

ТЕХНОЛОГИЯ РЕГРЕССИОННОГО  
ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
НА ОСНОВЕ ИНКРЕМЕНТАЛЬНОГО  
ДИНАМИЧЕСКОГО АНАЛИЗА  
Software Engineering Conference Russia

Марат Ахин Владимир Ицыксон

Санкт-Петербургский государственный политехнический университет

27-28 октября 2009 г.

- 1 Введение
  - Регрессионное тестирование – что это такое
  - Основная проблема регрессионного тестирования
  - Способы решения этой проблемы
- 2 Выборочное регрессионное тестирование
- 3 Предлагаемая технология ВРТ
- 4 Прототип системы выборочного регрессионного тестирования
- 5 Резюме

# Регрессионное тестирование – что это такое

## Регрессионное тестирование

Тестирование ПО, позволяющее убедиться, что модификация программного кода не вызвала нежелательных побочных эффектов, или что измененная система по-прежнему соответствует предъявляемым к ней требованиям.

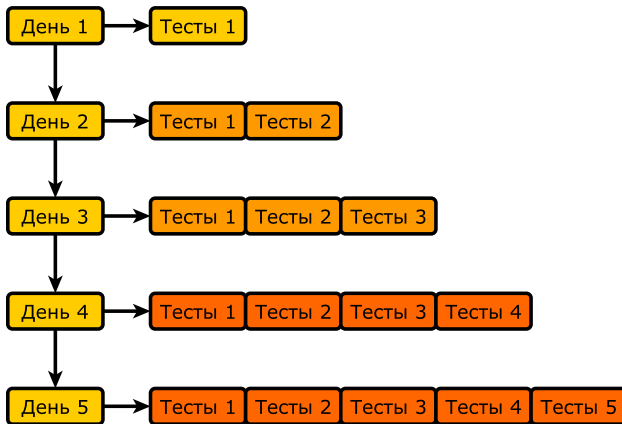
## Классический подход

Повторный запуск всех имеющихся регрессионных тестов после каждого значимого изменения в исходном коде программы.

## Полное регрессионное тестирование

# Основная проблема регрессионного тестирования

## Быстрый рост ресурсоемкости тестирования



## Способы решения этой проблемы

- Regression Testing in an Industrial Environment (A.K. Onoma et al.)
- Analyzing Regression Test Selection Techniques (G. Rothermel et al.)
- Test-suite Reduction and Prioritization for Modified Condition/Decision Coverage (J.A. Jones et al.)
- On the Limit of Control Flow Analysis for Regression Test Selection (T. Ball)
- TestTube: a System for Selective Regression Testing (Yih-Farn Chen et al.)
- Utilization of Extended Firewall for Object-oriented Regression Testing (L. White et al.)
- Program Slicing-based Regression Testing Techniques (R. Gupta et al.)
- A Method for Revalidating Modified Programs in the Maintenance Phase (S.S. Yau et al.)
- ...

## Варианты

- Увеличение производительности
- Установка приоритетов выполнения регрессионных тестов
- Выбор тестов при помощи недетерминированного выбора
- **Выбор тестов на основе анализа ПО**

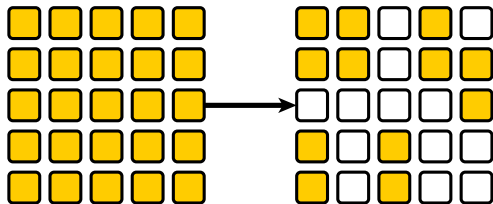
**Выборочное регрессионное тестирование**

- 1 Введение
- 2 Выборочное регрессионное тестирование
  - Основная идея ВРТ
  - Необходимые составляющие ВРТ
- 3 Предлагаемая технология ВРТ
- 4 Прототип системы выборочного регрессионного тестирования
- 5 Резюме

# Основная идея ВРТ

## Сокращение множества выполняемых тестов

На основе анализа ПО мы выбираем подмножество тестов, отвечающее определенным критериям.



