

Анализ подходов и средств для верификации программ. Использование фреймворков как средств автоматизации.

А.М. Полоцев гр. 63501/13

Современные проблемы информатики и вычислительной техники

Качество программного обеспечения - это совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

Верификация проверяет соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам.

Валидация проверяет соответствие любых создаваемых или используемых в ходе разработки и сопровождения ПО артефактов нуждам и потребностям пользователей и заказчиков этого ПО, с учетом законов предметной области и ограничений контекста использования ПО.

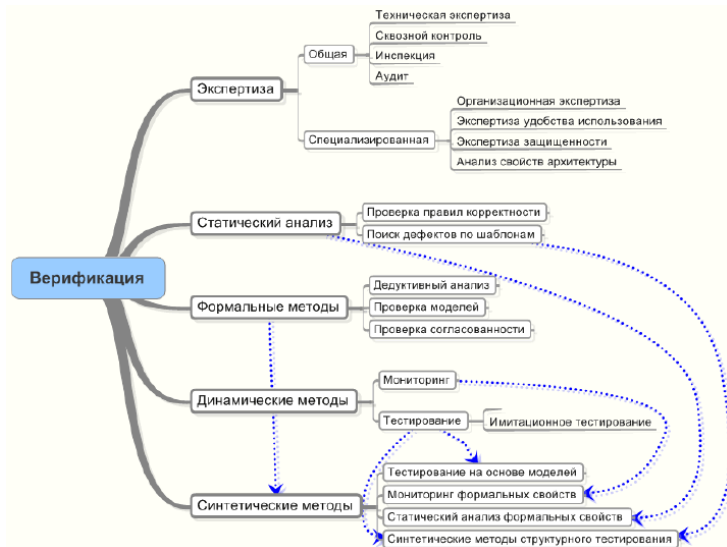
Актуальность проблемы

Сложность систем растет, сроки выпуска продуктов уменьшаются \Rightarrow качество падает.

Здесь должны быть примеры про падающие самолеты, облучение людей и т.д.



Методы обеспечения качества



При проведении анализа ПО часто возникают следующие задачи:

① Построение моделей

- Абстрактное синтаксическое дерево
- Граф потока управления
- Граф зависимостей по данным
- Граф программных зависимостей
- Абстрактный семантический граф
- Статическое однократное присваивание
- ...

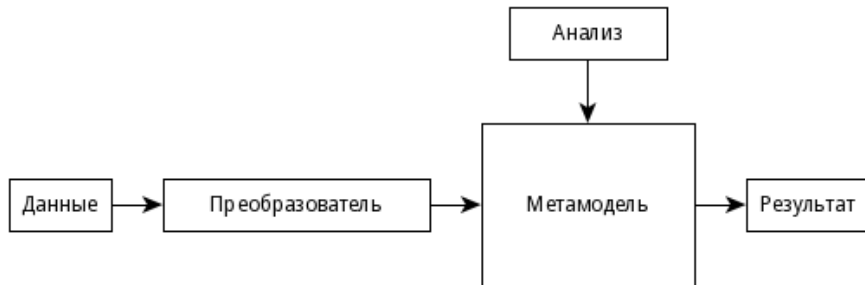
② Построение метрик

- Метрики Холстеда
- Метрики Константейна и Йордана
- Метрики Отта и Мехра
- Метрики Абреу
- LOC
- ...

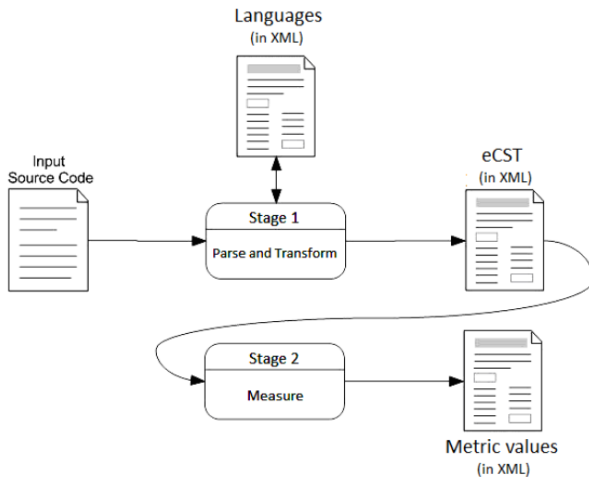
Для автоматизации анализа сложного ПО используются специализированные инструментальные средства - фреймворки. Решаемые задачи:

- Построение метрик.
- Построение моделей программ и применение различных алгоритмов над этими моделями.
- Интерактивная визуализация процесса анализа.

