

Qualidade de Software

Gabriel França

Objetivos deste encontro

- Conhecer conceitos de Teste de Software
- Como o Teste impacta em meu projeto
- O que devo testar
- Como devo testar
- Artefatos de teste
- Ferramentas de teste

Objetivos

- Ferramentas mais usadas
- Como as usar
- Automação de testes
- Quando aplicar automação
- Como começar a automatizar
- HP UFT – Selenium
- Certificações Profissionais

O que é Teste de Software

- Teste de software é a etapa de controle de qualidade, serve para assegurar que o software está contemplando todas as funcionalidades esperadas e que estas estão funcionando corretamente. Geralmente é a última etapa na construção de um sistema, aplicativo ou game a ser lançado, podendo ser um novo projeto ou uma nova versão.

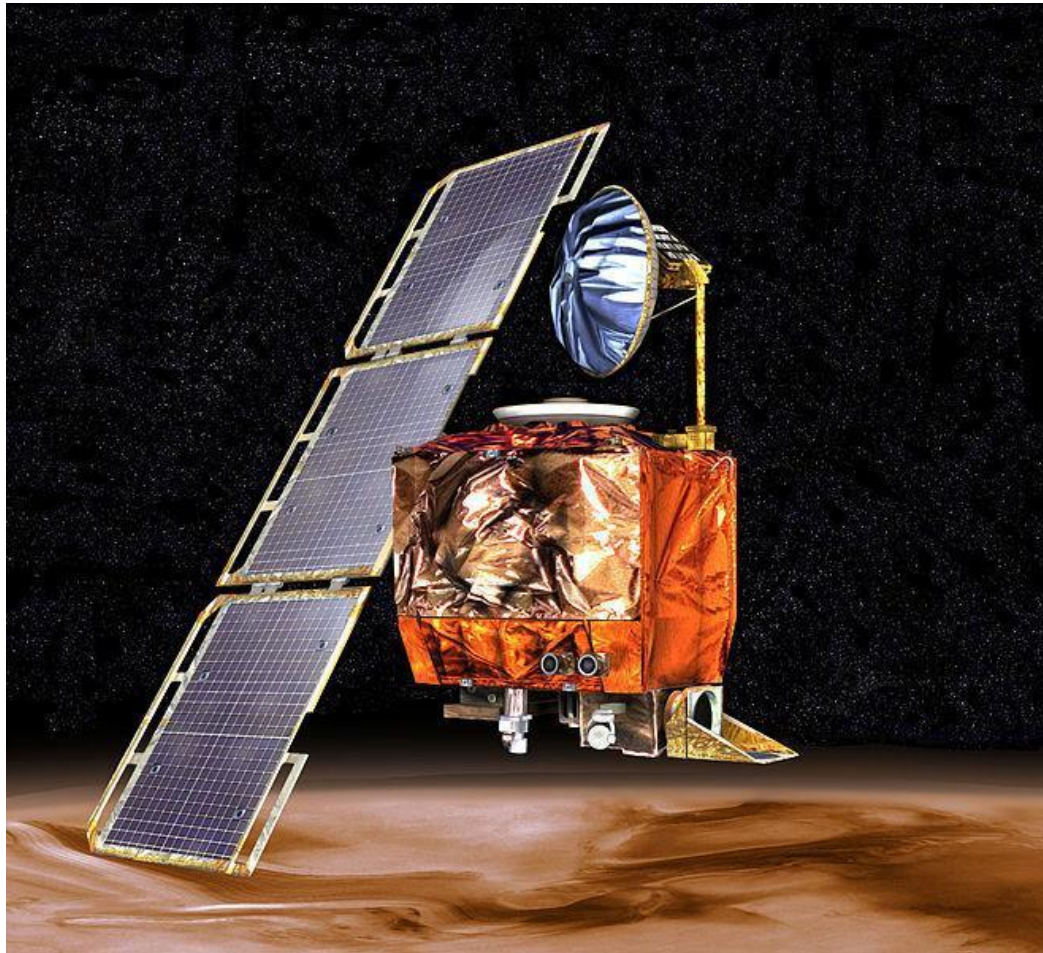
Por que o Teste é importante