В большинстве случаев необходимо выбрать *безопасное* множество тестов – множество тестов, которое способно обнаружить все регрессионные ошибки. Затем вместо полного регрессионного тестирования можно ограничиться тестированием над сокращенным набором тестов без снижения способности к обнаружению ошибок.



## Задачи:

### Анализ изменений в исходном коде

Используется для идентификации модифицированных программных компонентов, требующих повторного тестирования.

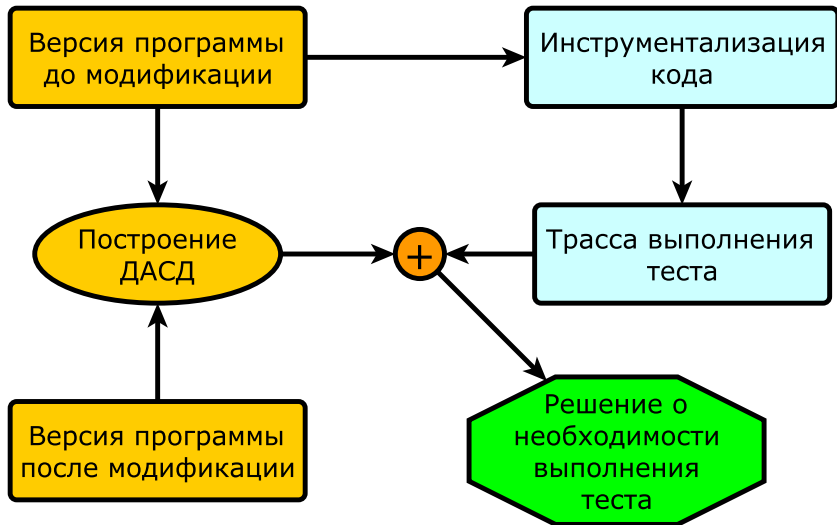
### Динамический анализ выполнения тестов

Используется для определения того, какие программные компоненты покрываются каждым регрессионным тестом.

**Характеристики ВРТ зависят от выбранных способов анализа**

- 1 Введение
- 2 Выборочное регрессионное тестирование
- 3 Предлагаемая технология ВРТ
  - Определение измененных программных компонентов
  - Получение информации о покрытии для регрессионных тестов
  - Выбор безопасного подмножества тестов
  - Ограничения предложенной технологии
- 4 Прототип системы выборочного регрессионного тестирования
- 5 Резюме

# Общая схема технологии ВРТ



# Определение измененных программных компонентов

## Классический подход

Анализ текстовой разницы исходных кодов

---

Недостатки:

- Отсутствие учета синтаксической структуры программы

## Классический подход

Анализ разности CFG программы

---

Недостатки:

- Значительная вычислительная сложность алгоритмов анализа

Что предлагаем мы?

**Анализ разности AST**

## Абстрактные синтаксические деревья

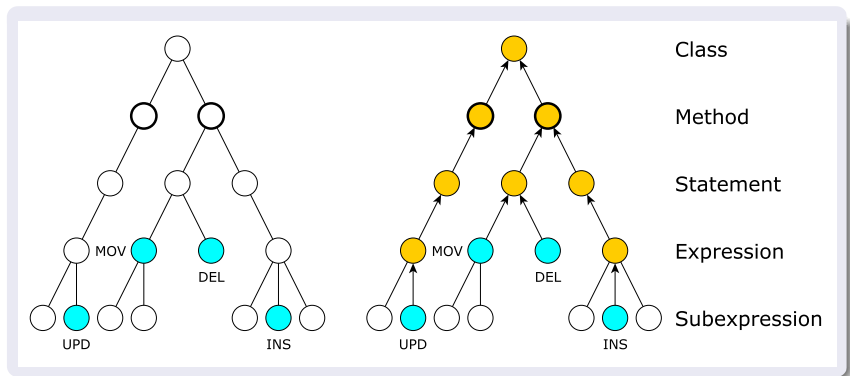
Существуют эффективные алгоритмы анализа изменений в древовидных структурах. Основная идея – представление изменения в виде сценария редактирования.

