Лекция #1

Методы анализа и обеспечения качества ПО

Михаил Моисеев

Качество ПО. Вводная лекция

Цели и задачи курса

- Актуальный вопрос современной индустрии ПО – **обеспечение качества**
- Тенденции в образовании: от теории и технологий программирования к программной инженерии
- Цель курса: рассмотреть теоретические и практические вопросы обеспечения качества ПО на различных этапах жизненного цикла
- Представляемые методы «в основном» являются формальными – могут быть представлены с помощью некоторого формализма

Организация курса

- Преподаватели
  - Моисеев Михаил Юрьевич, *Intel Labs*
  - Ахин Марат Халимович, *Digitek Labs*
  - Цесько Вадим Александрович,*Yandex*
  - Беляев Михаил Анатольевич, *Digitek Labs*
- Лекции и лабораторные работы
- Примеры программ и языки программирования
- Информационные ресурсы
  - Сайт кафедры - http://kspt.ftk.spbstu.ru/course/QA
- Отчетность и экзамен

Рекомендуемая литература

- Paul Ammann, Jeff Offutt. Introduction to Software Testing. -- Cambridge University Press, 2008
- Cem Kaner, Jack Falk, Hung Q. Nguyen. Testing Computer Software. -- Wiley, 1999
- Ю.Г. Карпов. Model Checking. Верификация параллельных и распределенных программных систем -- БХВ-Петербург, 2010
- Doron A. Peled. Software Reliability Methods. -- Springer, 2001
- Nielson F., Nielson H.R., Hankin C. Principles of Program Analysis. Springer, 2005
- M.R. Lyu Handbook of Software Reliability Engeneering, 1995
- Г.Майерс. Надежность программного обеспечения, 1980
- Shwarzbach M. Lecture notes on static analysis http://www.brics.dk/\~mis/static.pdf
- В. Кулямин. Методы верификации программного обеспечения – http://www.sci-innov.ru/icatalog_new/entry_62322.htm

Качество ПО

- Качество ПО – комплексная характеристика свойств ПО с точек зрения различных участников
- Стандартная модель качества ПО представлена в ISO 9126

Качество ПО

- Стандарт ISO 9126 учитывает точки зрения
  - Разработчиков – *внутреннее качество ПО*
  - Руководства и аттестации ПО – *внешнее качество ПО*
  - Конечных пользователей – *качество ПО при использовании*
- Качество ПО включает
  - 6 факторов
  - 27 атрибутов – для качественной оценки факторов
  - метрики или показатели – для количественной оценки атрибутов
- ГОСТ Р ИСО/МЭК 9126

Функциональность

- **Функциональность** – способность ПО в определенных условиях решать задачи, нужные пользователям
  - Функциональная пригодность – способность решать нужный набор задач
  - Точность – способность выдавать нужные результаты
  - Способность к взаимодействию, совместимость – способность взаимодействовать с нужным набором других систем
  - Соответствие стандартам и правилам – соответствие ПО имеющимся стандартам, нормативным и законодательным актам, другим регулирующим нормам
  - Защищенность – способность предотвращать неавторизованный и не разрешенный доступ к данным, коммуникациям и др

Надежность

- **Надежность** – способность ПО выполнять свои функции в заданных условиях
  - Зрелость – величина, обратная частоте критических отказов, вызванных ошибками в ПО
  - Устойчивость к отказам – способность поддерживать заданный уровень работоспособности при внутренних и внешних отказах
  - Способность к восстановлению – способность восстанавливать определенный уровень работоспособности и целостность данных после отказа
  - Соответствие стандартам надежности

Удобство сопровождения

- **Удобство сопровождения** – удобство проведения всех видов деятельности, связанных с сопровождение программ
  - Удобство проведения анализа – удобство проведения анализа ошибок, дефектов и недостатков, а также удобство анализа необходимости изменений и их возможных последствий
  - Удобство проверки – показатель, обратный трудозатратам на проведение тестирования и других видов проверки того, что внесенные изменения привели к нужным результатам
  - Удобство внесения изменений – показатель, обратный трудозатратам на выполнение необходимых изменений
  - Стабильность – показатель, обратный риску возникновения неожиданных эффектов при внесении необходимых изменений
  - Соответствие стандартам удобства сопровождения

