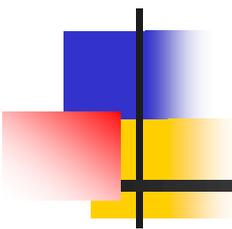
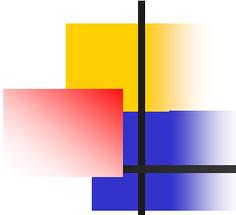


Управление разработкой программного обеспечения



Александр Воронов

Video Group
CS MSU Graphics & Media Lab



Содержание

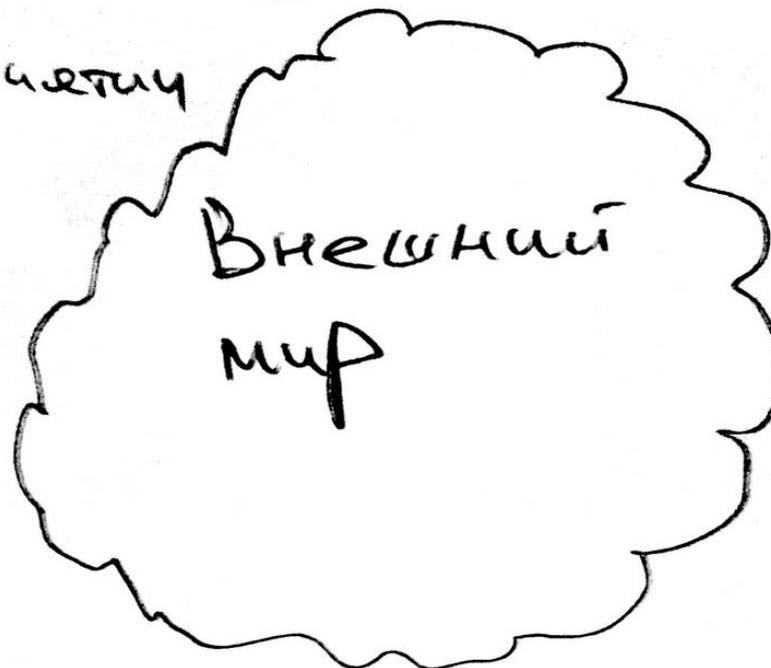
- Введение
- Брукс: Мифический человеко-месяц
- Йордон: Путь камикадзе
- Пара слов про Agile
- Заключение

Мир глазами программиста



С другой стороны

Начальник
на предприятии



Я хочу компьютеризировать
производство, чтоб мои
сотрудники работали
эффективнее.

