

# Технология разработки программного обеспечения

## Лекция 1: О программной инженерии

Юрий Литвинов  
y.litvinov@spbu.ru

17.02.2026

# Организационное

- ▶ Лекционный курс, с одной «контрольной»
- ▶ На самом деле, практика на паре с небольшим баллом
- ▶ В конце зачёт
- ▶ Примерно 70 вопросов без подготовки
- ▶ Балльная система ECTS
- ▶ 80 баллов за зачёт, 20 баллов за практическую работу
- ▶ Материалы будут выкладываться в HwProj, туда же сдавать практическую работу
- ▶ <https://hwproj.ru/courses/50078>
- ▶ Коммуникация в чате курса, можно писать в личку



## Что будет в курсе

- ▶ Что в разработке программного обеспечения делается помимо программирования, кем ещё можно работать после матмеха
- ▶ Работа с требованиями
- ▶ Жизненный цикл программного обеспечения
- ▶ Методологии разработки, Scrum
- ▶ Проектирование пользовательских интерфейсов
- ▶ Управление проектами: что делают менеджеры
- ▶ Работа в команде
- ▶ Качество программного обеспечения, работа с дефектами
- ▶ Сопровождение и поддержка, развитие проектов после релиза
- ▶ Рефакторинг (если успеем)
- ▶ Непрерывная интеграция/непрерывное развёртывание
- ▶ Экономические аспекты разработки

# Программа и программный продукт



# Особенности промышленной разработки ПО

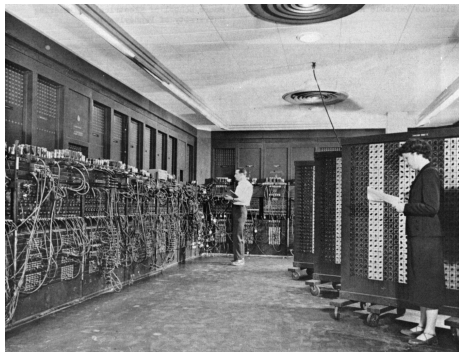
- ▶ Работа в команде
- ▶ Работа для заказчика и за деньги
- ▶ Требования, сроки и качество
- ▶ Поэтому нужны дополнительные действия:
  - ▶ Анализ и проектирование
  - ▶ Выбор технологий и планирование
  - ▶ Организация процесса разработки
  - ▶ Учёт необходимости сопровождения и интеграции, документирование, стайлгайд
  - ▶ Формирование команды, подбор персонала, оборудования и помещений
  - ▶ ...

# Программная инженерия как область знания

- ▶ Организация и улучшение процесса разработки ПО, управление коллективом разработчиков, разработка и внедрение средств поддержки жизненного цикла разработки ПО
- ▶ Осмысление, обобщение и оформление опыта
- ▶ Методы и практики тестирования, проектирования, работы над требованиями и т.п.
- ▶ Стандарты и методологии

# Источники: ENIAC

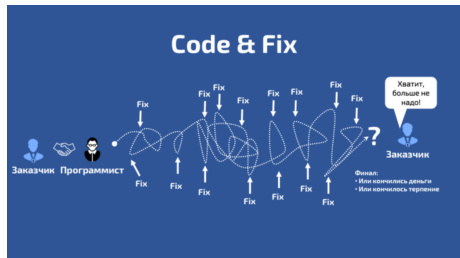
- ▶ ENIAC — Electronic Numerical Integrator and Computer
- ▶ Программирование происходило физически тумблерами и штекерами
- ▶ Порог вхождения очень высок
- ▶ Программ отдельно от компьютеров не существовало
- ▶ Управлять процессом не требовалось вообще



# Взрывной рост разработки

## Появление языков высокого уровня

- ▶ 1957 — Fortran (Formula Translator)
- ▶ Начало массовой разработки на заказ
- ▶ Процесс управления разработкой «Code & Fix» (также известный как Cowboy Coding)
- ▶ ПО всё ещё привязано к «железу»
- ▶ Стандартов разработки нет





# Первая попытка навести порядок

## Официальное рождение программной инженерии

- ▶ Кризис программного обеспечения
  - ▶ Стоимость проектов превышает бюджет
  - ▶ Превышаются сроки выполнения проектов
  - ▶ ПО слишком неэффективно
  - ▶ ПО имеет слишком низкое качество
  - ▶ ПО не отвечает необходимым требованиям
  - ▶ Неуправляемые проекты, трудности с поддержкой кода
  - ▶ ...
- ▶ Конференция NATO Software Engineering (1968 год)
  - ▶ Оборонка страдала больше всех