Эффективность

- **Эффективность** (производительность) – свойство ПО при заданных условиях обеспечивать необходимую работоспособность по отношению к выделяемым ресурсам
  - Временная эффективность – способность ПО решать определенные задачи за отведенное время
  - Эффективность использования ресурсов – способность решать нужные задачи с использованием заданных объемов ресурсов определенных видов (ресурсоемкость)
  - Соответствие стандартам производительности

Удобство использования

- **Удобство использования** – способность ПО быть удобным в обучении и использовании
  - Понятность – показатель, обратный к усилиям, которые затрачиваются пользователями на восприятие основных понятий ПО и осознание способов их использования для решения своих задач
  - Удобство обучения – показатель, обратный к усилиям, затрачиваемым пользователями на обучение работе с ПО
  - Удобство работы – показатель, обратный трудоемкости решения пользователями задач с помощью ПО
  - Привлекательность – способность ПО быть привлекательным для пользователей
  - Соответствие стандартам удобства использования

Переносимость

- **Переносимость** (мобильность) – способность ПО сохранять работоспособность при переносе из одного окружения в другое (аппаратное, программное окружение)
  - Адаптируемость – способность ПО приспосабливаться к различным окружениям без специальных действий
  - Удобство установки – способность ПО быть установленным или развернутым в определенном окружении
  - Способность к сосуществованию – способность ПО сосуществовать в общем окружении с другими программами, разделяя с ними общие ресурсы
  - Удобство замены другого ПО данным – возможность применения данного ПО вместо других программных систем для решения тех же задач в определенном окружении
  - Соответствие стандартам переносимости

Вопросы рассматриваемые в курсе

- **Функциональность**
  - Функциональная пригодность – проверка программы на соответствие функциональной спецификации
  - Защищенность – проверка программы на наличие уязвимостей безопасности
- **Надежность**
  - Зрелость – обнаружение ошибок, имеющихся в программе, оценка числа оставшихся ошибок и их влияния на работу программы
- **Эффективность**
  - Оценки эффективности
- **Удобство сопровождения**
  - Удобство проведения анализа
  - Удобство проверки