Заказчик — исполнитель

Начальник
на предприятии

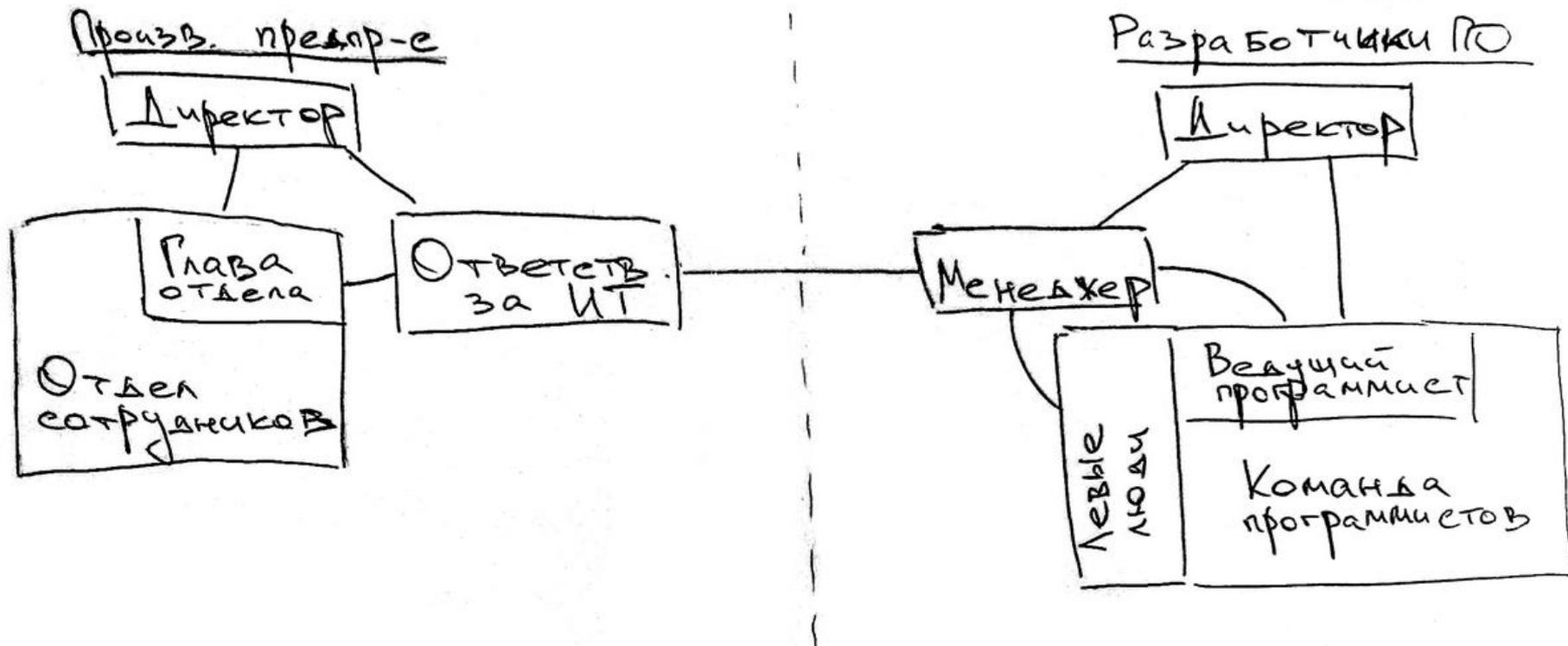


Программист



Пропасть

Схема отношений



Особенности отрасли

- Разработка программного обеспечения — молодая отрасль
- Основная часть себестоимости продукта — зарплата сотрудников
- Сложность оценки объёма работ
- Сложность объективной оценки качества и сравнения с аналогами

Особенности должности

- Необходимость планирования
 - Когда будет готов конечный продукт?
- Ответственность
 - Принятие решений
 - Контроль качества
 - Разгребание проблем
- Коммуникации
 - Видение ситуации в целом и обеспечение необходимой связи между участниками

Особенности должности

Планирование

- Все хотят знать, что получится в конце и когда этот конец настанет
- Программистам верить нельзя — они слишком оптимистичны
- Нужно угадать и предусмотреть
 - Время, деньги, количество людей
 - Промежуточные и конечный результаты
 - Возможные проблемы и пути их решения

Особенности должности

Ответственность

- Каждый программист хочет отвечать только за свой код
- У менеджера (ведущего разработчика) шире зона ответственности
 - Должно получиться то, что хочет заказчик
 - Несмотря ни на что, всё должно быть в срок
 - Не должно быть ситуаций, когда программисту нечего делать — он скучает
 - Если кто-то из программистов не справился — отвечать менеджеру

Особенности должности

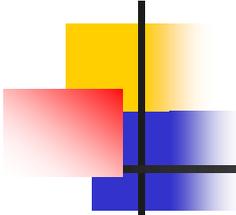
Коммуникации

- Мир глазами программиста и мир глазами заказчика — это разные вселенные
- Мир глазами разных программистов — разные планеты
- Программисты — люди умные, но порой своеобразные

Особенности должности

Коммуникации

- Здоровая атмосфера в коллективе — залог успеха проекта
- Плохая информированность участников коллектива — одна из частых причин провала



Содержание

- Введение
- **Брукс: Мифический человеко-месяц**
 - Планирование
 - Проектирование
 - Реализация
 - невыполнение плана
- Йордон: Путь камикадзе
- Пара слов про Agile
- Заключение

