

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ПРОГРАММА КУРСА

◎ 18 лекций; 9 практических занятий

◎ Основные баллы:

Блок	Максимум баллов	Баллов на аттестацию
Подготовка курсовой работы	15	7
Выполнение практических работ	30	15
Тесты на лекциях	5	-
Экзамен	50	25

◎ Дополнительные баллы:

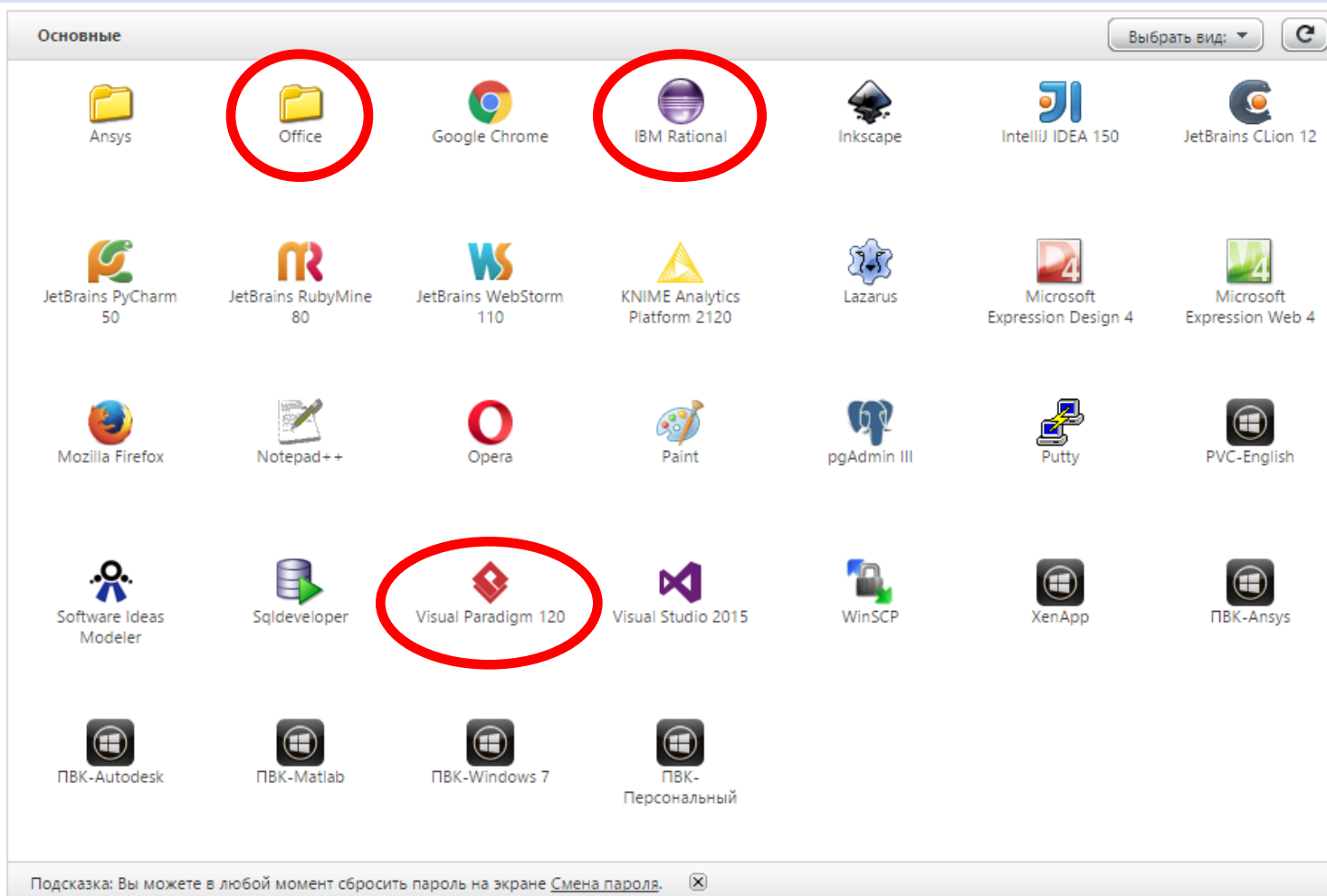
Блок	Максимум баллов
Посещаемость практических занятий	10
Подготовка и представление доклада	10
Выполнение дополнительных заданий по практическим работам	10

ИТОГОВАЯ АТТЕСТАЦИЯ

- 1) *Аттестован по всем блокам:* экзамен в виде теста
- 2) *Не аттестован хотя бы по одному блоку:* экзамен в виде теста + практическое задание сделать на листочке/в среде моделирования UML.
- 3) *Переэкзаменовка:* экзамен в виде теста + практическое задание сделать на листочке/в среде моделирования UML.