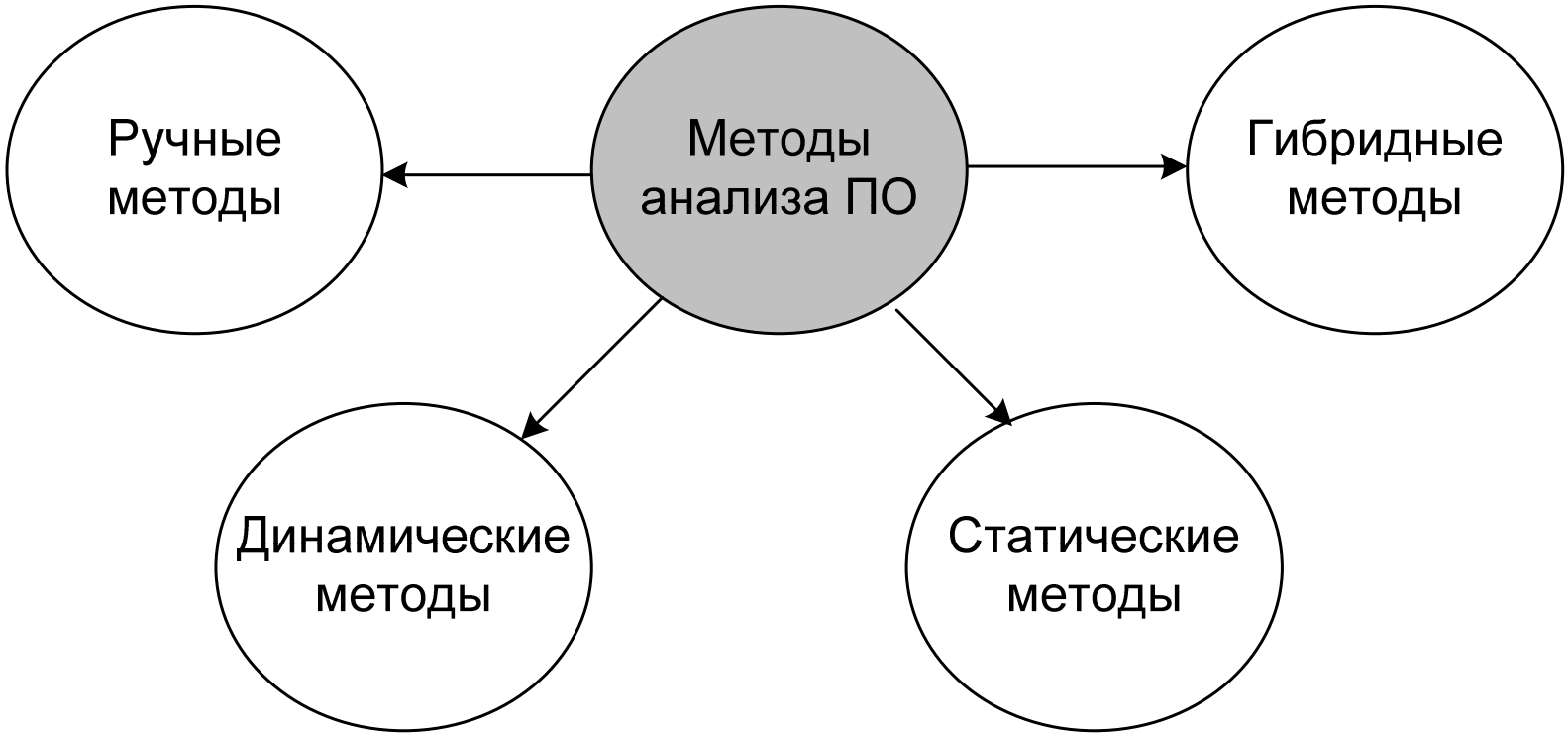
Причины недостаточного качества

- Функциональность
  - Функциональные ошибки – несоответствия требованиям пользователей, функциональной спецификации и т.п.
- Надежность
  - Нефункциональные ошибки – нарушение правил языка программирования, использования библиотечных функций и сторонних компонентов и т.п.
- Эффективность
  - Использование неэффективных алгоритмов и их реализаций
  - Ошибки анализа необходимого количества ресурсов, обычно проявляются только в определенных ситуациях

Задачи обеспечения качества

- Обеспечение качества
  - **Применение методов повышения качества**
  - **Измерение (оценка) качества программы**
- Повышение качества
  - Обнаружение ошибок и неудовлетворительных мест в программе
  - Исправление ошибок и другие изменения программы
- Необходимость оценки качества
  - Контроль текущего прогресса
  - Оценка эффективности затрат на повышение качества
  - Выбор наиболее эффективных методов повышения качества
- Основа для измерения и повышения качества – **анализ ПО**

Методы анализа ПО



Ручные методы

- Персональные проверки
- Аудит кода
- Парное программирование
- Ручная верификация

Динамические методы

- **Динамические методы** используют результаты выполнения программы
- Тестирование
  - Модульное
  - Системное
  - Нагрузочное
- Мониторинг
- Профилирование
- Анализ трасс выполнения

Статические методы

- **Статические методы** используют различные артефакты получаемые в процессе проектирования ПО
  - требования
  - спецификации
  - исходные код программы
- Методы формальной верификации
  - Дедуктивная верификация
  - Верификация на основе проверки моделей
- Статический анализ исходного кода