О книге

- **Фредерик Брукс**
 - Управлял разработкой OS/360 в IBM
 - Награждён Премией Тьюринга в 1999 году
- «Мифический человеко-месяц или Как создаются программные системы» — 1975 год (1995 — 2-е издание)



Планирование

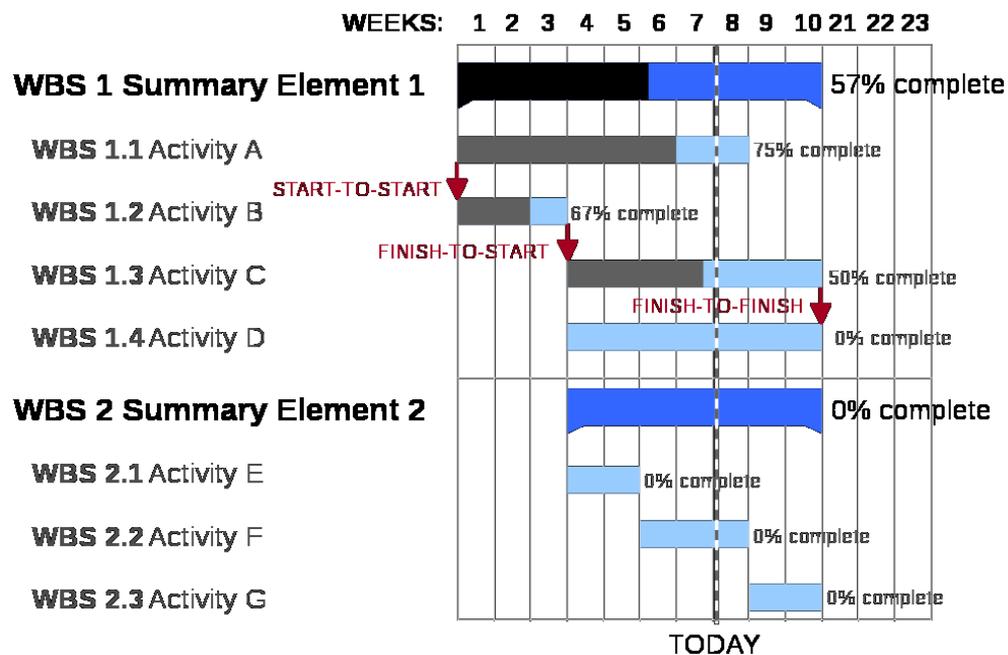
- Чтобы сделать всё в срок и по плану, нужно этот план хотя бы иметь
- В плане должны быть чётко обозначенные даты и вехи (milestones)
- Вехи — это конкретные достижения
- Задача занимает всё отведённое под неё время

Планирование

Критические пути

Диаграмма Ганта — график с критическими путями

- Последовательность действий
- Планирование рисков



Планирование

Планирование рисков

Закон Мёрфи: «Anything that can go wrong will go wrong»

Если при планировании проблеме было уделено мало внимания — она обязательно возникнет

Должны быть предусмотрены:

- Пути смягчения последствий неразрешимых проблем
- Минимальный гарантированный результат при каждой из таких проблем

Планирование

Планирование усилий

Правило Парето: 20% усилий обеспечивают 80% результата

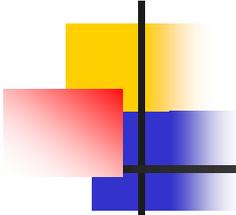
При разработке ПО:

- Проектирование — 30%
- Написание кода — 20%
- Тестирование компонентов — 25%
- Системное тестирование — 25%

Планирование

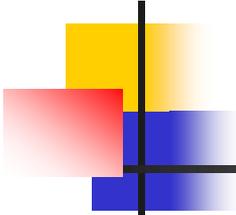
Планирование усилий

- Превращение программы в продукт — увеличение времени втрое
- Объём работ растёт как степенная функция размера программы v^a
По разным оценкам $a = 1,05-1,2$ или $1,5$
 v — некая метрика размера программы



