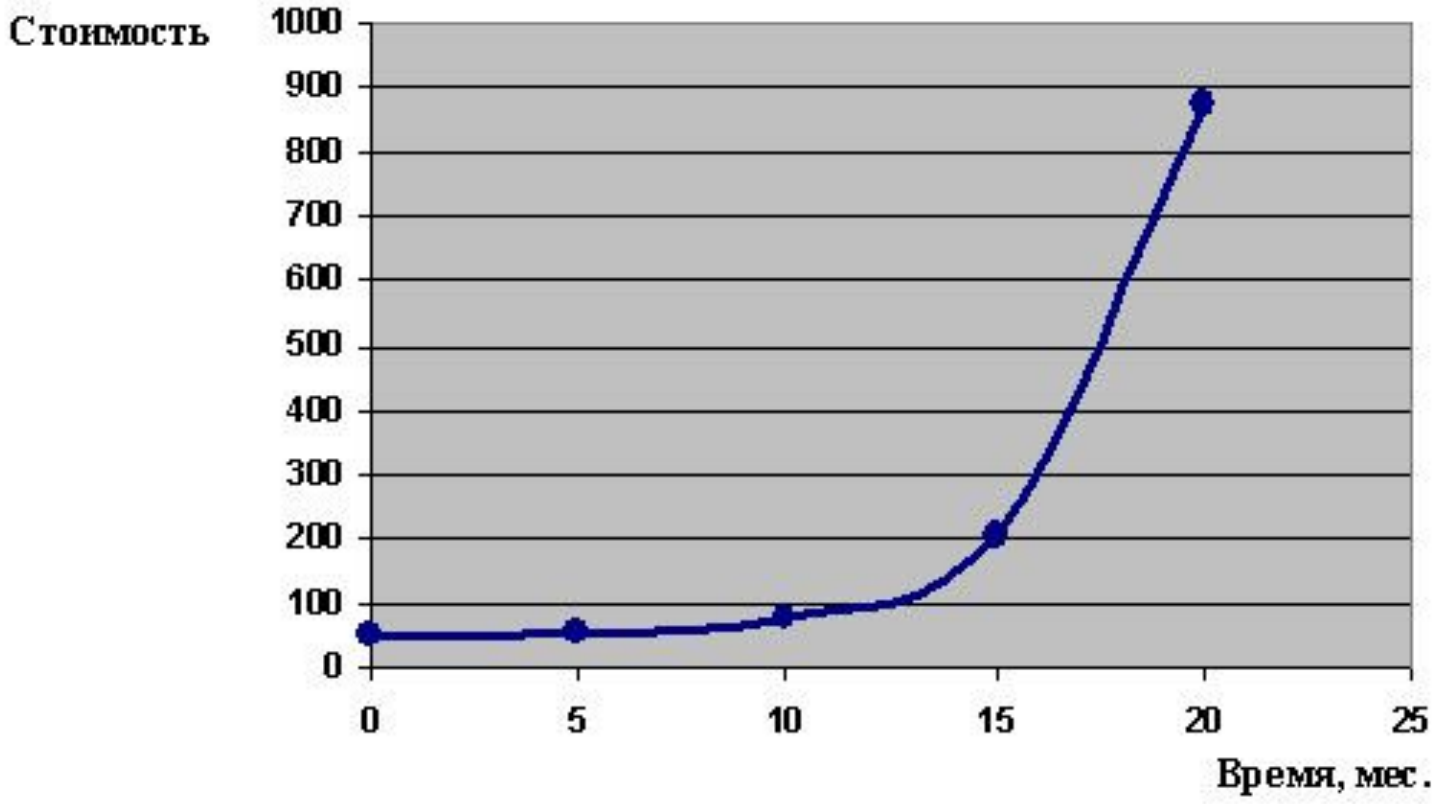
Недостатки классических методологий

- «Запаздывание» и «Бесполезность»
- Завышенная стоимость проекта
- Авральные работы в последние дни
- Слабая связь с заказчиком
- Необходимость точной оценки в начале проекта

Когда не использовать классику?