- Grupos de 5 alunos
- 15 minutos

Por que testar

- Mars Climate Orbiter
- Apple maps

Mars Climate Orbiter



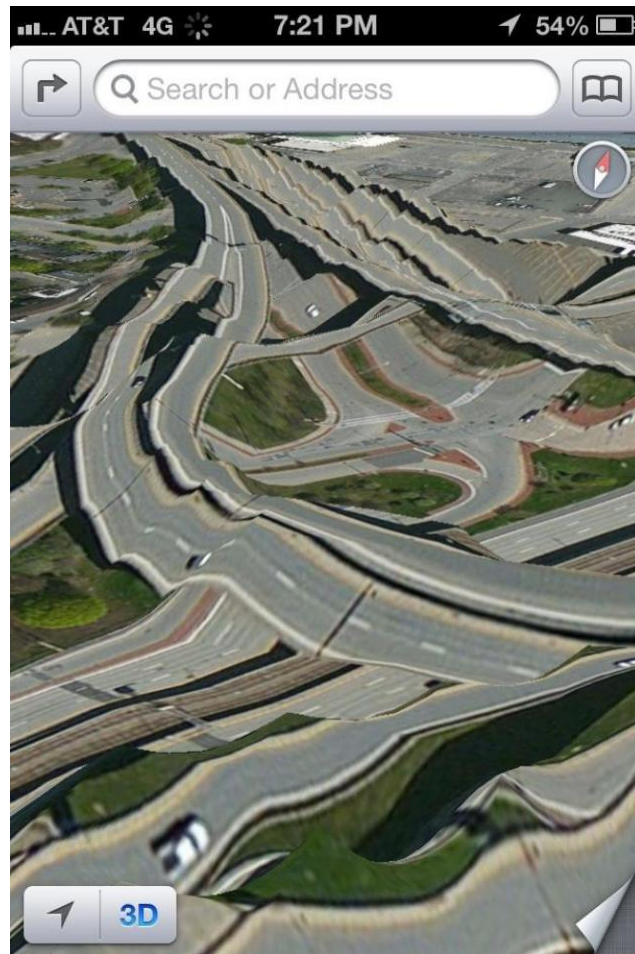
Mars Climate Orbiter



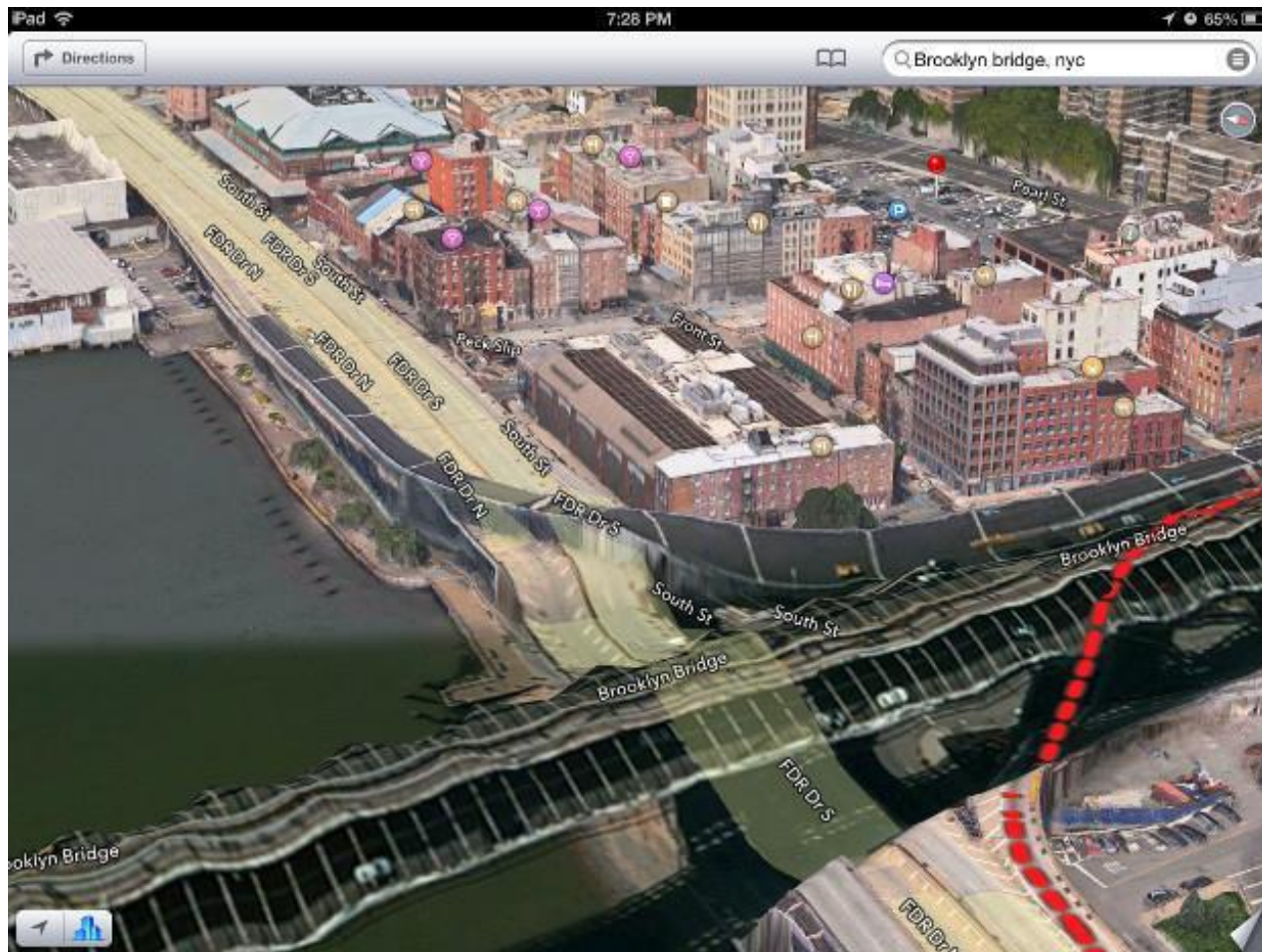
O que aconteceu ?

- National Aeronautics and Space Administration (NASA)
- Lockheed Martin
- O problema
 - *Libras/Segundo*
 - *Newton/Segundo*

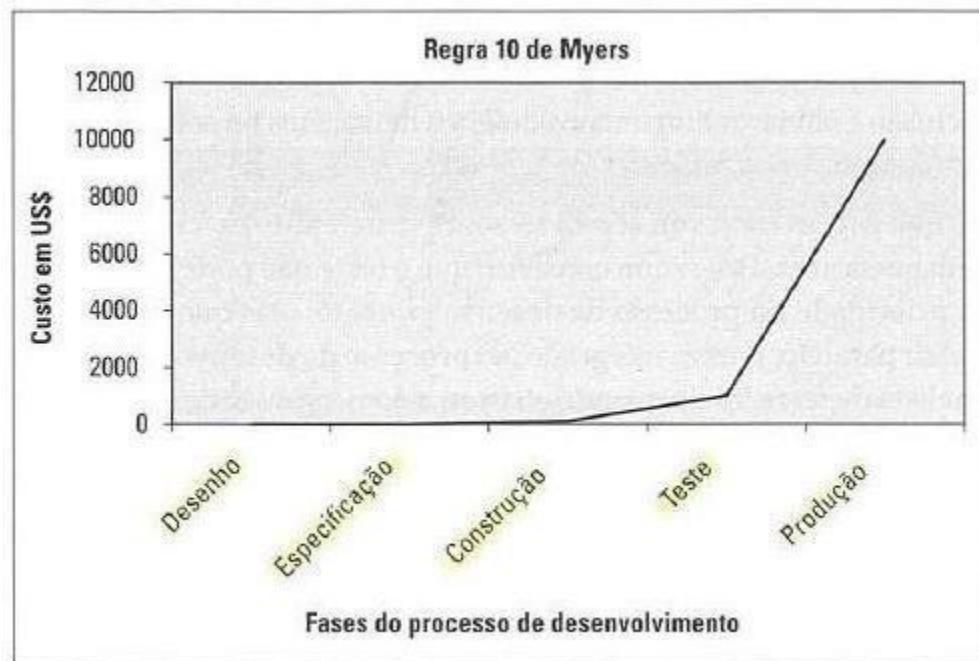
Apple Maps



Apple Maps



Quanto custa um defeito?



Diferenciar Verificação de Validação

- Verificação
 - Inspecionar e verificar os produtos gerados pelas etapas do processo de teste
- Validação
 - Avaliar se o sistema atende aos requisitos do projeto(Usuário)

Quanto tempo investir em teste?

Etapa	Distribuição de Tempo
Planejamento	Calcular à parte
Preparação	10%
Especificação	40%
Execução	45%
Entrega	5%