Содержание

- Введение
- Брукс: Мифический человеко-месяц
 - Планирование
 - **Проектирование**
 - Реализация
 - невыполнение плана
- Йордон: Путь камикадзе
- Пара слов про Agile
- Заключение



Проектирование

- Качественное проектирование сокращает затраты на кодирование и тестирование
- Ошибки, возможные на этом этапе — самые дорогие

Проектирование

- Концептуальная целостность — самое важное для получения качественного продукта
- Продукт должен позволять решать задачи, а не просто предоставлять набор функций
- Ответственность за концептуальную целостность на одном человеке или группе единомышленников (аристократизм)
- Отделить архитектуру от реализации (варианты реализации должны быть предложены)

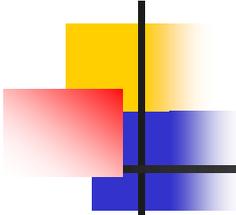
Проектирование Спецификации

- Описание 20% компонентов обозначает 80% требований
- Необходимо и формальное, и текстовое описание проекта (одно из них выбирается основным)
- Вместе со структурой проекта планируется структура документации

Проектирование Спецификации

Для каждого модуля должны быть описаны:

- Требуемый функционал
- Ожидаемые и допустимые характеристики (память, скорость)
- Взаимодействие с другими модулями



Содержание

- Введение
- Брукс: Мифический человеко-месяц
 - Планирование
 - Проектирование
 - **Реализация**
 - невыполнение плана
- Йордон: Путь камикадзе
- Пара слов про Agile
- Заключение

Документация

- Причина плохой документации — не только лень, но и неуверенность в текущих решениях (недостаток проектирования)
- Документации часто не хватает обзорности
- Важно отвечать не только на вопрос «как?», но и «почему?»

Тестирование

- После любого изменения нужно проводить серию регрессионных тестов
- Для каждого компонента должны быть отдельные тесты
- При интегральном тестировании компоненты должны добавляться по одному

Сопровождение

- Сопровождение — это исправление ошибок и усовершенствование функционала
- Возможность внесения изменений необходимо предусматривать
- Зависимость числа модулей от номера версии — линейная, числа затронутых изменениями модулей — экспоненциальная
- Исправления имеют тенденцию к разрушению структуры системы