- Для стартапов, в которых нет четкого понимания сроков и целей
- Когда цели, сроки и бюджет проекта может меняться по мере его развития
- Когда сложно оценить всевозможные риски на начальных этапах
- Когда нет большого опыта работы над подобными проектами («нетиповой» для исполнителя проект)

Основная проблема классики



Стоимость внесения изменений в проект экспоненциально нарастает к концу проекта, а значит, выполнение всех рискованных работ необходимо планировать на ранние стадии проекта

Практические проблемы разработки

- Изменение требований непосредственно в процессе разработки
- Нечеткое распределение ответственности за выполняемую работу и ее результат
- Наличие непрерывного потока мелких, «быстрых», наваливающихся требований, отвлекающих разработчиков и менеджеров от основного направления работ
- Как следствие, срыв сроков, раздувание бюджетов, потеря качества

Методология XP

- XP – экстремальное программирование
- **Дата появления** - была впервые предложена в 1996 году
- **Автор методологии** - Кент Бэк
- **Основные идеи методологии**
  - усовершенствовать взаимосвязь разработчиков
  - упростить проектные решения
  - усилить обратную связь с заказчиком
  - проявлять больше активности

Базовые идеи методологии

Взаимодействие с заказчиком

Общение

3) Обратная связь

1) Взаимосвязь

ЗАКАЗ

РАЗРАБОТКА

Проектные решения

Заказчик

Разработчики

2) Проще

Простота

4) Активность

Храбрость

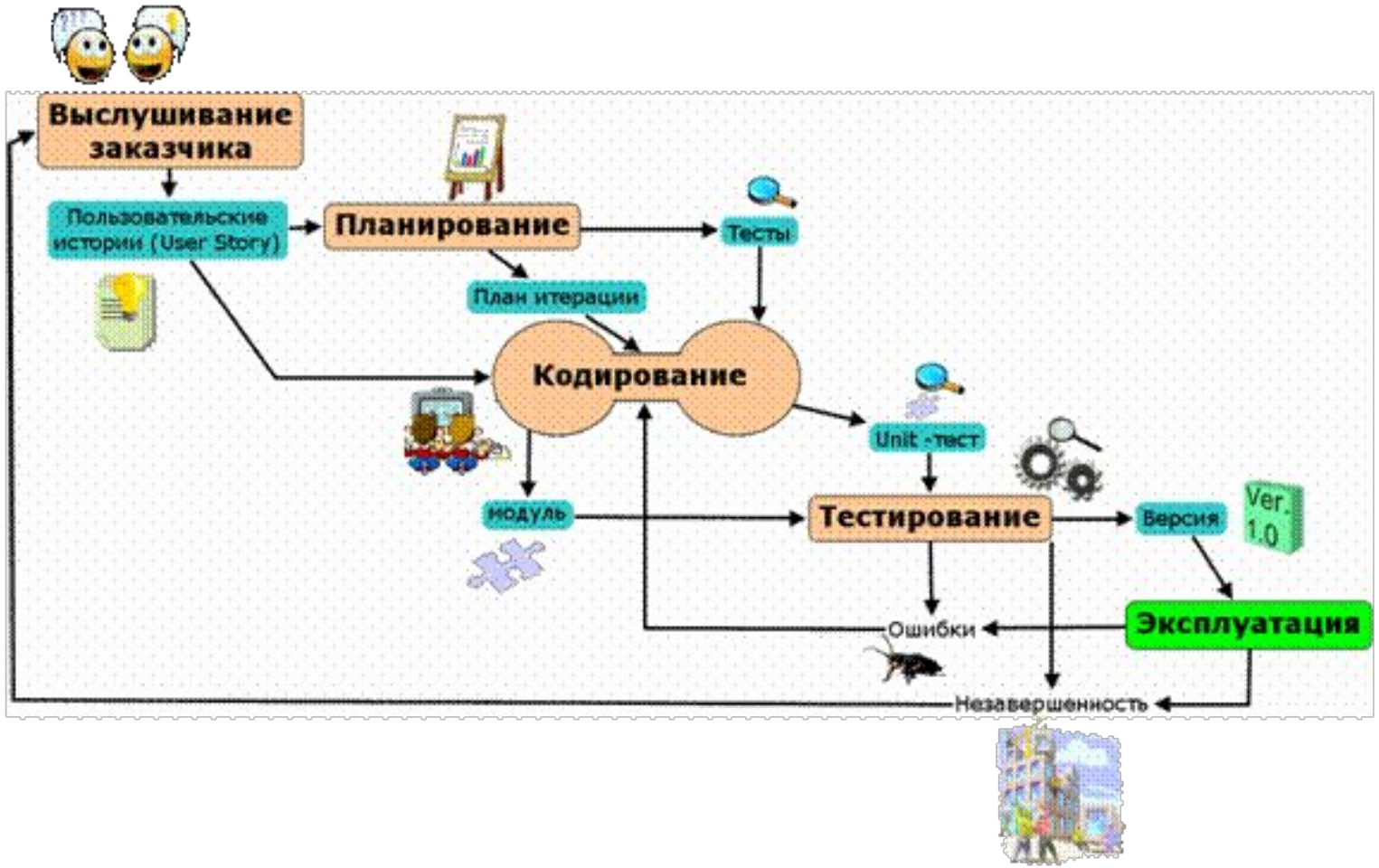
Основные принципы XP

1. Ежедневное планирование задач и ежедневный выпуск релизов
2. Тестирование до начала разработки – сначала пишется тест, затем код
3. Парное программирование
4. Постоянная переработка кода (рефакторинг)
5. Простота разработки

Основные принципы XP

6. Коллективное владение кодом
7. Постоянно продолжающаяся интеграция
8. Обязательное присутствие заказчика на рабочей площадке
9. Сорокачасовая рабочая неделя (без авралов и переработок)
10. Почасовая оплата (отсутствие жестких сроков и бюджетов)

Модель одной итерации XP



Область применения XP

Что такое Agile?

- **Гибкая методология разработки** (англ. Agile software development) – это набор принципов и правил, в рамках которого осуществляется разработка ПО.
- **Методология Agile** – это семейство процессов разработки, а не единственный подход к разработке программного обеспечения
- Ценности и принципы Agile методологии закреплены в документе '**Agile Manifesto**', принят в 2003 году

Ценности Agile

- **личности** и **их взаимодействия** важнее, чем процессы и инструменты
- **работающее программное обеспечение** важнее, чем полная документация
- **сотрудничество с заказчиком** важнее, чем контрактные обязательства
