

Программная инженерия

Software engineering

Основные темы курса

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

Классификация ПО и проектов

История

Стандарты и определения

**ВОЗНИКНОВЕНИЕ
ПРОГРАММНОЙ ИНЖЕНЕРИИ**

«Большое» и «малое» ПО

Практические выводы

**КЛАССИФИКАЦИЯ ПО И
ПРОЕКТОВ**

Классификация ПО

- «Малое» ПО
- «Большое» ПО

Примечание: Деление условное!

«Малое» ПО – 1/2

- Решает одну несложную, четко поставленную задачу
- Размер исходного кода не превышает нескольких сотен строк
- Скорость работы программного обеспечения и необходимые ему ресурсы не играют большой роли
- Ущерб от неправильной работы не имеет большого значения

«Малое» ПО – 2/2

- Модернизация программного обеспечения, дополнение его возможностей требуется редко
- Как правило, разрабатывается одним программистом или небольшой группой (5 или менее человек)
- Подробная документация не требуется, ее может заменить исходный код, который доступен

«Большое» ПО – 1/2

- «Большое» ПО имеет 2-3+ характеристик ниже
- Характеристики:
 - Решает совокупность взаимосвязанных задач
 - Использование приносит значимую выгоду
 - Удобство его использования играет важную роль
 - Обязательно наличие полной и понятной документации
 - Производительность
 - Низкая скорость работы приводит к потерям

«Большое» ПО – 2/2

- Характеристики (продолжение):
 - Отказоустойчивость
 - Сбои, неправильная работа, наносит ощутимый ущерб
 - Внешние интерфейсы
 - Программы в составе ПО во время работы взаимодействуют с другими программами и программно-аппаратными комплексами
 - Кроссплатформенность
 - Работает на разных платформах и/или связывает их
 - Требуется развитие, исправление ошибок, добавление новых возможностей
 - Группа разработчиков больше 5 человек

Размер проектов разработки ПО

Размер проекта	Команда разработчиков	Срок
Небольшой	Меньше 10 человек	3-6 месяцев
Средний	20-30 человек	1-2 года
Крупномасштабный	100-300 человек	3-5 лет
Гигантский	1000-2000 человек (включая консультантов)	7-10 лет

Практические выводы – 1/3

- Чем крупнее проект, тем выше необходимость использования в нем приемов Программной инженерии
- Чем крупнее проект, тем большее число приемов и методик ПИ полезно применять
- Для мелких проектов использование всех методик ПИ приведет к размыванию целей и перерасходу сил и средств

Практические выводы – 2/3

- Разработчики организуются в команды, взаимодействие идет не «каждый с каждым». Имеются связи:
 - Очевидно:
 - Внутри команд
 - Иерархические между соседними уровнями
 - Не так очевидно:
 - Горизонтальные между членами разных команд
 - Вертикальные, «пронизывающие» несколько уровней иерархии

Практические выводы – 3/3

- «Гигантских» проектов мало:
 - Нет политических и экономических причин планировать так далеко:
 - Только государства и/или крупные корпорации справятся с финансированием
 - Монополярный мир
 - Человечество не имеет сейчас целей кроме краткосрочных
 - Цикл изменений IT технологий короче длительности проекта (в 2-3 раза!)
- Есть «микроскопические» проекты – разработка выполняется одним человеком:
 - Пример: Приложения для смартфонов
 - Существуют в глобальной инфраструктуре – отдельно от нее не нужны и невозможны

Вехи

Кризис ПО

Современное ПО

ИСТОРИЯ ВОЗНИКНОВЕНИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ

История

- 1960-е годы – начало «Кризиса ПО»:
 - Время: реализация проектов задерживается (на годы!)
 - Деньги: Стоимость проектов превышает прогнозируемую (в десятки раз!)
- 1968 – термин «software engineering» появляется как название первой мировой конференции посвященной проблеме
- Сейчас:
 - Существует несколько хорошо структурированных подходов к разработке ПО
 - Есть отечественные и международные стандарты, описывающие разработку ПО
 - «Кризис ПО» так и не закончился