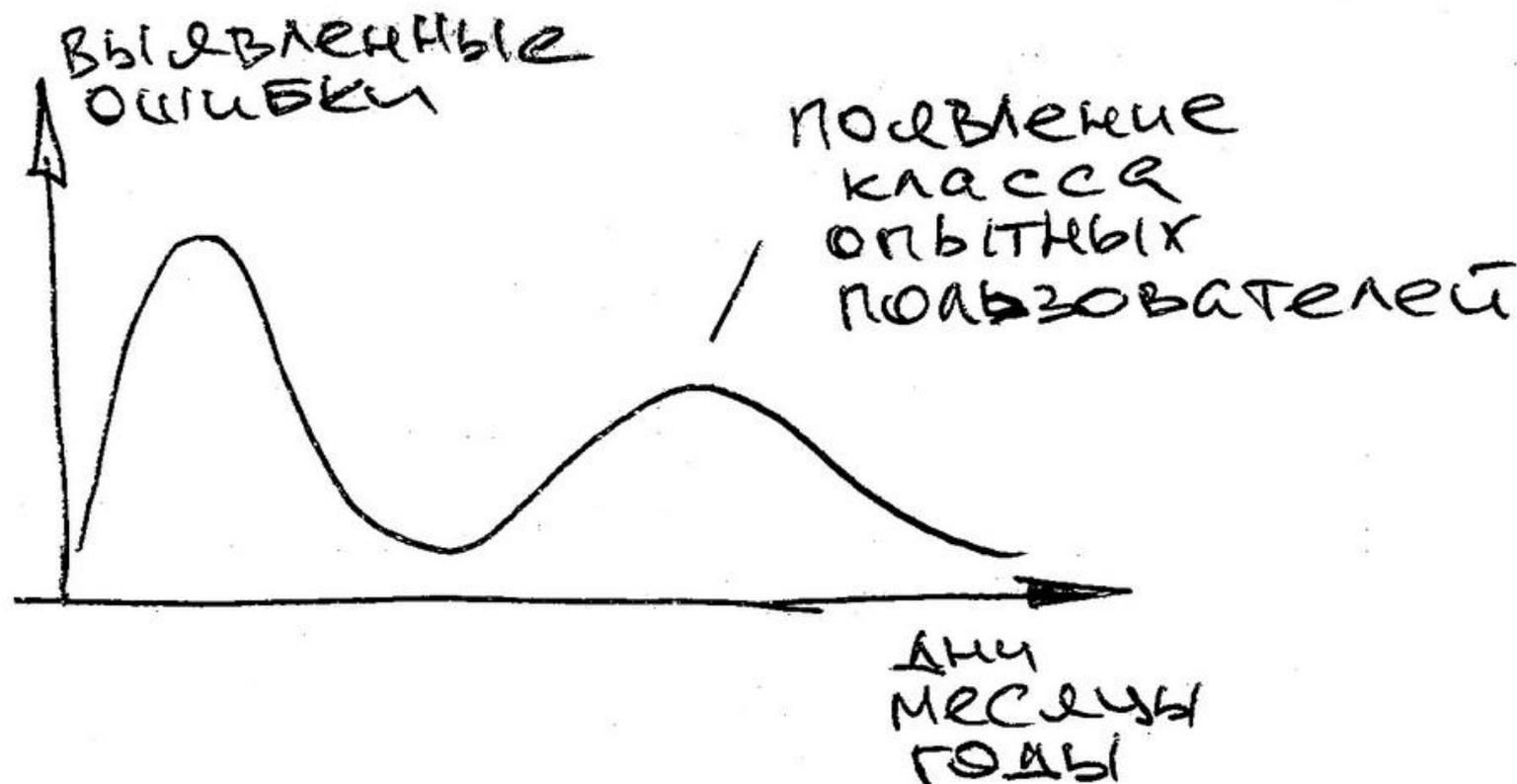
Сопровождение

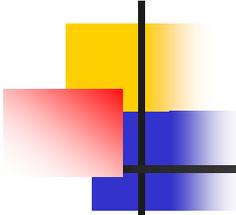
Выявление ошибок

- Чем больше пользователей, тем больше найденных ошибок
- Значительная часть ошибок выявляется сразу после выпуска продукта; очередной всплеск происходит по прошествии некоторого времени
- При исправлении одной ошибки с вероятностью 20–50% вносится другая

Сопровождение

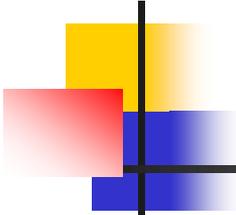
Выявление ошибок





Содержание

- Введение
- Брукс: Мифический человеко-месяц
 - Планирование
 - Проектирование
 - Реализация
 - **Невыполнение плана**
- Йордон: Путь камикадзе
- Пара слов про Agile
- Заключение



Невыполнение плана

- « — Как оказывается, что проект запаздывает на год?
— Сначала он запаздывает на один день. »

Фредерик Брукс

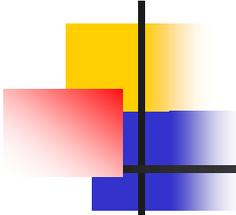
Невыполнение плана

- Одна из основных причин срыва сроков — неполнота или противоречивость информации о текущем состоянии
- Полезно иметь две оценки текущего состояния:
 - от ответственного
 - от исполнителей

Невыполнение плана

С точки зрения программиста

- Не бойтесь сообщать о том, что не укладываетесь в сроки
- По закону Мёрфи ситуация имеет свойство развиваться от плохого к худшему
- Нормальный руководитель подобную ситуацию уже предусмотрел