Упрощенная структура фреймворка

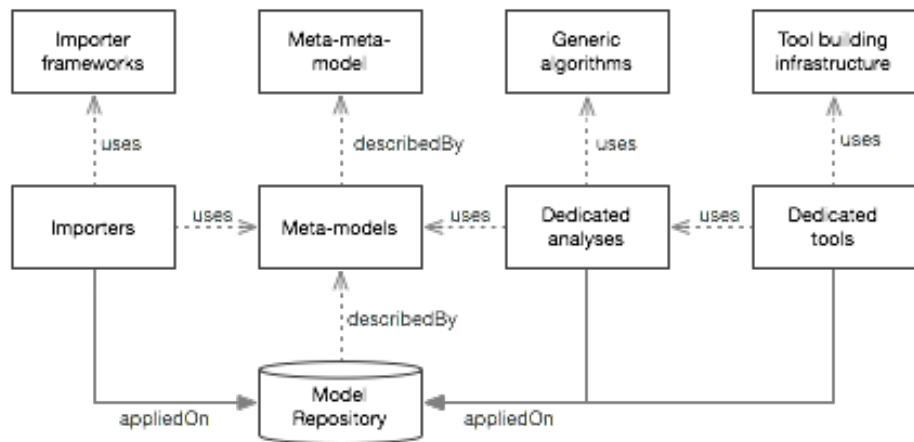


- Фреймворк для построения метрик.
- В качестве метамодели используется eCST (enriched Concrete Syntax Tree).
- Находится в стадии разработки.



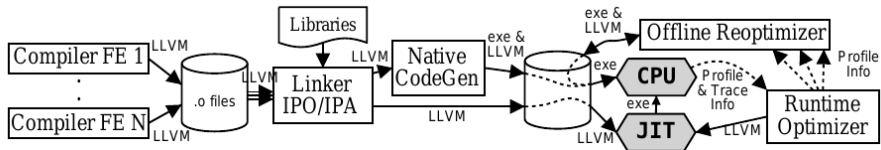
Moose является платформой для анализа программ и поддерживает большое количество различных видов анализа:

- 1 Построение и визуализация метрик.
- 2 Обнаружение клонов.
- 3 Построение графа зависимостей между пакетами.
- 4 Вывод словаря, используемого в проекте.
- 5 Поддержка браузеров исходного кода.



LLVM (Low Level Virtual Machine) - фреймворк для построения компиляторов, предназначен для анализа и трансформации программ.

LLVM использует промежуточное представление, в основе которого лежит представление в виде SSA. Промежуточное представление является набором RISC-подобных команд и содержит дополнительную информацию более высокого уровня, например информацию о типах.



Требования к разрабатываемому фреймворку

Требуется создать среду для анализа программ, удовлетворяющую следующим требованиям:

- 1 Предоставлять API на языке Java для проведения анализа, а именно для построения различных метрик и моделей.
- 2 Иметь модульную структуру - позволять подключать пользовательские алгоритмы анализа.
- 3 Отображать полученные результаты в графическом виде.

Все вышеперечисленные решения данным требованиям не удовлетворяют.

Требования к разрабатываемому фреймворку

Требуется создать среду для анализа программ, удовлетворяющую следующим требованиям:

- 1 Предоставлять API на языке Java для проведения анализа, а именно для построения различных метрик и моделей.
- 2 Иметь модульную структуру - позволять подключать пользовательские алгоритмы анализа.
- 3 Отображать полученные результаты в графическом виде.

Все вышеперечисленные решения данным требованиям не удовлетворяют.

А это значит, что...

Hoist the code!

Нужно писать свой собственный!