Гибридные методы

- **Гибридные методы** используют несколько разных методов
  - Создание тестов на основе статического анализа
  - Статический или динамический анализ для автоматического формирования моделей, для которых применяются формальные методы проверки моделей
  - Уточнение результатов статического анализа с помощью дедуктивной верификации
  - Комбинирование результатов статического анализа и тестирования для повышения точности результатов

Методы оценки качества

- Методы оценки функциональности
  - Динамические, на основе тестирования программы
  - Статические, на основе методов формальной верификации
- Методы оценки надежности
  - Динамические, на основе прогнозных моделей
  - Статические, на основе метрик сложности и обнаружения дефектов
  - Архитектурные, на основе анализа архитектуры ПО и надежности отдельных компонентов
- Методы оценки эффективности
  - Динамические, на основе профилирования
  - Статические, на основе анализа возможных путей выполнения

Разделы курса

Вводная лекция

1. Динамические методы анализа ПО
2. Статический анализ ПО
3. Верификация ПО на основе моделей
4. Дедуктивная верификация ПО
5. Оценка качества ПО

Динамические методы анализа ПО

1. Основы тестирования
2. Полнота тестирования: оценка и обеспечение
3. Тестирование и жизненный цикл ПО
4. Методы «случайного» тестирования

ЛР №1. Тестирование ПО

Статический анализ ПО

1. Теоретические основы статического анализа ПО
2. Обнаружение программных дефектов методами статического анализа
3. Статический анализ параллельных программ
4. Системы типов и эффектов

ЛР №2. Современные средства статического анализа программ (MS SCA, Aegis)

ЛР №3. Применение систем типов для анализа ПО (Coq)

Верификация ПО на основе моделей

1. Введение в верификацию на основе проверки модели
2. Метод проверки модели. Методы снижения размерности задачи
3. Построение абстрактных программ с помощью SAT-solvers

ЛР №4. Верификация ПО на основе проверки моделей (SPIN)

ЛР №5. Верификация программ с помощью SAT solvers (SatAbs)

Методы дедуктивной верификации

1. Математический аппарат дедуктивной верификации ПО
2. Изоморфизм Карри-Говарда

ЛР №6. Дедуктивная верификация (Frama-C)

Оценка качества ПО

1. Обзор методов оценки качества ПО

Надежность ПО

- Надежность по является одной из важнейшей характеристик качества
- **Надежность ПО** – вероятность его работы без отказов в течении периода времени, рассчитанная с учетом стоимости каждого отказа (Майерс)
- Надежность ПО должна учитывать не только частоту проявления ошибок, но и серьезность их последствий для пользователя системы
- Оценивать и повышать надежность можно на любой стадии проектирования, на основе одного или нескольких представлений программы, при этом можно говорить только о надежности исполняемой программы

Требования к надежности ПО

Для каждой программы можно определить необходимый уровень надежности

Назначение	Область применения	Требования к надежности
Программа, демонстрирующая возможности транзакций	Курс лабораторных работ по предмету «Базы данных»	Время работы – единицы минут, корректно работает при заранее определенных действиях пользователя
Программа расчета заработной платы	Финансовый отдел предприятия	Время работы – десятки часов, устойчива к любым действиям пользователя, обеспечивает восстановление данных после сбоя.
Программа управления срабатыванием подушки безопасности	Автомобиль	Время работы – десятки тысяч часов, устойчива к любым внешним воздействиям, определяет случаи перехода в нерабочее состояние.
Программа управления опасным объектом	Промышленность, авиация и космос, ВС	???

Причины ненадежности

Основными источниками ненадежности аппаратных систем являются внешние факторы, обычно неподвластные человеку:

- скачки напряжения питания;
- электромагнитное излучение;
- радиация;
- …

Источником ненадежности программ являются ошибки, сделанные разработчиками программ, на разных стадиях проектирования

Будем считать программу правильной, если она не содержит ошибок разработчиков, такая программа не дает неверных результатов

Абсолютно надежна

Источники ошибок в ПО

Что такое ошибка в программе ?