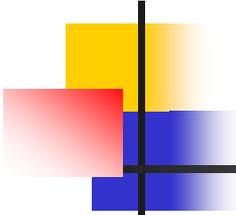
Невыполнение плана

С точки зрения руководителя

- Нехватка календарного времени — основной тип провалов
- Добавление людей в проект только затягивает сроки
 - Нарушение работы
 - Перекраивание задач
 - Обучение
 - Дополнительное общение

Брукс: выводы

- За 36 лет устарели только технологии, идеи и принципы по-прежнему актуальны
 - Произошла микрокомпьютерная революция
 - Значительно вырос рынок, изменился его характер
- Несмотря на большое количество исследований и накопленного опыта, кризис разработки ПО продолжается



Содержание

- Введение
- Брукс: Мифический человеко-месяц
- **Йордон: Путь камикадзе**
- Пара слов про Agile
- Заключение

О книге



- Эдвард Йордон
 - Учёный в области теории вычислительных систем
 - Независимый консультант
- «Путь камикадзе» — 1997 год
 - Оригинальное название:
«Death March. The Complete Software Developer's Guide to Surviving *Mission Impossible* Projects»

Безнадёжные проекты

Когда совсем всё плохо

- Один из параметров проекта как минимум вдвое отличается от нормы
 - Календарные сроки
 - Количество разработчиков
 - Бюджет или ресурсы
 - Требуемые функции или технические характеристики
- Вероятность провала выше 50%

Безнадёжные проекты

Откуда они берутся

- Неверное планирование
- Желание перехватить заказ у более крупной компании
- Переоценка собственных способностей
- Политика и маркетинг
- Стремление захватить мир

Безнадёжные проекты

Причины участия

- Большое вознаграждение
- Синдром покорителей Эвереста
- Глупость
- Перспективность
- Безальтернативность
- Оптимизм

Безнадёжные проекты

Характеристика проекта



Безнадёжные проекты

Роль менеджера

Два типа менеджеров, ответственных за проект:

- Тот, кто выжимает команду
 - Мотивирован на деньги
 - Не планирует в будущем продолжать работу с командой
- Тот, кто защищает команду
 - Мотивирован на деньги и результат
 - Рассчитывает сохранить нормальные отношения с командой

Безнадёжные проекты

Переговоры

При обсуждении условий проекта есть три переменных:

- Календарные сроки
- Ресурсы (в основном: люди и бюджет)
- Качество (функционал, требования, условия эксплуатации)

Безнадёжные проекты

Переговорные игры

- Удвой и добавь ещё
- Обратное удвоение
- Угадай число, которое я задумал
- Игра на понижение
- Туманы и миражи
- Фиктивное качество

Безнадёжные проекты

Формирование требований

- Формулировка требований и расстановка приоритетов
- Прояснение качества «достаточно хороший»
- Правило 20/80

Безнадёжные проекты

Использование новых технологий

Риски:

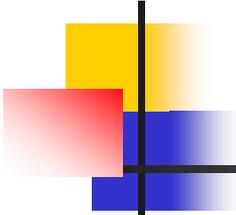
- Несовершенство технологии
- Затраты на обучение

- Для оценки эффективности нужно создавать прототип
- Нужно иметь оценку сроков/ресурсов с использованием уже изученных технологий

Безнадёжные проекты

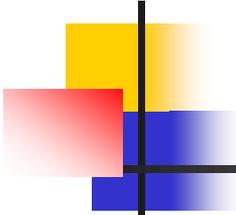
Выводы

- Избегайте безнадёжных проектов
- Ввязавшись, адекватно оценивайте свои силы



Содержание

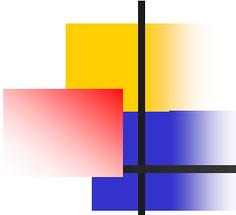
- Введение
- Брукс: Мифический человеко-месяц
- Йордон: Путь камикадзе
- **Пара слов про Agile**
- Заключение