- **реакция на изменения** важнее, чем следование плану

Методология SCRUM

Scrum — это методология управления проектами, которая построена на принципах тайм-менеджмета. Основной ее особенностью является вовлеченность в процесс всех участников, причем у каждого участника есть своя определенная роль

Роли в SCRUM

- **Scrum Master -** отвечает за успех Scrum в проекте. SM является интерфейсом между менеджментом и командой
- **Product Owner -** это человек, отвечающий за разработку продукта (представитель заказчика)
- **Team** – команда разработчиков

Обязанности Scrum Master

- Создает атмосферу доверия
- Участвует в митингах в качестве инициатора
- Устраняет препятствия
- Делает проблемы и открытые вопросы видимыми
- Отвечает за соблюдение практик и процесса в команде

Обязанности Product Owner

- Отвечает за формирование product vision
- Управляет ROI
- Управляет ожиданиями заказчиков и всех заинтересованных лиц
- Координирует и приоритизирует Product backlog
- Предоставляет понятные и тестируемые требования команде
- Взаимодействует с командой и заказчиком
- Отвечает за приемку кода в конце каждой итерации

Обязанности команды

- Отвечает за оценку элементов баклога
- Принимает решение по дизайну и имплементации
- Разрабатывает софт и предоставляет его заказчику
- Отслеживает собственный прогресс (вместе со Скрам Мастером).
- Отвечает за результат перед Product Owner

Процесс SCRUM

Что такое Sprint?

- Sprint – это итерация в SCRUM
- Длительность спринта – от одной недели до одного месяца
- Каждый Sprint – это меленький «водопад»
- Результатом спринта является готовая версия продукта - build

Backlog

- **Product Backlog** - это приоритезированный список имеющихся на данный момент бизнес-требований и технических требований к системе.
- **Sprint Backlog** содержит функциональность, выбранную Product Owner из Product Backlog. Все функции разбиты по задачам, каждая из которых оценивается командой.

Жизненный цикл спринта

Планирование спринта, митинг первый

- ***Участники***: команда, Product Owner, Scrum Master, пользователи, менеджмент
- ***Цель***: Определить цель спринта (Sprint Goal) и Sprint Backlog - функциональность, которая будет разработана в течение следующего спринта для достижения цели спринта.
- ***Артефакт***: Sprint Backlog

Планирование спринта, митинг второй

- ***Участники***: Скрам Мастер, команда
- ***Цель***: определить, как именно будет разрабатываться определенная функциональность для того, чтобы достичь цели спринта. Для каждого элемента Sprint Backlog определяется список задач и оценивается их продолжительность.
- ***Артефакт***: в Sprint Backlog появляются задачи

Burndown диаграмма

Когда бы помог SCRUM?