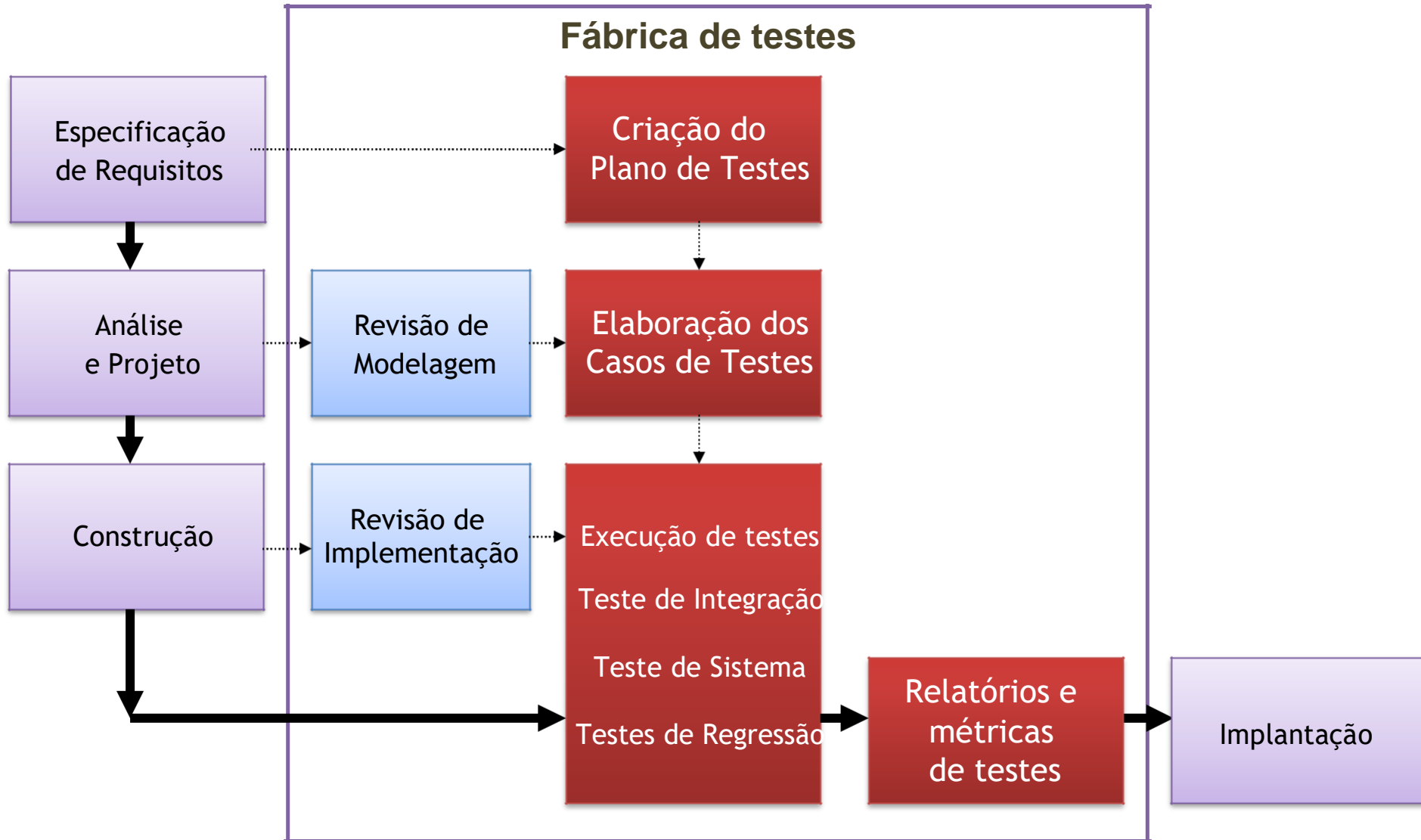
Verificação

- Revisão de requisitos
- Revisão de modelo
- Code review

Validação

- Teste unitário
- Teste de integração
- Teste de sistema
- Teste de aceitação

Fluxo de atividades de teste



Teste Estrutural Vs Teste Funcional

- Estrutural
 - Estruturalmente sólido
 - Funciona no contexto onde vai ser usado
- Funcional
 - Requisitos são atendidos
 - Sistema foi desenvolvido como foi especificado

Teste Estrutural

- Teste de estresse
 - Sistema processa grande volume de dados
 - Sistema responde no tempo esperado
- Teste de segurança
 - Acesso a área restrita
 - Acessibilidade dos dados
 - Regras de acesso

Teste Funcional

- Testes de Requisitos
 - Especificação do sistema foi implementada
- Teste de Regressão
 - Funcionalidades continuam estáveis
 - Correções impactam em funcionalidades existentes

Teste Funcional

- Teste de tratamento de erros
 - Erros são esperados pelo sistema
- Teste de interconexão
 - Respostas de serviços

Ambiente de testes

- Escopo
- Equipe
- Dados
- Ambiente

Escopo

- Teste Unitário
 - Estágio mais baixo da escala
 - Aplicado aos menores componentes do código
- Teste de integração
 - Aplicado a combinação das unidades de componentes

Escopo

- Teste de Sistema
 - Aplicado ao sistema como um todo
- Teste de aceitação
 - Teste final do sistema
 - Funcionalidade e usabilidade

Equipe

- Líder do projeto de teste
- Arquiteto de teste
- Analista de teste
- Testador

Equipe

- Líder do projeto de teste
 - Lidera o esforço de testes
 - Novo sistema
 - Manutenção

Equipe

- Arquiteto de teste
 - Monta a infra estrutura e ambiente
 - Seleciona ferramentas
 - Capacita equipe para usar ferramentas

Equipe

- Analista de teste
 - Levanta casos de teste
 - Elabora casos de teste

Equipe

- Testador
 - Executa casos de teste
 - Reporta problemas encontrados

Documentos

- Plano de teste
- Cenários de teste
- Casos de teste
- Relatório de teste
- Bugs

Plano de Teste

- O **plano de teste** é um documento com uma abordagem sistemática para o **teste** de sistemas como hardware ou software. Ele geralmente consiste numa modelagem detalhada do fluxo de trabalho durante o processo.

Componentes do Plano de Testes

- Descrição breve do sistema
- Descrição das funcionalidades a serem testadas
- Descrição das funcionalidade que não serão testadas
- Descrição do ambiente de testes
- Definição das ferramentas que serão utilizadas

Componentes do Plano de Testes

- Definição do fluxo de defeitos
- Definições dos marcos
- Definição da equipe e papéis

Descrição do sistema

- Descrição breve do contexto da aplicação funcionalidades e por que a aplicação existe

Funcionalidades a serem testadas

- Funcionalidades que serão verificadas pela equipe de teste
- Casos de uso
- Fluxos