Agile

Гибкие методика разработки (XP, Scrum)

- Минимизация бюрократии
- Нацеленность на результат
- Организация процессов в небольших командах



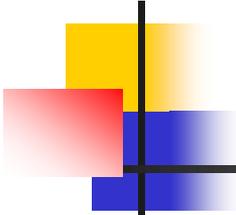
Agile

- Люди и взаимодействие важнее процессов и инструментов
- Работающий продукт важнее исчерпывающей документации
- Сотрудничество с заказчиком важнее согласования условий контракта
- Готовность к изменениям важнее следования первоначальному плану

Agile

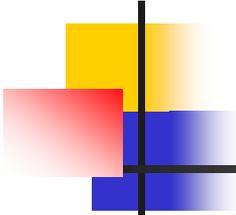
Некоторые особенности

- Взаимодействие команды и заказчика
- Release early, release often
- Самоорганизация команды
- Гибкость требований



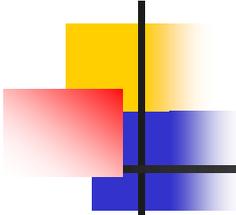
Содержание

- Введение
- Брукс: Мифический человеко-месяц
- Йордон: Путь камикадзе
- Пара слов про Agile
- **Заключение**



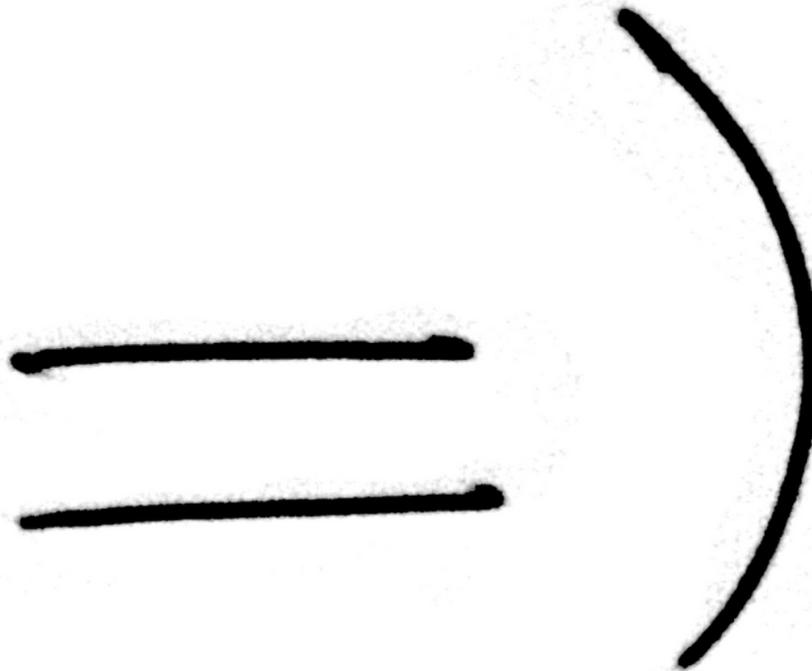
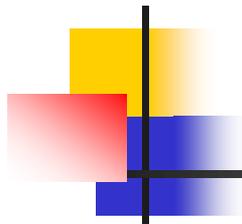
Заключение

- «Проектируй! @ Документируй!»
- Будьте ответственны
- Помните про конечный результат
- Помните про пользователя и его задачи



Литература

1. Фредерик Брукс, «Мифический человеко-месяц или как создаются программные системы», Символ-Плюс, СПб., 2010.
2. Эдвард Йордон, «Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте», Лори, 2003.
3. <http://agilemanifesto.org/>



Лаборатория компьютерной графики и мультимедиа



Видеогруппа — это:

- Выпускники в аспирантурах Англии, Франции, Швейцарии (в России в МГУ и ИПМ им. Келдыша)
- Выпускниками защищено 5 диссертаций
- Наиболее популярные в мире сравнения видеокодеков
- Более 3 миллионов скачанных фильтров обработки видео