- Куча денег и времени ушла на проработку ТЗ, но по ходу работы над проектом поменялась концепция или бизнес-процессы. Доводить проект до конца в том виде, как описано в ТЗ — нет смысла. Деньги на ТЗ выброшены напрасно. Разработчик отказывается вносить изменения по ходу работы, ссылаясь на ТЗ.

- Разработчик показывает проект в последний день перед запуском. Однако все сделано не так, как вы себе это представляли. Нужна значительная переделка. Разработчикпо-своему трактует описанные в ТЗ требования и отказывается вносить изменения в проект на этом основании.

- Нужно запустить костяк интернет-проекта с минимально-возможным бюджетом и сроками. Дополнительные функции разрабатывать уже после запуска, когда проект начнёт отбивать начальные инвестиции.

Положительные стороны SCRUM

- пользователи начинают видеть систему спустя всего несколько недель и могут выявлять проблемы на ранних стадиях разработки программного продукта;
- интеграция технических компонентов происходит в ходе каждого спринта и поэтому проблемы проекта (если они возникают) выявляются практически сразу;
- в каждом спринте команда фокусируется на контроле качества;
- гибкая работа с изменениями в проекте на уровне спринта

Когда SCRUM не подходит?

- ГОС заказы и тендеры, где изначально присутствуют фиксированные сроки и цены
- Низкая квалификация и ответственность команды исполнителей
- Некомпетентный менеджер проекта

Достоинства гибких методологий

- Риски распределены между заказчиком и исполнителем
- Хорошая обратная связь с заказчиком за счет более коротких итераций
- Изменение требований к проекту во время его разработки

Недостатки гибких методологий

- Необходимо полное доверие заказчика к исполнителю
- Заказчик не имеет полной гарантии получить то, что он ожидал
- Необходимо больше усилий на управление проектом

Опыт использования в ITECH.group

- В данный момент используется **классическая водопадная** модель
- Количество проектов в работе – от 25 до 40 проектов на разных стадиях
- Команда разработчиков – 25 человек (менеджеры, проектировщики, дизайнеры, верстальщики, программисты, тестировщики)
- Инструменты автоматизации – JIRA, BaseCamp, GitLab

Основные этапы работ

- Предварительный этап: Оценка проекта на этапе Presale, составление документации для начала работы
- Этап 1: Аналитика
- Этап 2: Проектирование
- Этап 3: Дизайн
- Этап 4: Верстка + Front-end
- Этап 5: Программирование и интеграция верстки
- Этап 6: Тестирование, наполнение контентом и передача проекта на сопровождение

Договор: на что следует обратить внимание разработчику?

- **Контактные лица**: обязательно должны быть указаны контакты менеджера и руководителя
- **Правила сдачи и приемки** работ: работы подтверждаются актом выполненных работ, который выставляется заказчику после выполнения всех работ
- **Гарантийные обязательства**: должны быть прописаны условия гарантийного обслуживания
- **Детальная смета и сроки**: необходимо включать в договор смету и сроки работ, которые разделены по конкретным задачам

Договор: гарантийные обязательства

- На созданный в соответствии Договору №__-МК от 5 августа 2014г. web-сайт устанавливается гарантия на срок 1 (Один) год с момента его передачи по акту сдачи-приёмки выполненных работ, в течение которого Исполнителем от Заказчика принимаются претензии в отношении качества выполненных работ.
- Гарантия распространяется на web-сайт, созданный в соответствии с Техническим заданием на разработку и все дополнительные сервисы и модули, реализованные Исполнителем в рамках дополнительных работ.
- Исполнитель гарантирует правильную работу web-сайта согласно Техническому заданию, отсутствие ошибок в работе сервисов web-сайта при условии соблюдения Технических требований для работы интернет-сайта, предоставленных Исполнителем в составе Технического задания, а так же своевременное, не позднее 2 (Двух) рабочих дней с момента получения обращения Заказчика, исправление недоработок, возникших в процессе эксплуатации.
- Гарантийное обязательство не распространяется на сторонние сервисы, от которых может зависеть правильная работа web-сайта.