Funcionalidades que não serão testadas

- Escopo que não seja coberto pelos testes
- Normalmente requisitos do ambiente

Ambiente de testes

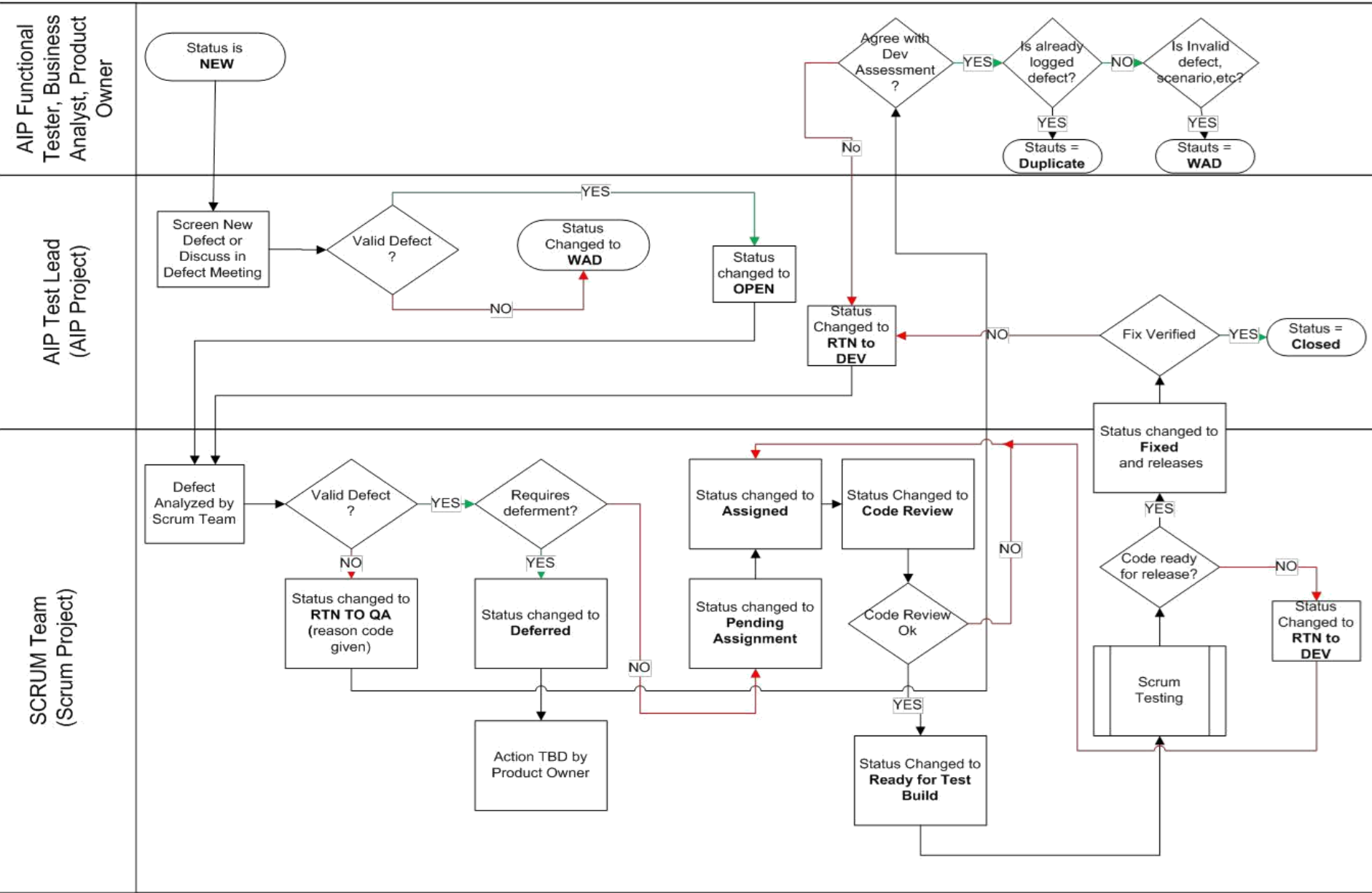
- Versões do OS
- Versões de Browser
- Resolução de tela
- Velocidade de conexão
- Qualidade de conexão

Ferramentas

- Ferramentas utilizadas para os testes
- Controle de execução de casos de teste
- Controle de bugs
- Controle de versões
- Gerador de build

Fluxos de defeitos

- Como os defeitos serão tratados
- Estados
 - Funciona como Especificado
 - Aberto
 - Fechado
 - Corrigido
 - Indeferido
 - Aguardando atribuição
 - Atribuído
 - Pronto para teste
 - Retornado ao QA
 - Review de código



Marcos

- Entregas de versões
- Sprints
- Datas de releases

Equipe

- Plano de comunicação
- Quem
- O que
- Contatos

Vamos Trabalhar

- Dentro do contexto exposto elaborar os seguintes componentes de um plano de testes
- Descrição do sistema
- Descrição das funcionalidades a serem testadas
- Descrição das funcionalidades que não serão testadas
- Descrição do ambiente de testes
- Definição do fluxo de defeitos
- Definições dos marcos
- Definição da equipe e papeis

Cenários

- E-commerce
- Agendamento de consultas
- Gerenciamento de Biblioteca
- Aluguel de carros
- Ponto Eletrônico
- Acompanhamento de atividades(kanban)

Elaboração de testes

- Cenários de teste
- Casos de teste

Cenários de teste

- Cenários de teste são uma classificação de alto nível de testes de um requerimento agrupado mediante a funcionalidade de um módulo

Exemplo

Req	ID Cenário	Cenário de teste	Nº de casos de teste
UC01	TC001	Verificar a função de login para garantir que somente usuários cadastrados conseguem acessar o sistema	4
UC02	TC002	Validar a funcionalidade adicionar ao carrinho	5
UC03	TC003	Validar a funcionalidade remover do carrinho	4

Vamos trabalhar um pouco

- E-commerce
- Agendamento de consultas
- Gerenciamento de Biblioteca
- Aluguel de carros
- Ponto Eletrônico
- Acompanhamento de atividades(kanban)

Caso de teste

- Define-se formalmente um caso de teste como a especificação mais detalhada do teste, com a pormenorização de campos de telas, formulários etc. Estabelece quais informações serão empregadas durante os testes dos cenários e quais serão os resultados esperados. Para isso, é necessário determinar a massa a ser utilizada no teste de modo a validar todos os requisitos do software.

Componentes do caso de teste

- Objetivo
- Pré-Condições
- Pós condições
- Passos
- Ambiente

Com base neste caso de uso

Nome	CT01 01 Login Inválido
Objetivo	Verificar se o sistema apresenta mensagem de erro ao informar login invalido e senha válida
Pré-condições	Estar na tela de Login Usuário possui uma senha válida
Pós condições	Sistema apresenta mensagem de erro
PA	Usuário informa um login inválido e uma senha valida
PA	Usuário clica no botão OK
PS	O sistema verifica se os campos obrigatórios foram informados
PS	O sistema verifica se o login do usuário está cadastrado e se a senha para o login especificado é válida
PS	O sistema exibe mensagem com o texto "O login ou a senha informados não são válidos" e retorna para o passo anterior

Vamos Trabalhar

- Elaborar casos de teste suficientes para cobrir as regras e mensagens do caso de uso proposto

BUG

- Report de um problema no software
 - Título
 - Severidade
 - Prioridade
 - Descrição
 - Plano geral
 - Passos para reproduzir o problema
 - Evidências

Prioridade

- Crítica
 - Deve ser corrigida imediatamente dentro de 24 horas. Geralmente usada em casos onde uma funcionalidade inteira é bloqueada e os testes não podem proceder.
- Alta
 - Quando todos os defeitos criticos foram corrigidos, um defeito desta prioridade é um candidato a ser corrigido para qualquer atividade de teste que não atinja o resultado esperado.

Normalmente uma funcionalidade que não está utilizavel.

Prioridade

- Média
 - Um defeito com esta prioridade deve ser corrigido, uma vez que também pode estar ligado a problemas de funcionalidade que não estão funcionando adequadamente. Erros, por vezes, até mesmo cosméticos, tais como espera a mensagem de erro correta durante uma falha esperada poderia se qualificar para um defeito de prioridade 3.
- Baixa
 - Um defeito com prioridade baixa indica que existe um problema, mas não tem de ser corrigido para que a funcionalidade atinja seu resultado esperado.