- *INS*
- *DEL*
- *MOV*
- *UPD*

На основе двух версий AST до и после модификации получаем сценарий перевода  $T_1 \rightarrow T_2$ . Он содержит все изменения для узлов AST.

- Change Detection in Hierarchically Structured Information (S.S. Chawathe et al.)
- Change Distilling: Tree Differencing for Fine-Grained Source Code Change Extraction (B. Fluri et al.)

# Определение измененных программных компонентов



**Множество всех измененных методов программы**

# Получение информации о покрытии для регрессионных тестов

## Классический подход

- Получение трасс тестов при помощи метапрограммирования
- Анализ итоговых трасс в той или иной форме

## Что предлагаем мы?

**Анализ модифицированных трасс выполнения**

# Получение информации о покрытии для регрессионных тестов

## Особенности модифицированной трассы выполнения

### Классическая трасса выполнения

Последовательность вызовов всех методов программы с сохранением порядка

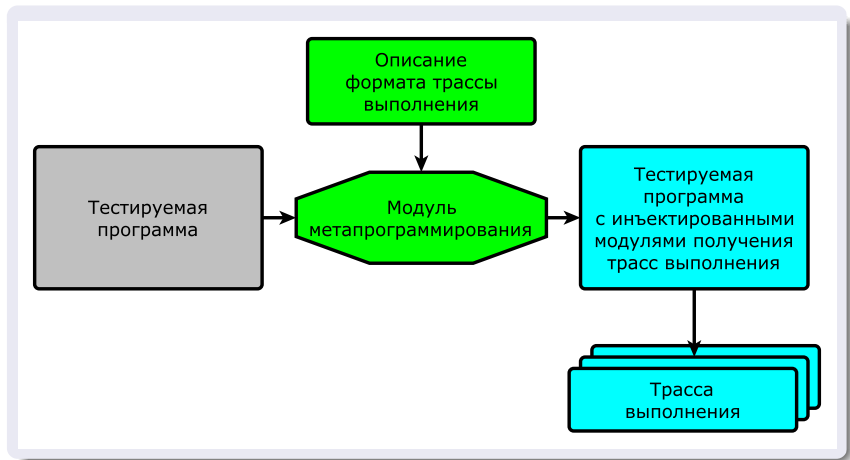
### Модифицированная трасса выполнения

Множество всех вызванных методов программы без сохранения порядка

Такой информации достаточно для решения задачи ВРТ



# Получение информации о покрытии для регрессионных тестов



Набор трасс выполнения регрессионных тестов

## Алгоритм выбора:

- Просмотреть трассы всех регрессионных тестов
- Если трасса теста содержит хотя бы один измененный метод программы, то тест включается в множество тестов для повторного выполнения

## Итог:

**Безопасное подмножество регрессионных тестов**

Предложенная технология не является безопасной, если:

- Регрессионные тесты не являются детерминированными
- Изменилось тестовое окружение
- В тестируемой программе есть элементы метапрограммирования

**В большинстве случаев данные ограничения выполняются**

# Содержание

- 1 Введение
- 2 Выборочное регрессионное тестирование
- 3 Предлагаемая технология ВРТ
- 4 Прототип системы выборочного регрессионного тестирования
  - Общая структура
  - Экспериментальные исследования
- 5 Резюме

## Основные компоненты

- Библиотека анализа древовидных структур
  - `com.panasonic.spbspu.fee.patterns`
- Библиотека получения трасс тестов
  - `ice.jaster.trace`
- Библиотека работы с исходным кодом Java
  - `ice.jaster.dast`
- Дополнительная функциональность
  - `ice.jaster.scripts`
  - `ice.jaster.junit`
  - `ice.jaster.util`

