Técnicas de teste

Em uma visão simplificada de um processo de teste de software, os desenvolvedores realizam duas tarefas distintas: projetar casos de teste e executar casos de teste.

A primeira tarefa, como acabamos de mencionar, é sobre analisar e projetar casos de teste. Em palavras simples, o objetivo deste estágio é conceber sistematicamente diferentes casos de teste que, juntos, nos darão um certo nível de confiança quanto a saber se o software está pronto para ser enviado ou não. Elaborar casos de teste é uma atividade frequentemente realizada por humanos (embora iremos explorar o que há de mais moderno em pesquisas de teste de software, onde as máquinas também tentam criar casos de teste para nós). Afinal, é necessário um bom conhecimento dos requisitos. No exemplo dos numerais romanos de um capítulo anterior, criamos três casos de teste durante a fase de design do caso de teste.

A segunda tarefa é sobre a execução dos casos de teste que planejamos. Freqüentemente, fazemos isso executando o sistema de software, alimentando-o com as entradas que criamos e verificando se o sistema respondeu da maneira esperada. Embora essa fase também possa ser realizada por humanos, essa é uma atividade que podemos automatizar facilmente. Como discutimos antes, podemos (e devemos) escrever um programa que execute nosso software e execute os casos de teste.

Como uma observação lateral, na indústria, o termo "teste de software automatizado" geralmente está relacionado à execução automática de casos de teste (que foram criados manualmente). Na academia, sempre que um artigo de pesquisa diz "teste automatizado de software", isso significa criar casos de teste automaticamente (por meio de técnicas de inteligência artificial, por exemplo).

Quando se trata de desenvolver casos de teste, embora nossa experiência nos ajude muito a encontrar bugs, pode não ser suficiente:

O teste baseado na experiência é altamente sujeito a erros. O desenvolvedor pode esquecer de testar um caso de canto.

Isso varia de pessoa para pessoa. Nosso objetivo é definir técnicas para que qualquer desenvolvedor no mundo seja capaz de testar qualquer software.

Sem critérios claros, é difícil saber quando parar o teste. Nossos instintos podem não ser precisos o suficiente.

Os capítulos a seguir têm como objetivo explorar diferentes técnicas para testar um sistema de software de maneira eficaz, rigorosa e sistemática e como automatizar o maior número possível de etapas ao longo do caminho. Essas técnicas contarão com os diferentes artefatos que estão presentes durante o processo de desenvolvimento de software.

Mais especificamente, discutiremos os seguintes tópicos:

Teste baseado em especificações: técnicas para derivar testes de requisitos textuais. Compreender o método de categoria / partição e particionamento de equivalência.

Teste de limite: Derivando testes que testam os limites de nosso requisito.

Teste estrutural: casos de teste baseados na estrutura do código fonte.

Teste baseado em modelo: Aproveitando a documentação mais formal, como máquinas de estado e tabelas de decisão para derivar testes.

Projeto por contratos: Elaborar contratos explícitos para métodos e classes para garantir que eles se comportem corretamente quando esses contratos são (e não são) atendidos.

Teste baseado em propriedades: derivando propriedades do sistema (semelhantes aos contratos) e usando-as para gerar casos de teste automaticamente.