Severidade

- **Critica**
 - Um defeito que interrompe completamente o teste de um produto ou funcionalidade é um defeito crítico.
- **Severa**
 - Uma funcionalidade que está em grande não conformidade com a especificação
- **Normal**
 - Quando a aplicação não age conforme o esperado ou não atinge certo critério porem a funcionalidade como um todo não é afetada
- **Menor**
 - Um problema menor que não tem impacto na funcionalidade, porem ainda é um defeito encontrado na aplicação.

FERRAMENTAS

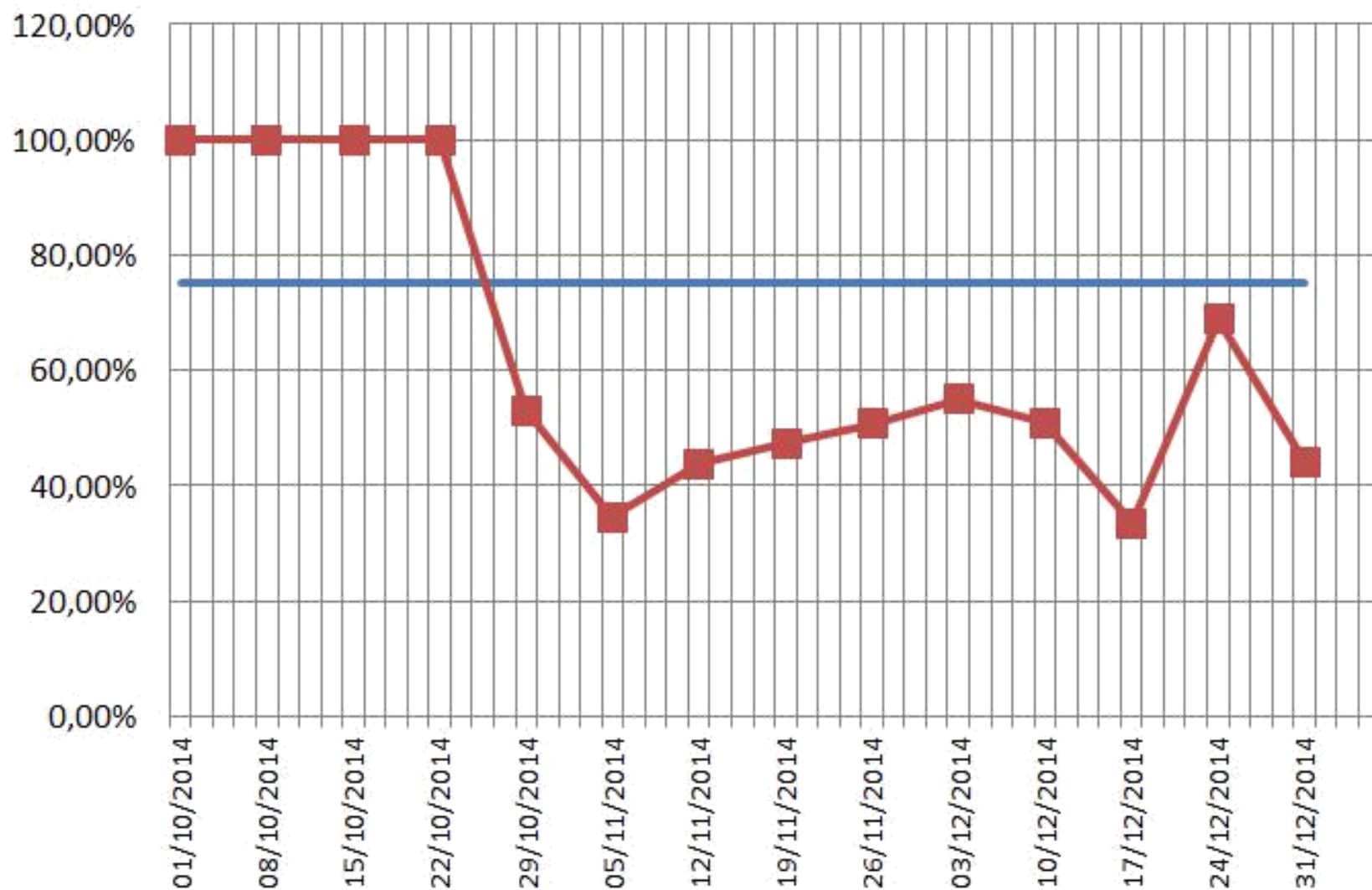
- BUGS
- Testes

Analizando defeitos

- Efetividade dos testes
 - Quantos defeitos eu encontrei
 - Quantos defeitos meu cliente encontrou