Причины «Кризиса ПО» – 1/2

- Непреодолимые:
 - Недостаточная поддержка со стороны высшего руководства
 - Недостаточно высокая квалификация разработчиков, отсутствие необходимого опыта
 - Отсутствие необходимых ресурсов
- Преодоленные:
 - Новизна и несовершенство используемой технологии
- Преодолимые известно как:
 - Нечеткая и неполная формулировка требований
 - Недостаточное вовлечение пользователей в работу над проектом

Причины «Кризиса ПО» – 2/2

- Преодолимые теоретически:
 - Неудовлетворительное планирование и отсутствие грамотного управления проектом
 - Сжатые сроки
 - Недостаток ресурсов (разработчиков, финансов, средств технического обеспечения)
 - Частое изменение требований и спецификаций
 - Завышение требований

Особенности современного ПО – 1/2

- Сложность
- Отсутствие полных аналогов → высокая доля вновь разрабатываемого ПО
- Наследование:
 - Какое-то ПО уже существует
 - Требуется его сохранить и интегрировать или доработать
 - Сохранение функциональности, пользовательских интерфейсов, данных, приемов использования

Особенности современного ПО – 2/2

- Распределенность:
 - Среда функционирования
 - Среда разработки (sic!)
 - Команды разработчиков
- Неоднородность среды функционирования

Разработка современного ПО – 1/2

- Групповая разработка:
 - Большое количество участников проектирования
 - Разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и опыту
- Неформальность требований к ПО vs формальность самого ПО
- Творческий характер разработки

Разработка современного ПО – 2/2

- Дуализм ПО:
 - ПО это статический объект – совокупность текстов [и сгенерированных исполнимых модулей]
 - ПО это динамический объект – при работе (эксплуатации) ПО порождаются процессы обработки данных
- При своем использовании (эксплуатации) ПО не расходуетя и не изнашивается
- «Нереальность» ПО: подталкивает к постоянному переделыванию

Определения

Стандарты

SWEVOK

СТАНДАРТЫ И ОПРЕДЕЛЕНИЯ

Определения – 1/2

- Сложно
 - [Wikipedia]: Программная инженерия — это интегрирование принципов математики, информатики и компьютерных наук с инженерными подходами, разработанными для производства осязаемых материальных артефактов.
 - [ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB)]: the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software

Определения – 2/2

- Проще:
 - Инженерная дисциплина, отражающая все грани разработки программного обеспечения
 - An engineering discipline that is concerned with all aspects of software production
- Почти неформально:
 - Обозначение всех видов деятельности, входящих в программирование (кодирование) и системный анализ

Стандарты

- Определения
 - ISO/IEC/IEEE 24765 - Systems and software engineering – Vocabulary (Системы и программная инженерия. Словарь)
- Жизненный цикл ПО
 - ISO/IEC 12207:2008 «System and software engineering — Software life cycle processes»
 - ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»

Определения

Knowledge Areas

Связанные дисциплины

SWEBOK

SWEBOK

- SWEBOK (Guide to the Software Engineering Body of Knowledge) – документ, подготавливаемый комитетом Software Engineering Coordinating Committee, в который вовлечено сообщество IEEE Computer Society
- 2004 – Выпущена первая версия
- 2013 – Выпущена текущая версия v3.0

SWEBOK: Назначение – 1/3

- Формулировка согласованного представления о Программной инженерии
 - Разрабатываемый документ рецензировали 150 человек из 33 стран
- Определение области применения и объяснения места ПИ среди других дисциплин, таких как информатика, управление проектами, инженерных дисциплин и математики
 - Материал разбит на 15 Knowledge Areas (см. ниже)
 - Часть пересечений с общими дисциплинами отражена как отдельные КА (см. ниже)