Оценка	Баллы
2 (Неуд.)	0-50
3 (Удовл.)	50.01-65
4 (Хор.)	65.01-75
5 (Отл.)	75.01-100

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ



ЛИТЕРАТУРА

- ◎ Арлоу Д., Нейштад А. **UML 2 и Унифицированный процесс**. Практический объектно-ориентированный анализ и проектирование, 2-е издание. 2007. 624 с.
- ◎ Сергей Орлов, Борис Цилькер. **Технологии разработки программного обеспечения** 4-е издание. 2012 г. 608 с.
- ◎ Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. 3-е издание. 2010 г. 718 с.
- ◎ Брукс Ф. **Мифический человеко-месяц**, или Как создаются программные системы. 2007. 304 с.

ДОКЛАДЫ

Максимум 10 баллов за доклад (минимум 45 минут).

1. Выбрать (или указать) тему доклада в [форуме на странице курса](#).
2. Выбрать занятие, на котором вы хотите сделать свой доклад, назначив встречу на соответствующий день в [календаре докладов](#).
3. Минимум за неделю до доклада загрузить на форум презентацию, получить отзыв и отредактировать презентацию в соответствии с замечаниями.

ВОЗМОЖНЫЕ ТЕМЫ ДОКЛАДОВ

Вы можете выбрать одну из тем представленных ниже, либо предложить собственную тему по программной инженерии:

- ◎ Гибкие методологии разработки ПО. Методология экстремального программирования. Scrum, Kanban.
- ◎ Системы управления версиями (VCS) ПО: история, виды VCS, современные VCS: SVN, GIT, Mercurial.
- ◎ Разработка через тестирование (TDD). Разработка на основе поведения (BDD). Системы поддержки TDD и BDD (Cucumber, SpecFlow).
- ◎ Концепция непрерывной интеграции ПО. Системы поддержки непрерывной интеграции. TeamCity, Jenkins (Hudson) - функциональные возможности, настройка и использование.
- ◎ Методы и подходы к командной разработке ПО. Системы и методологии поддержки командной разработки.

Пожалуйста, проверяйте, чтобы желаемая вами тема доклада не была выбрана до вас.

ОСНОВНЫЕ ТЕМЫ КУРСА

- ◎ Жизненный цикл ПО
- ◎ Составление требований к ПО и варианты использования
- ◎ Объектно-ориентированный анализ ПО
- ◎ Объектно-ориентированное проектирование ПО
- ◎ Кодирование и тестирование
- ◎ Архитектура программных систем
- ◎ Метрики и оценка качества ПО

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ИНЖЕНЕРИЯ

- ◎ **Инженерия** обеспечивает решение поставленных задач посредством существующих теорий и методов.
- ◎ Инженер начинает с **постановки задачи** и поиска инструментов для наилучшего **решения задачи** в рамках существующих **организационных, финансовых и временных ограничений**.
- ◎ **Программная инженерия** делает значительный упор на **методы и подходы** а не на инструменты.