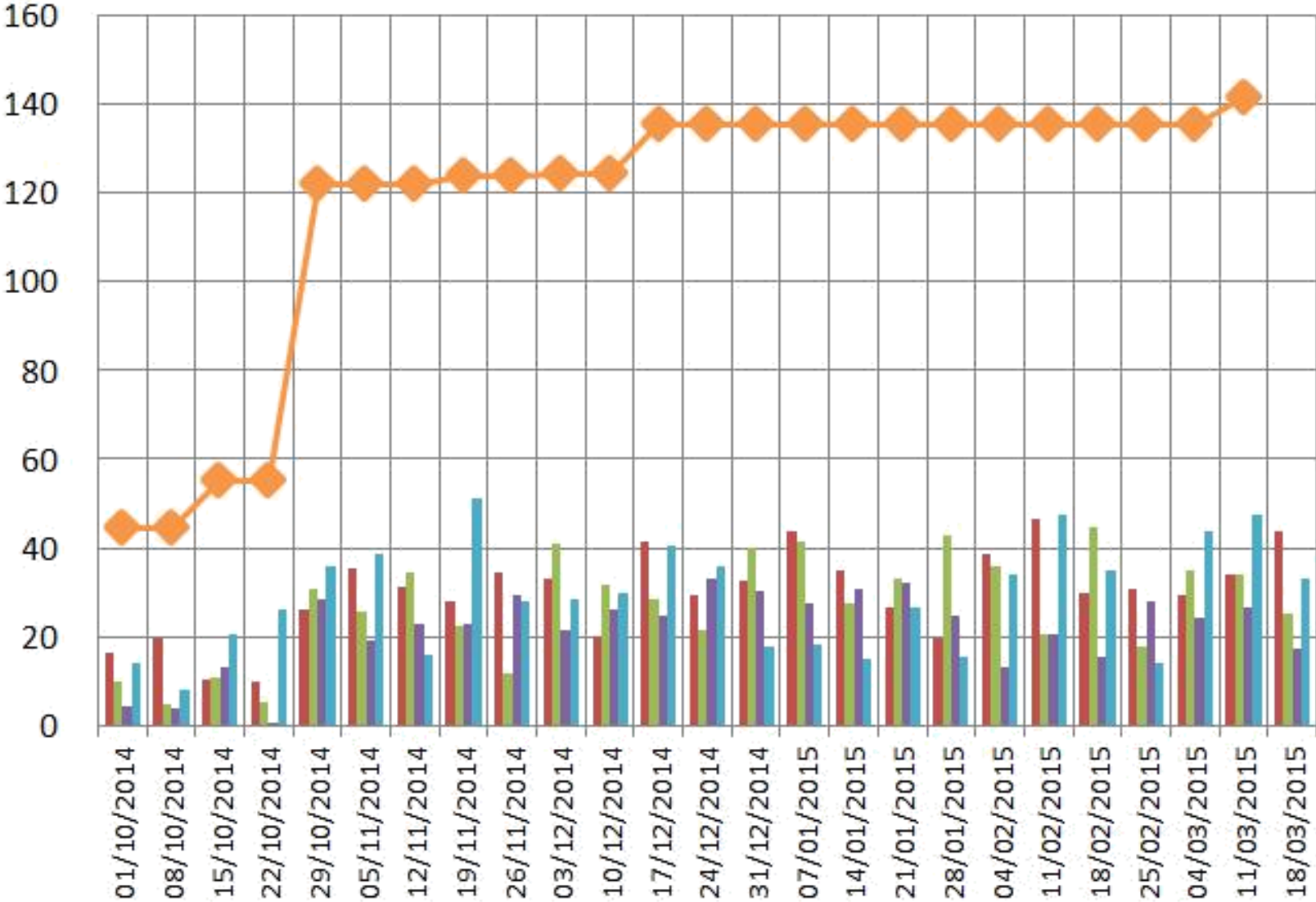
Efetividade dos Testes

— Meta —■— Efetividade



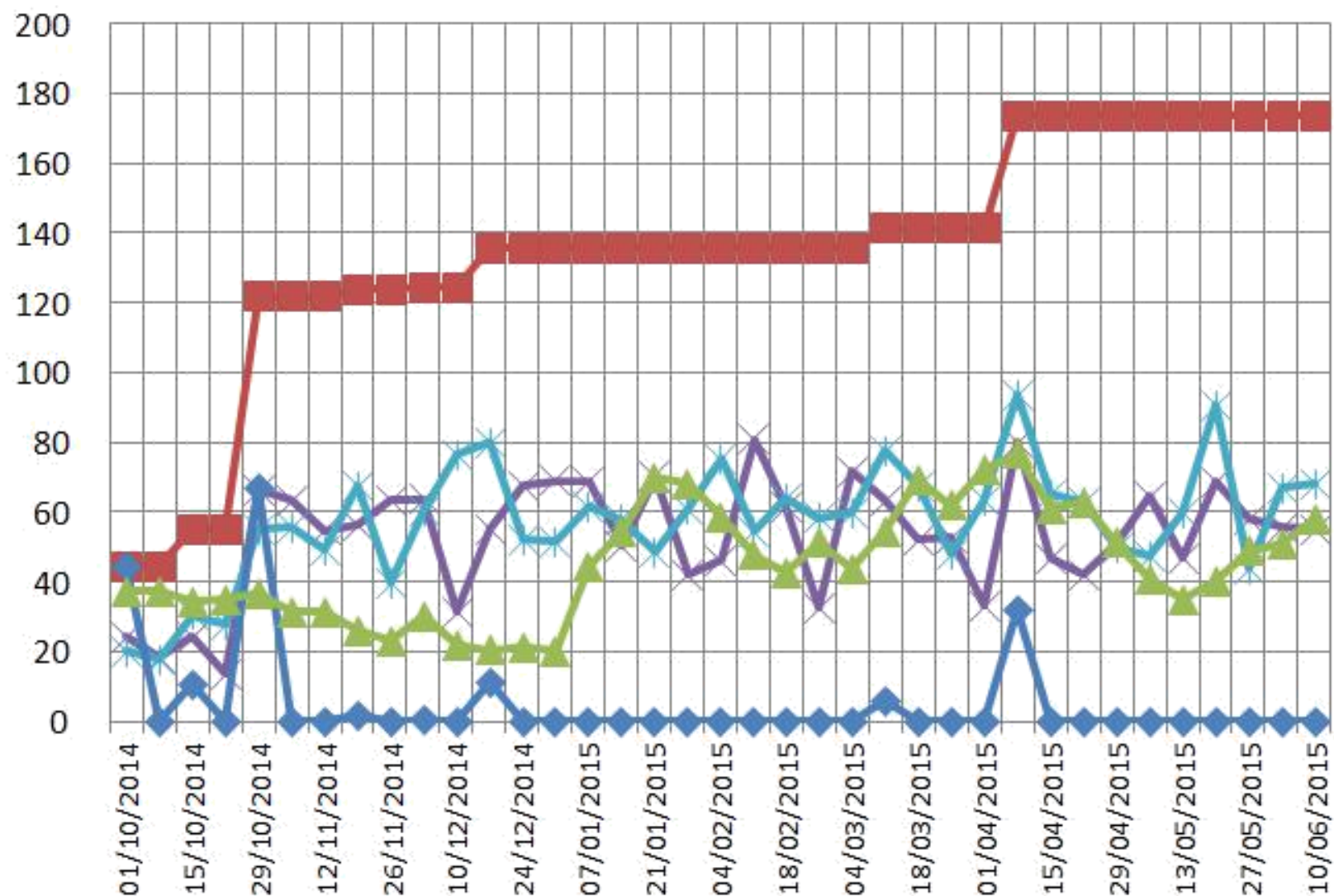
Defeitos - Por Criticidade

1 2 3 4 Defeitos Totais



Defeitos - Por Status

Def_Tot Def_Fech Def_Aber
Def_Novos Def_Rejeit

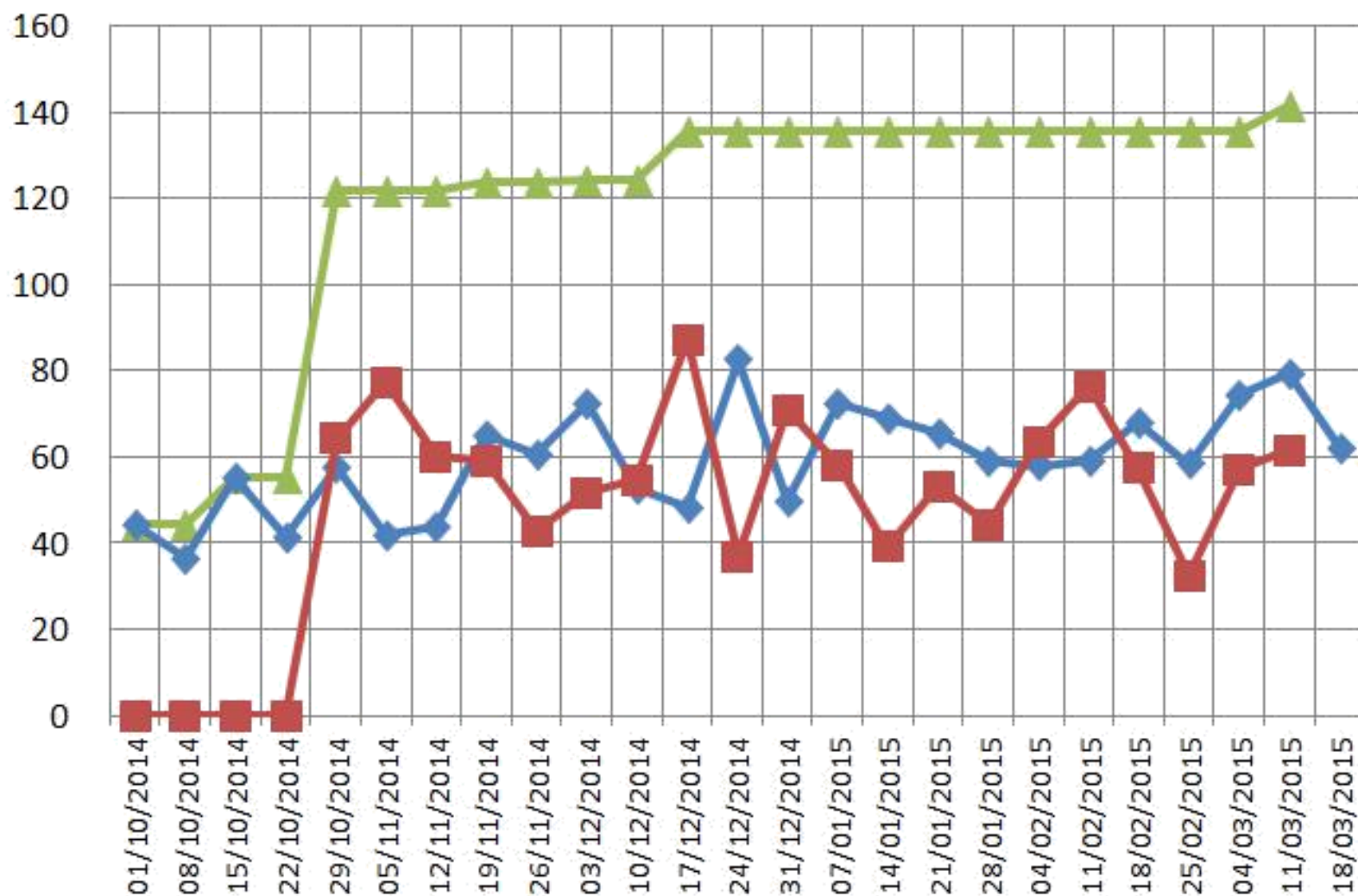


Defeitos - Por Origem

Def_Tot

Def_Int

Def_Ext



O que é automação de testes?

- Utilizar um sistema para testar outro
- Comparar resultados esperados e resultados reais
- Utilizar massa de dados
- Verificar resultados de cálculos

Por que automatizar?

- Reduzir esforço com testes
 - Escrever testes automatizados é difícil?
- Diminuir custo
 - Quanto custa o teste manual?
 - Quanto custa o teste automatizado?
- Aumentar a qualidade das entregas
- Garantir consistência na aplicação
 - Código não alterado parou de funcionar?

O que automatizar

- Teste Unitário
- Teste Funcional
- Teste de regressão
- Smoketest

Quando automatizar?

- Qual a duração do projeto?
 - 3 meses
 - 5 meses
 - 1 ano
- Produto?
- Relação Custo x Benefício

Automatizar é DIFÍCIL

- Criação dos scripts
- Execução dos scripts
- Manutenção dos scripts
 - Sistema muda os scripts tem que mudar

Vamos automatizar.....

- Prérequisitos:
 - Consolidado
 - Livre de defeitos
 - Fluxo crítico
- Priorize: Quando tudo é prioridade nada é prioridade....
 - Funcionalidade não depende de outro sistema
- Qual o nível de dependência
- Qual a estabilidade deste sistema
- É possível verificar o mesmo antes de iniciar os testes?

Lembre-se.....

- Automação de testes não é perfeita
- Ferramentas tem limitações
- Faz somente o que foi programada para fazer
- Erros encontrados precisam ser investigados
- A cobertura de automação nunca é 100%

Ferramentas

- Pagas
 - HP UnifiedFunctionalTest
 - Test Complete
 - SeeTest Automation
 - IBM RationalFunctionalTester
 - MonkeyTalk
- Free
 - Appium
 - Selenium

E o preço?

Buy TestComplete

A. REQUIRED¹ the TestComplete Platform

PLATFORM	NODE-LOCKED LICENSE	FLOATING USER LICENSE
TestComplete Platform	\$1,000 SHOP NOW	\$2,500 SHOP NOW