## Тестовые проекты

### Feature Extractor/Editor (*FEE*)

- Совместный проект с компанией Panasonic по поиску шаблонов исходного кода в программах на языке C
- Около 4 тысяч строк исходного кода
- Около 20 регрессионных тестов

### Static Software Analyzer (*s<sup>2</sup>a*)

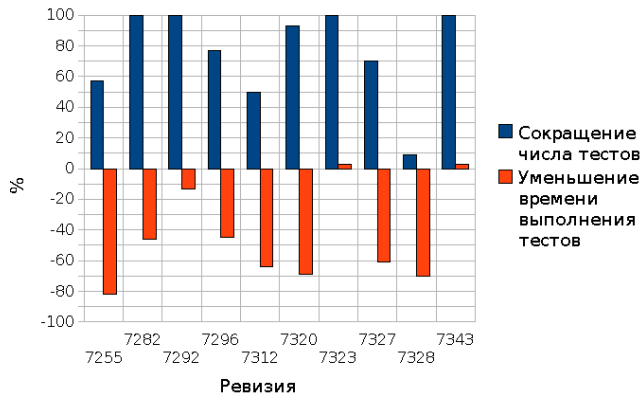
- Проект по статическому анализу программ на языке C/C++
- Более 40 тысяч строк исходного кода
- Более 500 регрессионных тестов

# Экспериментальные исследования

$$Efficiency = 1 - \frac{P_{with\ RTS}}{P_{without\ RTS}}$$
$$\sum_{t \in S} time(t) < RTS\ Time + \sum_{t \in S'} time(t)$$

4 тыс. строк  
20 тестов

Эффективность ВРТ для FEE

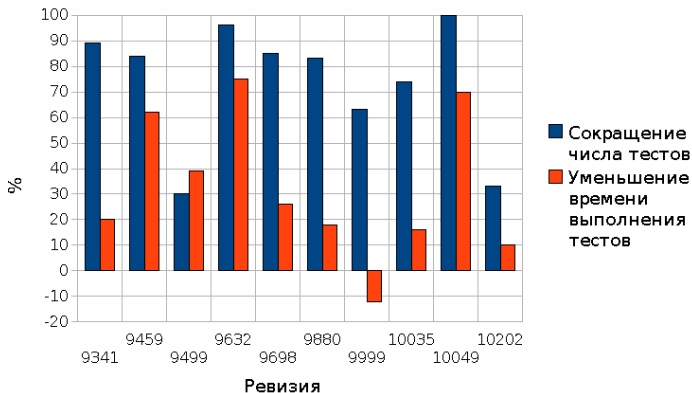


# Экспериментальные исследования

$$Efficiency = 1 - \frac{P_{with\ RTS}}{P_{without\ RTS}}$$
$$\sum_{t \in S} time(t) > RTS\ Time + \sum_{t \in S'} time(t)$$

40 тыс. строк  
500 тестов

Эффективность ВРТ для s2a





# Содержание

- 1 Введение
- 2 Выборочное регрессионное тестирование
- 3 Предлагаемая технология ВРТ
- 4 Прототип системы выборочного регрессионного тестирования
- 5 Резюме

## Основные теоретические результаты

- Предложена технология выборочного регрессионного тестирования на основе инкрементального динамического анализа
- Показана безопасность предложенной технологии при выполнении определенных ограничений

## Основные практические результаты

- Разработан прототип системы выборочного регрессионного тестирования
- Выполнена апробация прототипа на реальных программных проектах

## Теоретические перспективы

- Развитие предложенной технологии для улучшения показателей выборочного регрессионного тестирования
- Применение семантического анализа для анализа изменений

## Практические перспективы

- Создание модуля расширения для IDE, CI, и т.д. на основе предложенного подхода
- Внедрение технологии в реальный жизненный цикл разработки ПО

# Используемые обозначения и сокращения

ВРТ	Выборочное регрессионное тестирование
ДАСД	Дифференциальное абстрактное синтаксическое дерево
ПО	Программное обеспечение
AST	Abstract syntax tree
CFG	Control flow graph