Инжен'ерия

Engin'eering

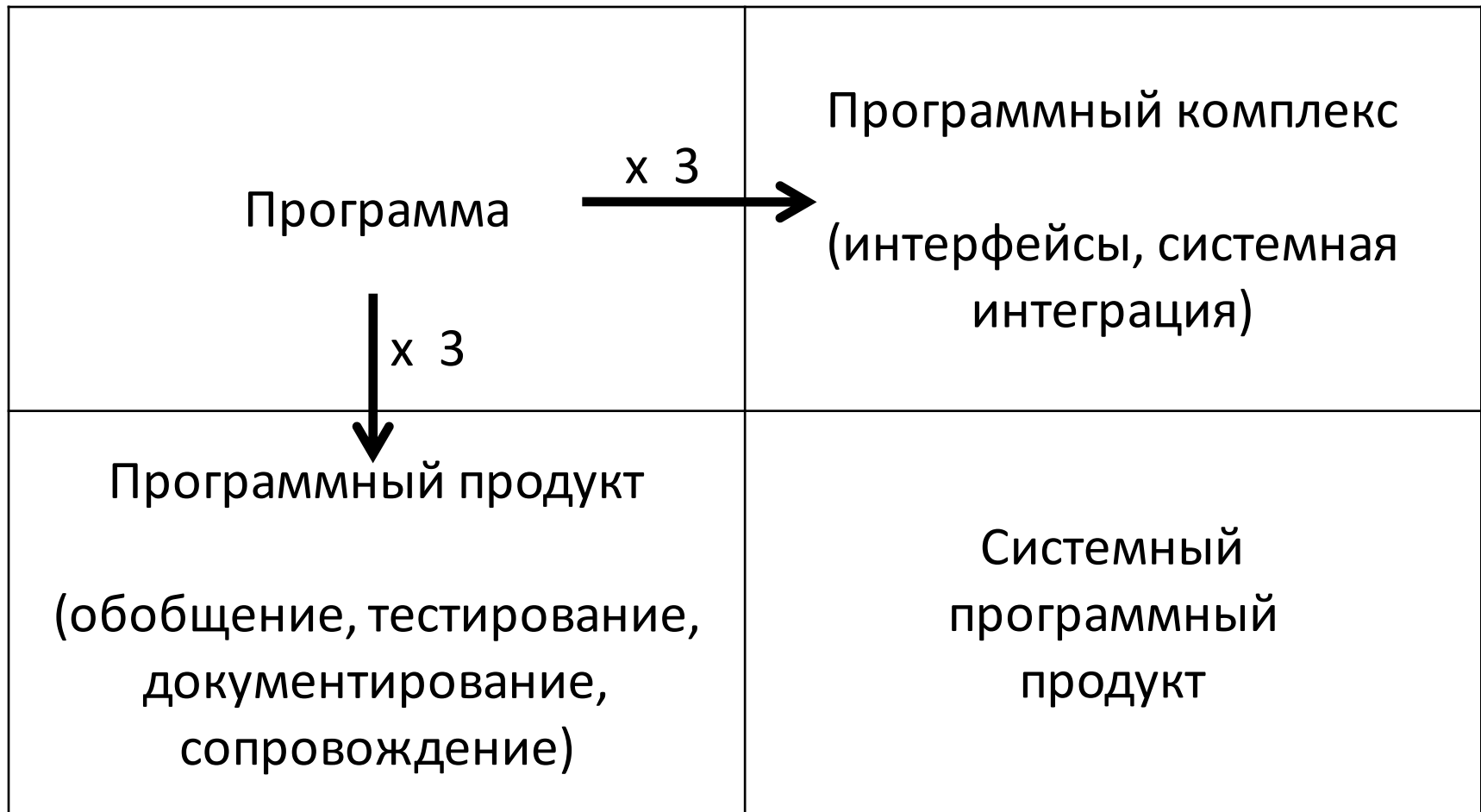
ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- ◎ Термин был предложен в 1968 г. на конференции посвященной «**Кризису ПО**», возникшего в результате появления интегральных схем и катастрофического усложнения ПО:
 - ◎ Реализация проектов задерживалась на годы
 - ◎ Стоимость проектов в десятки раз превышала прогнозируемую
 - ◎ Необходимы были методы разработки и контроля таких сложных программных систем

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

- © ***Программное обеспечение (программный продукт)*** – это компьютерная программа и соответствующая документация. Программные продукты могут быть разработаны как для конкретного заказчика, так и для всего рынка в целом.

ПРОГРАММА И ПРОГРАММНЫЙ ПРОДУКТ



ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- © ***Программная инженерия*** – это инженерная дисциплина, отражающая все грани разработки программного обеспечения.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ VS КОМПЬЮТЕРНЫЕ НАУКИ

Программная инженерия

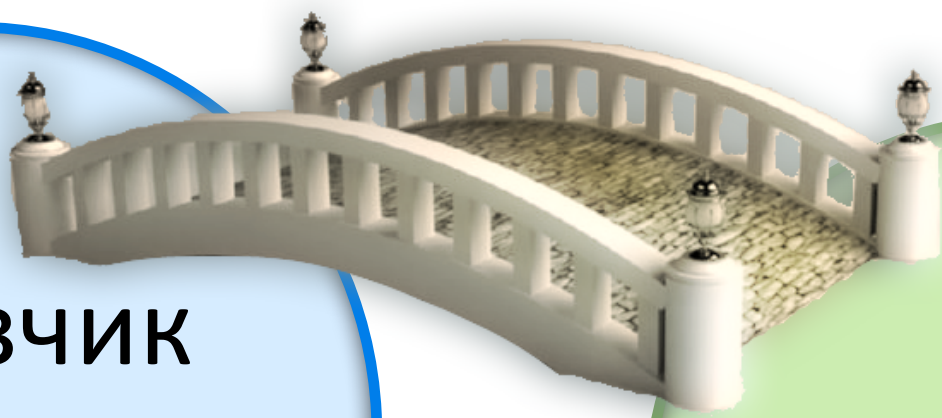
Практика и подходы к разработке полезных для конечного пользователя программных продуктов