¹ TestComplete Platform provides a test IDE with foundational components for automated testing, recording, and integration with third-party tools. To run a TestComplete module, you must first install and license a TestComplete platform upon which the module can run.

B. REQUIRED²: at least 1 of the three modules: Web, Mobile, or Desktop

MODULE TYPE	NODE-LOCKED LICENSE	FLOATING USER LICENSE
TestComplete Web	\$999 ADD TO CART	\$1,998 ADD TO CART
TestComplete Mobile	\$1,299 ADD TO CART	\$2,598 ADD TO CART
TestComplete Desktop	\$999 ADD TO CART	\$1,998 ADD TO CART

² Technology modules extend the platform to enable testing of specific application technologies (Web, Mobile, and Windows Desktop and Client Server). To use any of the technology modules, you will need to have TestComplete platform installed and licensed. You can use multiple Modules (1 web + 1 mobile + 1 desktop) on one platform, which allows you to build complex system tests for multi-client, multi-tier applications with a database backend.

E o preço?

Mobile Add On for UFT

\$3500/ year

All models, All versions of SINGLE mobile OS
iPhone / Android / Blackberry / WindowsPhone

- UFT Recorder
- UFT Object Spy
- UFT Command Auto Completion
- UFT Reporting
- E-mail support (response time 72 hours)

BUY

Free
trial

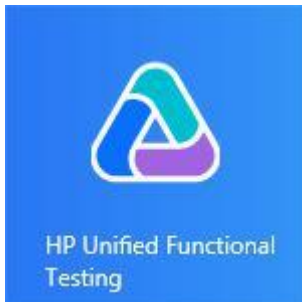
Language add on

500\$/ Year

- Per language
- List of languages

BUY

Vamos ver na prática



Certificações

- BSTQB CTFL
- CTFL é a sigla para **Certified Tester Foundation Level**. Trata-se da certificação de nível fundamental destinada a qualquer pessoa envolvida em testes de software. Isto inclui as pessoas em papéis como Testadores, Analistas de Testes, Engenheiros de Testes, Consultores de Teste, Gerentes de Teste, Aceitação e Desenvolvedores de Software.

CTAL

- CTAL é a sigla para **Certified Tester Advanced Level**. Trata-se da certificação de nível avançado do ISTQB destinada a pessoas que possuam experiência em suas carreiras de teste de software. Isto inclui pessoas em papéis como Testadores, Analistas de Testes, Engenheiros de Testes, Consultores de Teste, Gerentes de Teste, Aceitação do Usuário e Desenvolvedores de Software.