# Текущее положение дел

## Standish Group Chaos Report

|            | 2011 | 2012 | 2013 | 2014 | 2015 | 2020 |
|------------|------|------|------|------|------|------|
| Successful | 29%  | 27%  | 31%  | 28%  | 29%  | 31%  |
| Challenged | 49%  | 56%  | 50%  | 55%  | 52%  | 50%  |
| Failed     | 22%  | 17%  | 19%  | 17%  | 19%  | 19%  |

## Отличия от других областей производства

- ▶ Очень высокая сложность систем
  - ▶ <http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>
- ▶ Меньше накоплено опыта
  - ▶ Более непредсказуем результат
  - ▶ Хуже поддается планированию
  - ▶ Больше творчество, чем ремесло
- ▶ Подверженность постоянным изменениям
- ▶ Но и стоимость изменений значительно ниже

# Разработка ПО крайне социализирована

- ▶ Разработка ведётся людьми
  - ▶ Общение внутри команды
- ▶ Разработка ведётся для людей
  - ▶ Общение за пределами команды
- ▶ Успех определяется социальными факторами
  - ▶ Технологии вторичны?

# Команда



## Востребованные компетенции

- ▶ Умение работать в команде
- ▶ Владение современными стратегиями и технологиями организации коллективной разработки программного обеспечения, включая системы управления версиями, процессы непрерывной интеграции, стандарты оформления кода и методы инспекции кода
- ▶ Понимание основных направлений развития методов коллективной разработки, их отличий и целесообразности применения в зависимости от типа решаемых задач и требований организации
- ▶ Владение более чем одним языком программирования/стеком технологий

# Профстандарты

- ▶ Собрание трудовых функций, знаний и умений для данной профессии, относительно которых есть консенсус в индустрии
- ▶ Разрабатываются комитетами из крупных компаний, утверждаются Минтруда
- ▶ Нужны прежде всего для стандартизации требований и подготовки
  - ▶ В перспективе — стандартизованной сертификации
- ▶ Разбиты на уровни квалификации
  - ▶ Всего 9 уровней, от неквалифицированного труда до «управления крупными техносистемами, генерации фундаментальных знаний»
  - ▶ Диплом бакалавра — с шестого, магистерский — с седьмого
  - ▶ Девятый уровень требует окончания аспирантуры

# Профстандарт «Программист», 3-й уровень квалификации

Минимальный. Низкоквалифицированный труд

- ▶ Формализация и алгоритмизация поставленных задач для разработки программного кода
- ▶ Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных
- ▶ Оформление программного кода в соответствии с установленными требованиями
- ▶ Работа с системой управления версиями программного кода
- ▶ Проверка и отладка программного кода



# Профстандарт «Программист», 4-й уровень квалификации

## «Миддл»

- ▶ Разработка процедур проверки работоспособности и измерения характеристик компьютерного программного обеспечения
- ▶ Разработка тестовых наборов данных для проверки работоспособности компьютерного программного обеспечения
- ▶ Проверка работоспособности компьютерного программного обеспечения
- ▶ Да-да, тестировщик должен иметь большую квалификацию, чем программист
- ▶ Рефакторинг, оптимизация и инспекция программного кода
- ▶ Исправление дефектов программного кода, зафиксированных в базе данных дефектов
- ▶ Осуществление сборки однородных программных модулей в программный проект

# Профстандарт «Программист», 5-й уровень квалификации

- ▶ Разработка процедур интеграции программных модулей
- ▶ Осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта

# Профстандарт «Программист», 6-й уровень квалификации

## «Сеньор»

- ▶ Анализ возможностей реализации требований к компьютерному программному обеспечению
- ▶ Разработка технических спецификаций на программные компоненты и их взаимодействие
- ▶ Проектирование компьютерного программного обеспечения

## Какие ещё профстандарты учитываются

- ▶ 06.003 «Архитектор программного обеспечения»
- ▶ 06.004 «Специалист по тестированию в области информационных технологий»
- ▶ 06.011 «Администратор баз данных»
- ▶ 06.015 «Специалист по информационным системам»
- ▶ 06.019 «Технический писатель (специалист по технической документации в области информационных технологий)»
- ▶ 06.022 «Системный аналитик»
- ▶ 06.026 «Системный администратор информационно-коммуникационных систем»
- ▶ 06.028 «Системный программист»
- ▶ 40.011 «Специалист по научно-исследовательским и опытно-конструкторским разработкам»
- ▶ 40.057 «Специалист по автоматизированным системам управления производством»

# Software Engineering Book of Knowledge

1. Software Requirements
2. Software Design
3. Software Construction
4. Software Testing
5. Software Maintenance
6. Software Configuration Management
7. Software Engineering Management
8. Software Engineering Process
9. Software Engineering Models and Methods
10. Software Quality
11. Software Engineering Professional Practice
12. Software Engineering Economics
13. Computing Foundations
14. Mathematical Foundations
15. Engineering Foundations