Компьютерные науки

Теория и фундаментальные основы по созданию алгоритмов и компьютерных программ

В идеале, каждый программный инженер должен знать компьютерные науки (как каждый электрик должен знать физику)

ПРОГРАММНАЯ ИНЖЕНЕРИЯ



Заказчик

Задача,
требования,
ограничения

Программист

ХОРОШЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Можно выделить следующие важные признаки хорошего ПО:

- ① *Удобство сопровождения* - ПО должно быть написано таким образом, чтобы оно могло развиваться, чтобы соответствовать изменяющимся требованиям пользователей.
- ① *Функциональная надежность* – надежное ПО не должно приносить физического или экономического вреда в случае системной ошибки.
- ① *Эффективность ПО* – должно эффективно использовать ресурсы системы, включая память (оперативную и/или постоянную), процессорное время и т.п.
- ① *Применимость* – ПО должно обеспечивать решение задач пользователя для которых оно было разработано, без чрезмерных усилий

СОВРЕМЕННЫЕ ПРОБЛЕМЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

- ◎ Проблема гетерогенности
- ◎ Проблема своевременного представления результатов
- ◎ Проблема доверия

SWEBOK

ЧТО ТАКОЕ SWEBOOK

- ◎ SWEBOOK – это **Software Engineering Body of Knowledge** (Свод знаний по программной инженерии)
- ◎ SWEBOOK – это открытый проект, реализуемый при поддержке международного комитета IEEE начиная с 1998 года.
- ◎ В SWEBOOK определен набор знаний и рекомендуемые практики по инженерии программного обеспечения.
- ◎ Последняя версия (SWEBOOK V3) выпущена в 2013 году



*Guide to the Software
Engineering Body of Knowledge*

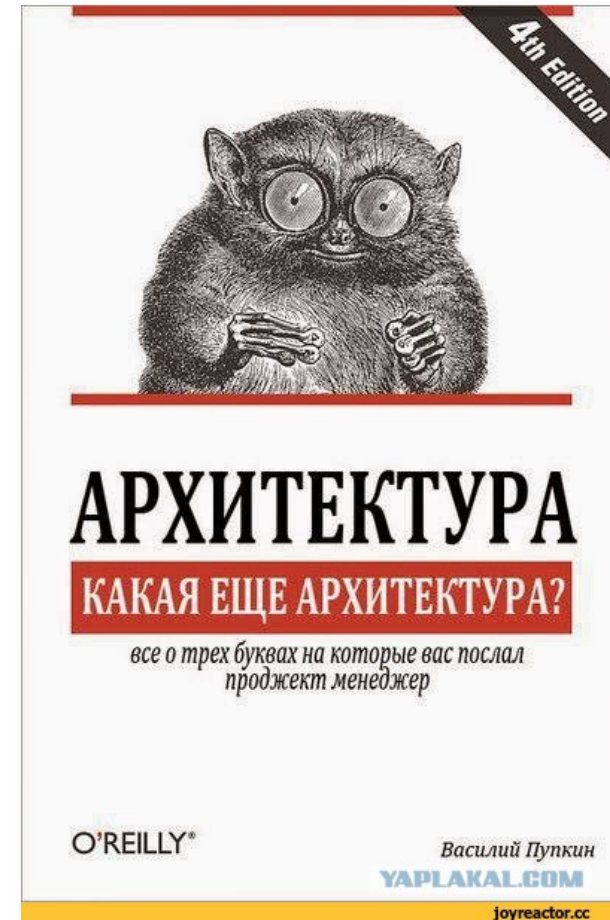
1. РАБОТА С ТРЕБОВАНИЯМИ К ПО

- Основы требований к ПО (что такое требование к ПО; Функциональные и нефункциональные требования)
- Процесс работы с требованиями
- Методы выявления требований
- Анализ требований к ПО
- Методы спецификации и валидации требований
- Системы для поддержки работы с требованиями