Если программа не соответствует

- Спецификации – в ней то же могут быть ошибки
- Неформальным требованиям пользователя – пользователь может не учесть всех возможных ситуаций или неправильно сформулировать свои требования, у программы может быть много пользователей с различными требованиями
- Непредусмотренные входные данные и воздействия
- Ошибки окружения программы – некорректная работа другого ПО и аппаратуры

Является ли луна вражеским объектом ?

Одна из первых компьютерных систем противовоздушной обороны США (60-е годы) в первое же дежурство подняла тревогу, приняв восходящую из-за горизонта Луну за вражескую ракету, поскольку этот «объект» приближался к территории США и не подавал сигналов что он «свой»

Определение надежной программы

В программе имеется ошибка, если она не выполняет действия, которые ожидает от нее некий абстрактный пользователь(эксперт), в том числе и при недопустимых внешних воздействиях и входных данных, а также при отказах другого ПО и сбоях и отказах аппаратуры.

Наличие ошибки – функция самой программы и нереализованных ожиданий ее пользователей (Майерс)

Из этого определения следует:
Программа не имеющая ошибок может давать неверные результаты, но в любом случае минимизирует возможный ущерб

Такой программы не существует

Ошибки в программах

- **Ошибки имеются практически во всех программах**
- Для программ на языке C в среднем

  0,25 нефункциональных ошибок на 1 KLOC
- Примерно 45% ошибок являются критическими
- В ядре ОС Android (765 KLOC) найдено 359 ошибок*

* Coverity Scan: 2010 Open Source Integrity Report

Последствия ошибок в программах

- Переоблучение больных из за ошибки в программе управления радиотерапевтической установкой
  - Печально известная ошибка в линейном ускорителе Therac-25 стала причиной гибели нескольких больных, получивших смертельные дозы радиации во время лечения, проводимого с июня 1985-го по январь 1987 года в нескольких онкологических клиниках в США и Канаде. Эти дозы, как было оценено позже, более чем в 100 раз превышали те, что обычно применяются при лечении. Частично причиной этих несчастий стала ошибка типа race condition.

Последствия ошибок в программах

- Авария при запуске французской ракеты «Ариан-5» (1996)
  - на 37-й секунде полёта компьютер, находившийся на борту ракеты, получил от датчиков системы управления неверную информацию о пространственной ориентации ракеты. Исходя из этой информации, компьютер начал корректировать траекторию полёта для того, чтобы компенсировать несуществующую на самом деле погрешность. Ракета стала отклоняться от курса, что привело к возрастанию нагрузок на её корпус. В результате чрезмерных нагрузок верхняя часть ракеты отвалилась, и по команде с земли ракета была взорвана.

Последствия ошибок в программах

- Неудача при запуске первого американского спутника к Венере
  - Единственная ошибка в программе на Фортране – вместо требуемой в операторе запятой программист поставил точку, в результате аппарат повернул назад.
- Потеря связи с космической станцией «Фобос-1»
  - Произошла из-за ошибочной команды, переданной с Земли на бортовой компьютер
- Ошибка не учета отрицательной высоты
  - При полетах над Мертвым морем американских самолетов произошла ошибка деления на ноль что привело к перезагрузке системы
- Падение спутников системы ГЛОНАСС
  - Три спутника навигационной системы ГЛОНАСС упали в Тихий океан недалеко от Гавайских островов вскоре после их запуска. Причина аварии была признана ошибка в программировании, которая привела к тому, что в ракету залили неправильное количество топлива.

Фобос-Грунт

- «… никаких фатальных ошибок и дефектов при создании станции обнаружено не было". Причиной возникновения "нештатной ситуации", установили специалисты, стал "перезапуск двух полукомплектов устройства ЦВМ22 БВК, выполнявших на этом участке полета управление КА "Фобос-Грунт"…