Когда гарантия не работает

Гарантийное обслуживание не производится в следующих случаях:

- Если в программный код были внесены изменения сторонними лицами;
- Если неправильная работа web-сайта обусловлена ошибками в работе сервера, которая привела к повреждению файлов сайта, либо структуры базы данных;
- Если web-сайт был подвержен при атаке злоумышленников через хостинг, на котором web-сайт расположен, либо в связи с неправильной работой с web-сайтом (простые пароли, предоставление доступа к панели администрирования или хостингу третьим лицам, самостоятельное внедрение стороннего кода);
- Если ошибку в работе web-сайта повлекло стороннее ПО, расположенное с ним на одном сервере;
- Если Заказчик самостоятельно, либо через третьи лица:
  - пытался включить в работу web-сайта сторонние сервисы;
  - произвёл изменения css, javascript файлов и\или их имён как через файловую систему, так и через панель администрирования;
  - произвёл изменения изображений и\или их имён, используемых для отображения шаблонов как через файловую систему, так и через панель администрирования;
  - произвёл изменения шаблонов web-сайта и\или их частей как через файловую систему, так и через панель администрирования;
  - использовал в контентной части web-сайта элементов, не предусмотренных страницей стилей;
  - использования при заполнении контента на web-сайте визуального редактора

Пример указания сметы и сроков

Указание стоимости различных видов работ

Наименование работы	Цена, руб.
Расширенная постановка задачи	16 000
Навигационный прототип	15 000
Дизайн-концепция	69 000
Итого стоимость всех работ:	**100 000**

Указание сроков различных видов работ

Наименование работы	Ответственная Сторона	Срок рабочих дней.
Расширенная постановка задачи	Исполнитель	2
Согласование	Заказчик	1
Навигационный прототип	Исполнитель	2
Согласование	Заказчик	1
Дизайн-концепция	Исполнитель	6
Согласование	Заказчик	1
Итого срок выполнения всех работ с учётом согласования:		**13**

Этап 1: Аналитика

- Составление расширенной постановки задачи
- Формирование целевой аудитории
- Анализ текущего сайта и анализ конкурентов
- Анализ современных трендов
- Анализ систем статистики (если они были установлены на текущем сайте)
- Разработка структуры сайта
- Разработка списка предлагаемых сервисов

Этап 2: Проектирование

- Разработка **интерактивного прототипа** Web-сайта с перечнем всех страниц и сервисов (имитация работы сервисов)
- Написание **технического задания** (описание всех сервисов и разделов сайта + нефункциональные требования)
- Разработка **дизайн-концепции** и ее защита перед заказчиком (в виде презентации)

Нефункциональные требования

- Требования к верстке (перечень браузеров и устройств)
- Требования к программному обеспечению
- Требования к БД
- Требования к поисковой оптимизации
- Требования к пиковым нагрузкам
- Требования к аппаратному обеспечению
- Требования к защите информации

Этап 3: Дизайн

- Подготовка и передача дизайн-макетов в формате jpg
- **Исходные макеты не передаются** заказчику до тех пор, пока не будет подписан акт выполненных работ и не будет проведена оплата

Этап 4: Верстка

- Статичная верстка дизайн-макетов
- Программирование логики front-end сервисов
- Размещение результатов верстки на тестовом домене разработчика
- Изолирование тестового домена от поисковых систем и защита общего доступа паролем

Этап 5: Интеграция и программирование

- Интеграция сверстанных шаблонов в CMS
- Программирование сервисов
- Программирование интеграции со сторонними сервисами и системами

Этап 6: Внедрение

- Тестирование сайта и исправление ошибок
- Наполнение сайта контентом
- Подготовка закрывающих документов:
  - Акт выполненных работ
  - Счет на оплату
  - Акт сверки
  - Акт передачи проекта на сопровождение
- Выливка сайта на домен заказчика только после подписания акта и оплаты!!!