2. ПРОЕКТИРОВАНИЕ ПО

- Основы проектирования ПО (принципы и концепции проектирования ПО)
- Основные сложности при проектировании (управление параллелизмом, обработка событий, распределение компонентов систем, согласованность данных, обработка исключительных ситуаций, представление данных, безопасность)
- Структура и архитектура ПО (паттерны проектирования, архитектурные решения и принципы, основные фреймворки)
- Проектирование пользовательского интерфейса
- Оценка качества ПО
- Языки и нотации для описания архитектуры ПО
- Стратегии и методы проектирования ПО
- Системы поддержки проектирования ПО



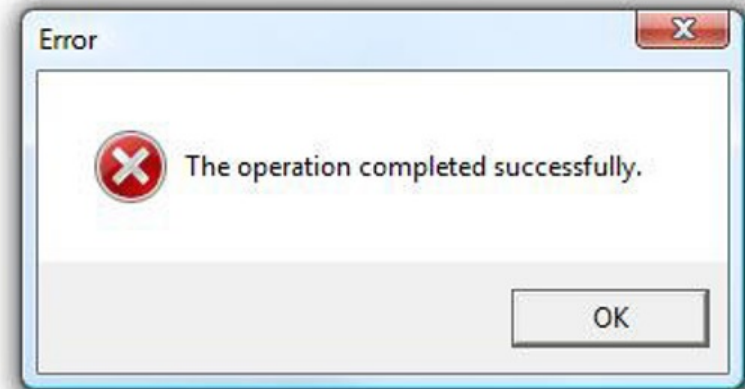
3. РАЗРАБОТКА (КОДИРОВАНИЕ) ПО

- Основы разработки ПО
- Практические подходы к кодированию с учетом повторного использования и качества кода
- Технологии разработки (создание API, объектно-ориентированная разработка, обработка ошибок и исключительных ситуаций, и др.)
- Среды разработки ПО.



4. ТЕСТИРОВАНИЕ ПО

- ⊙ Основы тестирования ПО
- ⊙ Уровни тестирования ПО
- ⊙ Методы тестирования ПО
- ⊙ Метрики тестирования ПО
- ⊙ Процесс тестирования
- ⊙ Системы поддержки тестирования ПО

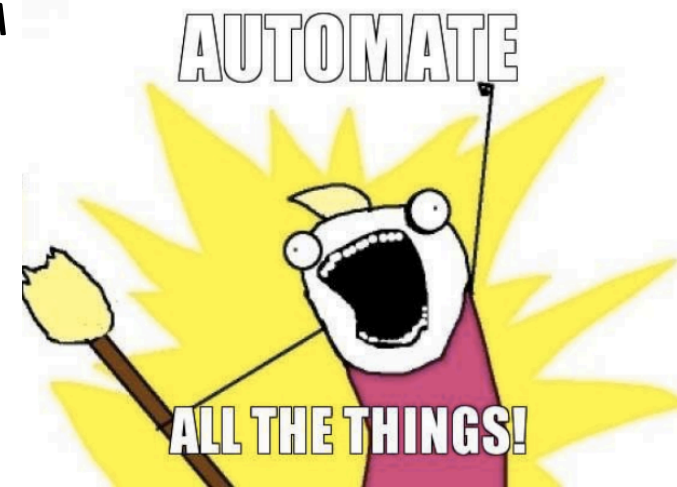


5. ПОДДЕРЖКА ПО

- ⊙ Основы поддержки ПО
- ⊙ Основные проблемы при поддержке ПО
- ⊙ Процесс поддержки
- ⊙ Техники обеспечения поддержки ПО

6. УПРАВЛЕНИЕ КОНФИГУРАЦИЯМИ:

