

ADAPTSUITE - FERRAMENTA PARA SELEÇÃO AUTOMÁTICA DE TESTES



IME-USP



USP

ANDRÉ PESTANA CIÊNCIA DA COMPUTAÇÃO - IME/USP
ORIENTADOR: ALFREDO GOLDMAN <GOLD@IME.USP.BR>

OBJETIVOS

A criação desse projeto tem como objetivo desenvolver uma técnica de seleção automática de testes JUnit visando:

1. Garantir a validação do sistema quando não há tempo para se rodar todos os testes
2. Encontrar erros de maneira eficaz
3. Verificar que uma alteração realmente corrigiu um determinado erro

METODOLOGIA

O problema que a ferramenta resolve é encontrar a melhor suíte de testes para ser executada dentro de um tempo limite. Cada teste tem sua própria duração e seu próprio peso. Abstraindo, esse problema, percebe-se que o mesmo se trata do "Problema da Mochila".

No caso da AdaptSuite, o peso de cada teste é definido pelas seguintes variáveis:

- Quantidade de falhas registradas pelo teste
- Cobertura
- Última vez que o teste foi executado
- Frequência com que o teste é executado pela ferramenta

É uma prioridade para cada variável de 0 à 5 definida pelo usuário.

Seja T um teste qualquer, E_T a quantidade de erros referente ao teste T , C_T a cobertura, U_T a última vez que foi executado, F_T e P_1, P_2, P_3, P_4 os pesos desses parâmetros que podem ser modificados pelo usuário. O valor V_T associado à esse teste é definido como:

$$V_T = (P_1 * E_T) * (P_2 * C_T) * (P_3 * U_T) * (P_4 * F_T) \quad (1)$$

FUNCIONAMENTO

Para se usar a ferramenta, basta instanciar o seguinte objeto:

```
import main.java.org.adaptsuite.suite.AdaptSuiteBuilder;

@RunWith(AllTests.class)
public class Seconds3Suite {
    public static TestSuite suite(){
        return new AdaptSuiteBuilder().sec(3).build();
    }
}
```

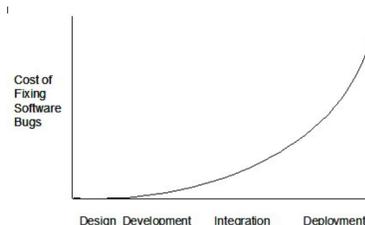
Figura 3: Iniciando a ferramenta

Pontos importantes desse objeto são:

- Dessa forma, todas as prioridades serão definidas como 1.

INTRODUÇÃO

Testar um sistema, embora seja algo crucial, é reconhecido como uma atividade cara. Sistemas grandes possuem muitos testes e testá-los em sua totalidade, além de custoso, é demorado. Porém, problemas não detectados na etapa de teste são mais caros para serem consertados.



PROBLEMA PROPOSTO

O problema que a ferramenta AdaptSuite tenta resolver é encontrar o melhor conjunto de testes quando não há tempo suficiente para se rodar todos os testes.

Um exemplo de quando esse problema acontece é quando um sistema entra no estágio de "Early Majority" na curva de adoção de produto, o número de clientes cresce exponencialmente, fazendo com que a demanda por novas features e correção de bugs aumente. Mesmo nesse cenário, deseja-se manter o ritmo de deploy, resultando em um menor tempo para a etapa de testes.

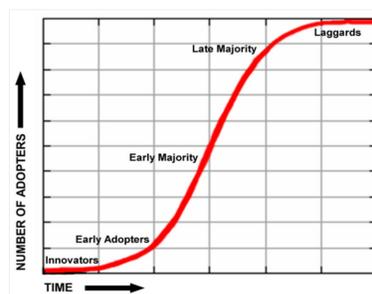


Figura 5: Curva de adoção de produto de Rogers

RESULTADOS

Considere o caso em que todos os testes funcionam perfeitamente até o momento em que um deles falhará na próxima execução e deseja-se saber qual o tamanho da suíte mínima para se encontrar tal teste. O AdaptSuite foi testado comparando-o ao algoritmo guloso que resolve o mesmo problema e ao de escolha aleatória.

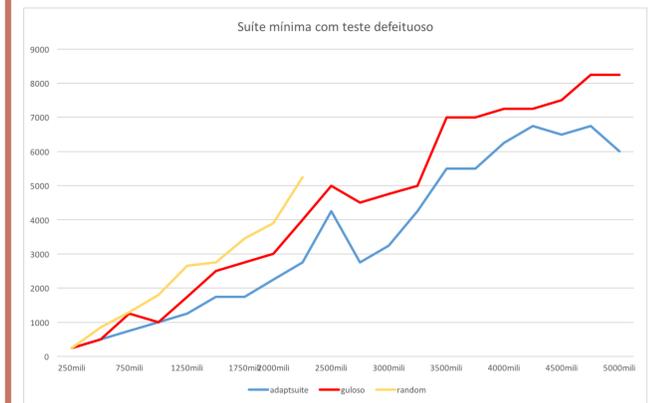


Figura 1: Procurando teste defeituoso

Em um segundo caso, considere que um erro apresentou defeito e o mesmo foi corrigido. Deseja-se agora encontrar a suíte mínima que execute esse teste e verifique que o erro foi corrigido.

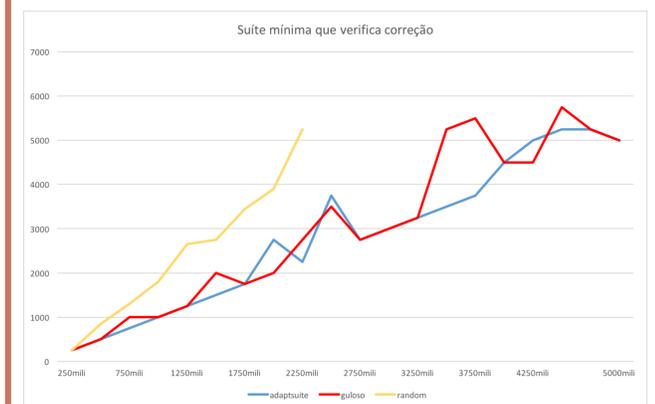


Figura 2: Verificando que o erro foi corrigido

PRÓXIMOS PASSOS

Como a seleção de uma suíte ótima se trata de um problema aberto na comunidade, não existem muitos algoritmos que tentem resolvê-lo, logo, não há muitas opções para compará-lo à eficiência do AdaptSuite. Dito isso, os possíveis próximos passos seriam:

- Otimizar a escolha da suíte de testes pelo algoritmo guloso e tentar encontrar mais variáveis para serem consideradas no cálculo da obtenção da suíte.
- Resolver esse mesmo problema considerando que as variáveis estejam em planos distintos. Para tal, seria necessário estudar técnicas de Teoria dos Jogos e Programação Linear.

REFERÊNCIAS

- [1] Pressman, R. S. Engenharia de Software - Uma Abordagem Profissional. AMGH Editora 7.ed., 2011
- [2] Rogers, E. M. Diffusion of innovations. New York Free Press 5.ed., 2003
- [3] Wu, K., Fang, C., Chen, Z., Zhao, Z. Understanding the effects of changes on the cost-effectiveness of regression testing techniques., 2012

CONTATO

GitHub github.com/andrepestana92
Email andre.pestana@usp.br
Phone +55 (011) 95791-2981