SWEBOK: Назначение – 2/3

- Описание общего содержания ПИ
 - Материал, описывающий КА, организован иерархически
- Описания содержимого ПИ по темам
 - Описание каждого КА завершается список тем
 - Приложение С содержит общий тематический указатель

SWEBOK: Назначение – 3/3

- Содержит материалы для составления учебных планов, сертификации и лицензирования
 - Материалы документа «generally accepted»:
 - Не являются «продвинутыми» (advanced) или исследовательскими
 - Не являются специализированными (относящимися к узкой области применения)
 - Такая практика широко распространена в США, но практически применима всюду

SWEBOK: Knowledge Areas – 1/3

- КА (Knowledge Area) – область знаний, определяющая содержание ПИ
- Всего в SWEBOK определено 15 КА:
 - I. Этапы жизненного цикла ПО:
 1. Software Requirements — требования к ПО
 2. Software Design — проектирование ПО
 3. Software Construction — конструирование ПО
 4. Software Testing — тестирование ПО
 5. Software Maintenance — сопровождение ПО

SWEBOK: Knowledge Areas – 2/3

II. Сопутствующие процессы:

6. Software Configuration Management — управление конфигурациями
7. Software Engineering Management — управление IT проектом
8. Software Engineering Process — процесс программной инженерии
9. Software Engineering Models and Methods — модели и методы разработки

SWEBOK: Knowledge Areas – 3/3

III. Оценки и качество:

10. Software Engineering Professional Practice — описание критериев профессионализма и компетентности
11. Software Quality — качество ПО
12. Software Engineering Economics — экономические аспекты разработки ПО

IV. Основы применимых технологий:

13. Computing Foundations — основы вычислительных технологий, применимых в разработке ПО
14. Mathematical Foundations — базовые математические концепции и понятия, применимые в разработке ПО
15. Engineering Foundations — основы инженерной деятельности

Дисциплины, связанные с ПИ – 1/3

- Computer Engineering (компьютерная инженерия , инженерия компьютерных систем)
 - Объединяет информатику и электронику (в том числе компьютерные сетевые технологии)
 - Нацелена на разработку компьютеров и программного обеспечения. При этом особое внимание уделяется взаимосвязи компьютеров и ПО, в отличие от «чистой» разработки ПО или компьютеров
 - Инженеры, работающие в этой сфере, работают в области разработки специальных микроконтроллеров, ПК, суперкомпьютеров, компьютерных сетей.
 - Эта область инженерии сосредоточена не только на самой работе компьютерных систем, но и на их интеграции для выполнения задач более высокого уровня

Дисциплины, связанные с ПИ – 2/3

- Computer Science

- Определение: Научный и практический подход к вычислениям и их применению. Изучается вычислимость, структуры, выражения, а также автоматизация процессов лежащих в основе сбора, представления, обработки, хранения, обмена и доступа к информации.
- Определение 2: изучение автоматизации масштабирующихся алгоритмизируемых процессов
- Инженеры, работающие в этой сфере, занимаются теорией вычислений и дизайном вычислительных систем

- = Информатика (?)

- Наука о методах и процессах сбора, хранения, обработки, передачи, анализа и оценки информации с применением компьютерных технологий, обеспечивающих возможность её использования для принятия решений

Синим выделено то, части чего приведены в SWEBOOK КА

Дисциплины, связанные с ПИ – 3/3

- General Management – теория управления
 - Управление – это функция согласования усилий людей для достижения целей и выполнения задач путем рационального и эффективного использования имеющихся ресурсов
- Mathematics – математика
- Project Management – управление проектами
- Quality Management – управление качеством
- Systems Engineering – общая (системная) инженерия
 - Междисциплинарная область знаний, описывающих как спроектировать и управлять сложными системами на всей продолжительности их жизненного цикла
 - Когда мы имеем дело со сложными или большими системами такие задачи как управление требованиями, надежность, логистика, координирование межгруппового взаимодействия, тестирование и валидация, поддержка и другие значительно усложняются по сравнению с «обычным» уровнем

Синим выделено то, части чего приведены в SWEBOOK КА