NIVEIS CTAL

- **CTAL-TM:** Advanced Level Test Manager
- **CTAL-TA:** Advanced Level Test Analyst
- **CTAL-TTA:** Advanced Level Technical Test Analyst

CTAL-TM

- **[TM01]** Gerenciar um projeto de teste, implementando a missão, objetivos e processos de testes estabelecidos para a organização de testes.
- **[TM02]** Organizar e levar a identificação de riscos e sessões de análise de risco e utilizar os resultados de tais sessões para o teste de avaliação, planejamento, monitoramento e controle.
- **[TM03]** Criar e implementar planos de teste consistentes com as políticas e estratégias organizacionais.
- **[TM04]** Monitorar e controlar continuamente as atividades de teste para alcançar os objetivos do projeto de teste.
- **[TM05]** Avaliar e relatar o status de teste relevante e oportunamente para os participantes do projeto.
- **[TM06]** Identificar as habilidades e as lacunas de recursos em sua equipe de testes e participar no fornecimento de recursos adequados.
- **[TM07]** Identificar e planejar o desenvolvimento de competências necessárias dentro de sua equipe de teste.
- **[TM08]** Propor um caso de negócios para as atividades de teste que define os custos e os benefícios esperados.
- **[TM09]** Garantir uma comunicação adequada dentro da equipe de teste e com outras partes interessadas no projeto.
- **[TM10]** Participar de iniciativas de melhoria de processos de teste.

CTAL-TA

- **[TA01]** Executar as atividades de teste apropriadas com base no ciclo de vida de desenvolvimento de software específico a ser utilizado.
- **[TA02]** Determinar a priorização adequada das atividades de teste com base nas informações fornecidas pela análise de risco.
- **[TA03]** Selecionar e aplicar técnicas de teste adequadas para garantir um nível adequado de confiança, com base em critérios de cobertura definidos.
- **[TA04]** Fornecer o nível apropriado de documentação relevante para as atividades de teste.
- **[TA05]** Determinar os tipos apropriados de testes funcionais a serem executados.
- **[TA06]** Assumir a responsabilidade para o teste de usabilidade para um determinado projeto.
- **[TA07]** Participar efetivamente das avaliações formais e informais com as partes interessadas, aplicar os conhecimentos de erros típicos feitos em produtos de trabalho.
- **[TA08]** Modelar e implementar um esquema de classificação de defeitos.
- **[TA09]** Aplicar ferramentas para suportar um processo de teste eficiente.

CTAL-TTA

- **[TTA1]** Reconhecer e classificar os riscos típicos associados com desempenho, segurança, confiabilidade, portabilidade e manutenção de sistemas de software.
- **[TTA2]** Criar planos de teste que detalham o projeto, o planejamento e a execução de testes de desempenho para mitigação de riscos de segurança, confiabilidade, portabilidade e manutenção.
- **[TTA3]** Selecionar e aplicar técnicas adequadas de projeto estrutural para garantir que os testes forneçam um nível adequado de confiança baseada na cobertura de código e cobertura de modelagem.
- **[TTA4]** Participar efetivamente das revisões técnicas com os desenvolvedores e arquitetos de software, aplicando os conhecimentos obtidos de erros típicos no código e modelagem.
- **[TTA5]** Reconhecer os riscos na arquitetura do código de software e criar elementos do plano de teste para mitigar esses riscos através de análise dinâmica.
- **[TTA6]** Propor e implementar medidas para melhorar a segurança, manutenção e testabilidade do código através da aplicação de análise estática.
- **[TTA7]** Delinear os custos e os benefícios que podem ser esperados da introdução de determinados tipos de automação de teste.
- **[TTA8]** Selecionar as ferramentas adequadas para automatizar tarefas de testes.
- **[TTA9]** Entender as questões técnicas e conceitos na aplicação de automação de teste.

CBTS - Certificação Brasileira de Teste de Software

- A Associação Latino Americana de Teste de Software (ALATS), criou a Certificação Brasileira de Teste de Software (CBTS) para atender a uma exigência do mercado brasileiro na Área de Qualidade de Software.
- O objetivo da CBTS é estabelecer um padrão em conformidade com os requisitos para uma avaliação da qualificação dos profissionais na área de Qualidade de Software. Adquirir o certificado CBTS para o profissional da área é um grande diferencial, pois indica que o mesmo possui um excelente nível de competência profissional nos princípios e nas práticas de Qualidade de Software, dentre os demais profissionais de TI.

Perfil

- O perfil dos candidatos ao exame do CBTS se estende aos profissionais que desejam obter um reconhecimento técnico para o mercado brasileiro na Área de Qualidade de Software.
- Isto inclui desde à Diretores de TI, Gerentes de Projetos, Gerentes de Qualidade, Gerentes de Desenvolvimento de Software, Analistas de Sistemas, bem como todos os profissionais já envolvidos diretamente na área de Qualidade de Software como os Testadores, Líder de Teste, Gerente de Teste, Auditor de Qualidade de Software, Arquiteto de Teste e Analista de Teste.

Recertificação

- A certificação é válida no ano da prova (a partir da data divulgada da aprovação) e nos 3 anos seguintes completos, terminando sempre em 31/12.
- O período de aquisição dos PDTS (Ponto de Teste de Software) é correspondente ao período de validade da certificação, ou seja no decorrer dos três anos. 50 PDTS devem ser obtidos neste período para que não seja necessário prestar novo exame.
- A partir do primeiro período, a cada 3 anos completos, novos 50 PDTS devem ser adquiridos, ou seja o certificado poderá ser validado sem a necessidade de outro exame.

Pontos

Item	Atividade	PDTS	Total de PDTS no período
A	Treinamento relacionado a teste de software (necessita comprovação do provedor)	1 PDTS por cada hora do curso de teste de software	40 PDTS
B	Participar do Seminário do BRATESTE (2 dias)	15 PDTS	45 PDTS
C	Participar do Tutorial do BRATESTE	12 PDTS para cada dia de tutorial	40 PDTS
D	Autor ou coautor de artigo sobre teste de software publicado em revista técnica de software	Autor: 10 PDTS por artigo Coautor: 5 PDTS por artigo	20 PDTS
E	Autor ou coautor de artigo sobre teste de software publicado em sites e blogs	Autor: 5 PDTS por artigo Coautor: 3 PDTS por artigo	10 PDTS
F	Autor ou coautor de livro sobre teste de software (com ISBN)	Autor: 30 PDTS por livro Coautor: 20 PDTS por livro	40 PDTS
G	Ser palestrante em conferências, seminários, congressos sobre Teste de Software ou em palestras promovidas pela ALATS	5 PDTS por palestra	20 PDTS
H	Atestado da empresa, comprovando que o profissional trabalha com atividades relacionadas à disciplina de teste de software	2 PDTS por período 30 dias	30 PDTS

Referências

- Base de Conhecimento Em Teste de Software - 3ª Ed. 2012; Rios, Emerson; Bastos, Aderson; Cristalli, Ricardo; Editora Martins
- The Software Test Engineer's Handbook: A study guide for the ISTQB Test Analyst and Technical Test Analyst Advanced Level Certificates Graham Bath, Judy McKay
- *Software Testing Foundations: A Study Guide for the Certified Tester Exam* Andreas Spillner, Tilo Linz, Hans Schaefer
- Mantis - www.mantisbt.org
- Testlink - testlink.org
- Selenium - seleniumhq.org