Инструменты управления

- Автоматизированные системы
  - JIRA – для постановки задач и контроля их исполнения
  - BaseCamp – система общения с клиентом
  - gitLab – система управления версиями
- Диаграмма процессов - для описания регламентов для бизнес процессов
- Диаграмма Ганта – инструмент календарного планирования
- Матрица ответственности
- Карточки рисков

Построение диаграммы процесса с учетом ролей участников проекта

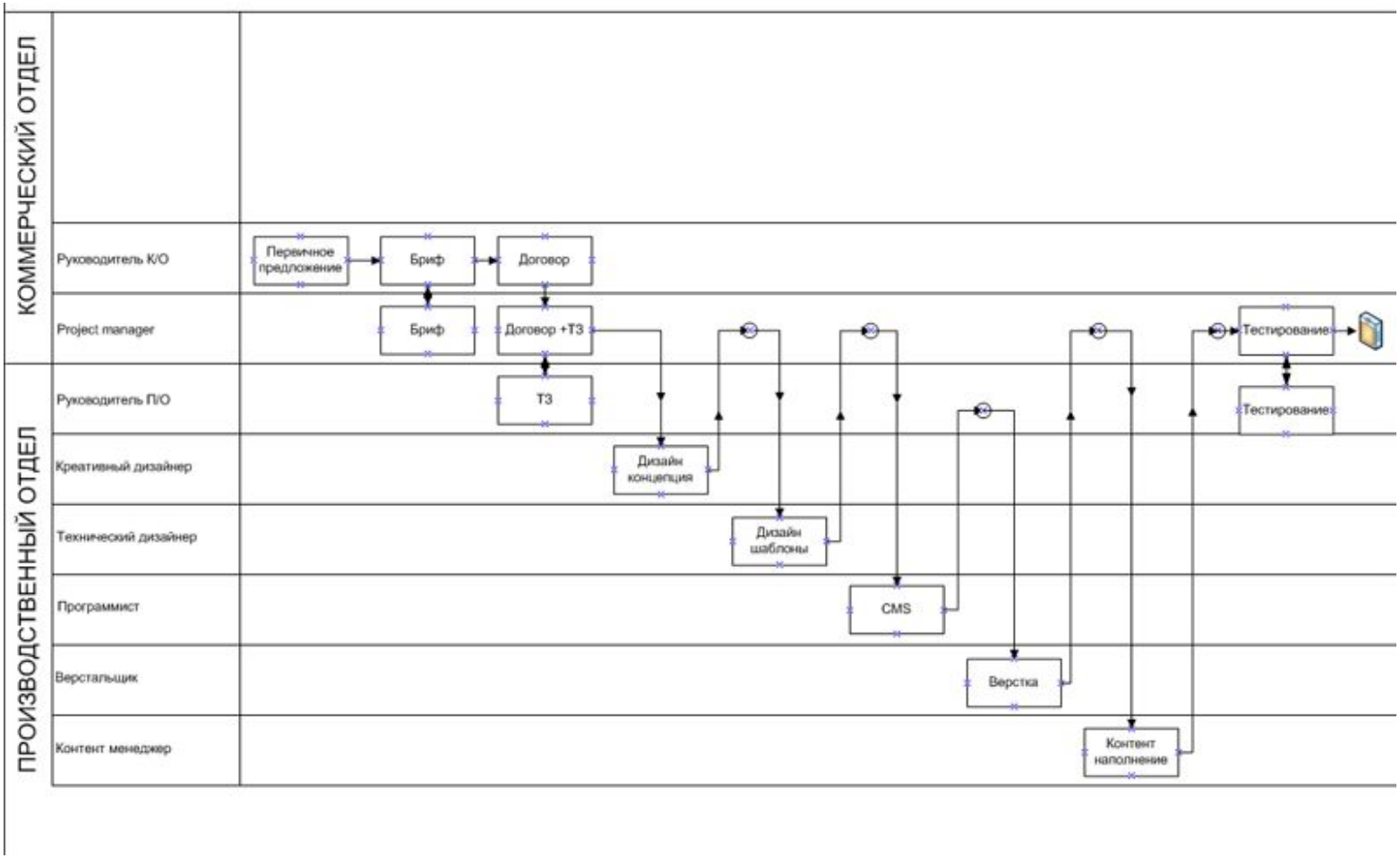
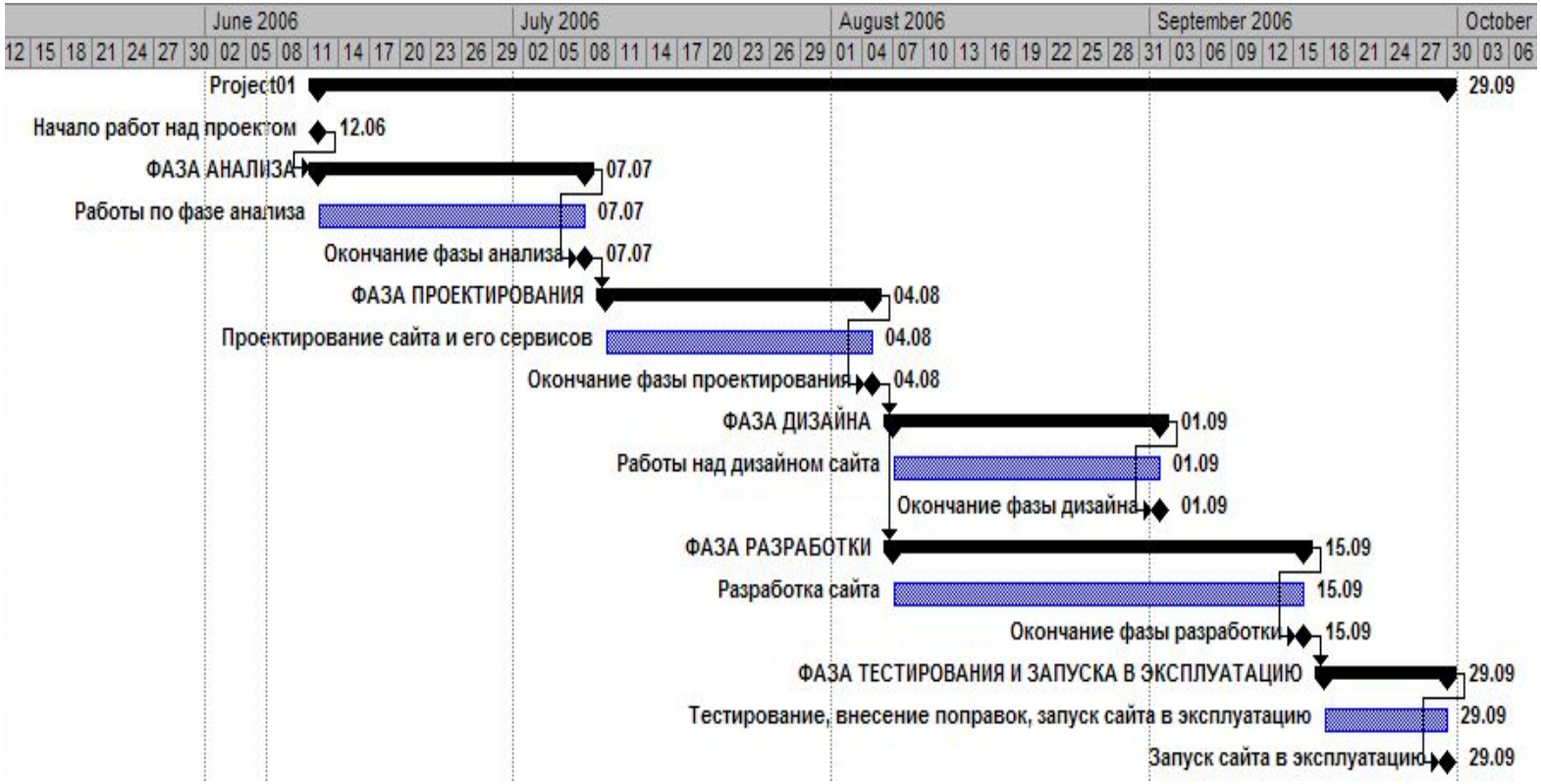


Диаграмма Ганта



Матрица ответственности: кто отвечает за продвижение проекта

Руководство	**Руководитель отдела анализа**	**Руководитель отдела проектирования**	**Руководитель отдела разработки и тестирования**
Лидер 1 (проекты 1,2,3)	Исполнитель	Исполнитель	Исполнитель
Лидер 2 (проекты =4,5,6)	Исполнитель	Исполнитель	Исполнитель
Лидер 3 (проекты 7,8,9)	Исполнитель	Исполнитель	Исполнитель

Карточки рисков: как не наступать на грабли дважды

- Описание проблемы
- Возможные последствия
- Что было предпринято
- Реальные последствия
- Что надо было предпринять «до», для избежания проблемы
- Рекомендации
  - Исправление договора
  - Исправление документации
  - Изменение процессов
  - Разработка инструкций
  - и т.д.