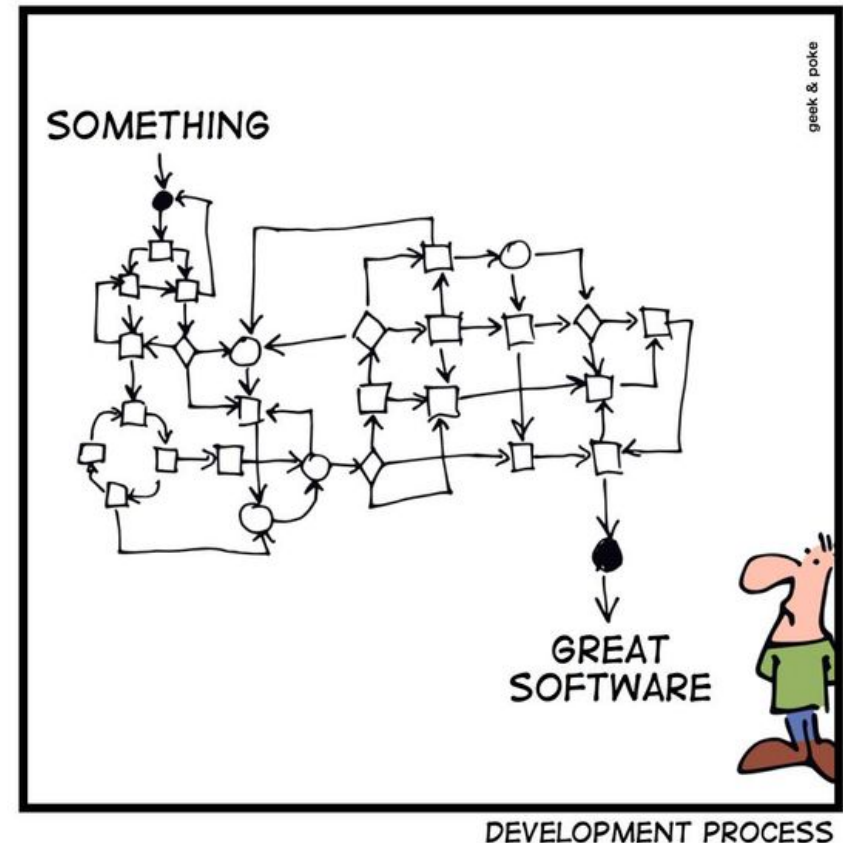
- Что такое конфигурация ПО и что такое управление конфигурациями.
- Процесс управление конфигурациями ПО
- Аудит конфигурации ПО
- Управление процессом сборки и развертывания программных систем
- Программные системы для поддержки процесса управления конфигурациями



8. ПРОЦЕСС РАЗРАБОТКИ ПО

- Определение понятия процесса разработки ПО
- Понятие жизненного цикла ПО
- Оценка качества и улучшение процесса разработки
- Измерение ПО
- Программные системы для поддержки процесса разработки ПО

SIMPLY EXPLAINED



9. МОДЕЛИ И МЕТОДЫ РАЗРАБОТКИ ПО

- ⊙ Понятие моделирования в разработке ПО
- ⊙ Типы моделей (информационные, структурные, поведенческие)
- ⊙ Методы программной инженерии (эвристические, формальные, прототипирование, гибкие методы)

10. КАЧЕСТВО ПО

- ⊙ Понятие качества ПО (культура и этика при разработке ПО, безопасность ПО)
- ⊙ Управление качеством ПО
- ⊙ Практические методы управления качеством (требования к качеству ПО, определение дефектов, измерение качества ПО)
- ⊙ Программные системы для управления качеством

11. ПРОФЕССИОНАЛЬНЫЕ ПРАКТИКИ

- ⊙ Понятие профессионализма:
 - профессиональная аккредитация и лицензирование, кодекс этики;
 - профессиональные сообщества, профессиональные стандарты;
 - экономические эффекты ПО;
 - понятие контракта на разработку ПО, контракт на работу;
 - юридические особенности разработки ПО
- ⊙ Управление группами
 - особенности групповой работы
 - управление сложностью
 - взаимодействие с заказчиком
 - взаимодействие с людьми из других культур
- ⊙ Коммуникация
 - Чтение
 - Письмо
 - Взаимодействие внутри команды
 - Навыки презентации

12. ЭКОНОМИКА РАЗРАБОТКИ ПО

- ⊙ Основы экономики разработки ПО (финансы, бухгалтерия, денежный поток, налоги)
- ⊙ Экономика программного продукта (проект, продукт, жизненный цикл проекта и продукта, инвестиции, горизонты планирования, определение цены)
- ⊙ Риски и неопределенность
- ⊙ Модели экономического анализа

14. МАТЕМАТИЧЕСКИЕ ОСНОВЫ

- ⊙ Множества, отношения, функции
- ⊙ Математическая логика
- ⊙ Методы доказательств
- ⊙ Графы и деревья
- ⊙ Дискретная вероятность
- ⊙ Машины конечных состояний
- ⊙ Грамматики
- ⊙ Численные методы, точность
- ⊙ Теория чисел
- ⊙ Алгебраические структуры

15. ОСНОВЫ ИНЖЕНЕРИИ

- ⊙ Эмпирические методы и техники эксперимента
- ⊙ Статистический анализ
- ⊙ Основы измерений
- ⊙ Моделирование, симуляция, прототипирование
- ⊙ Стандарты