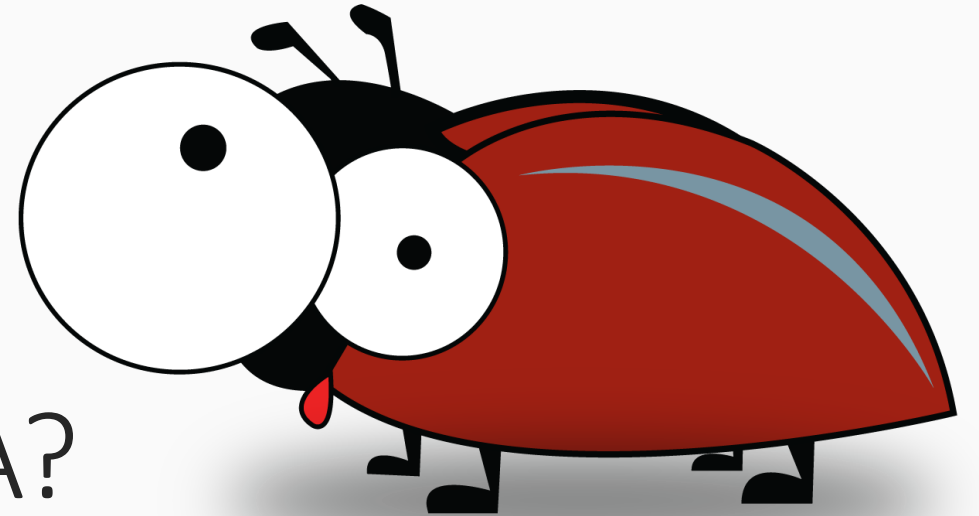


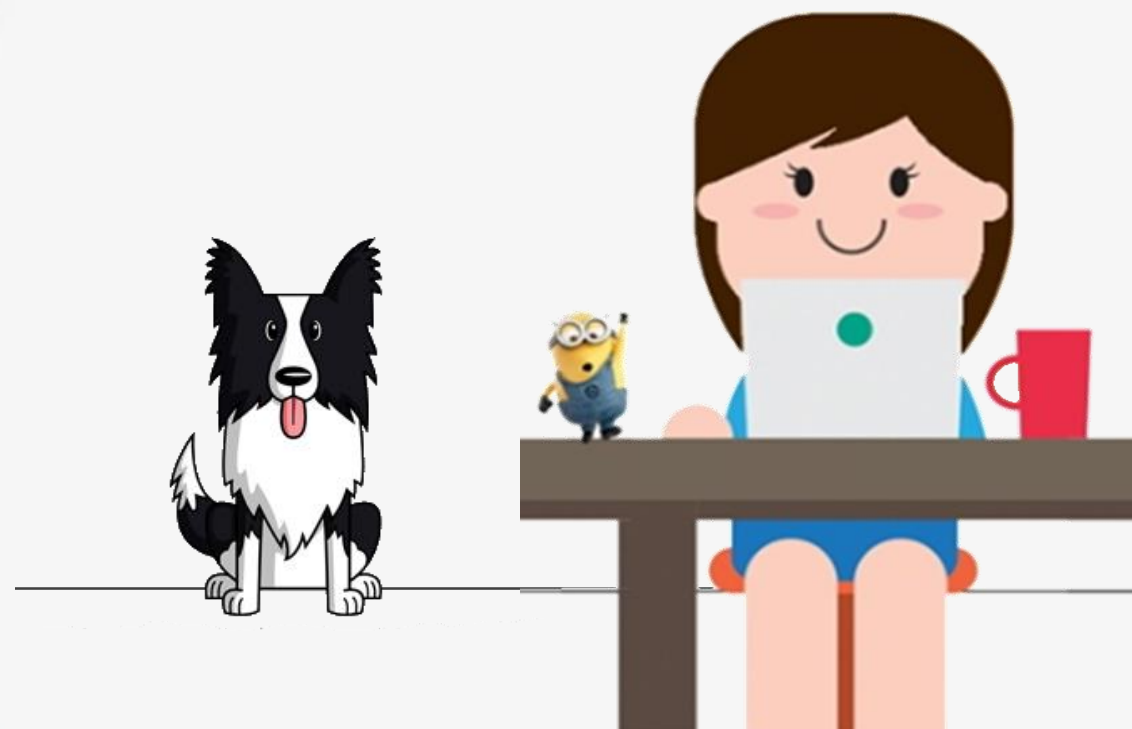
VIXE UM BUG... E AGORA?

BY JULIANA MARTINS



Sobre mim

- Professora
 - Gerente de Projetos e QA Lead na OSF-Global
 - 15 anos em TI
 - CTAL-TA, CTFL, PSM I, Adobe Practitioner
 - Membro GTS-CE (Grupo de Teste do CE)
 - Membro PHP com Rapadura
-
- Doutoranda em Informática Aplicada
 - Pesquisadora em RV para reabilitação infantil
 - Mestra em Informática Aplicada
 - Tecnóloga em ADS



✉ jullianamartins@gmail.com

🐦 @jullianamar

A DESCOBERTA DE UM
BUG TEM FASES...



1ª FASE

SURPRESA



2ª FASE

NEGAÇÃO



3ª FASE

PROCURA

UM

CULPADO

É CACHE

Não é bug... é
uma feature

Então confessa... O que
você digitou de errado
para travar?

Você deve estar rodando
a versão errada.

O bug é de quem acha

É apenas uma coincidente falta de sorte.

Isto funcionava
bem até ontem

NA MINHA MÁQUINA FUNCIONA

Tudo bem, mesmo que não
funcione, aparentemente
funciona?

Esta versão não é
compatível com seu
computador.

Alguém deve ter alterado meu código.

Isso nunca aconteceu antes.

Por que você fez isso?

Onde você estava quando o
programa estava
funcionando?

4ª FASE

ARRANJA

UMA

DESCULPA



5ª FASE

RAIVA

DE QUEM DESCOBRIU O BUG



6ª FASE

CORRIGE

O

BUG



7ª FASE

EXPECTATIVA



8ª FASE

ALÍVIO

ENVIA PRO CLIENTE



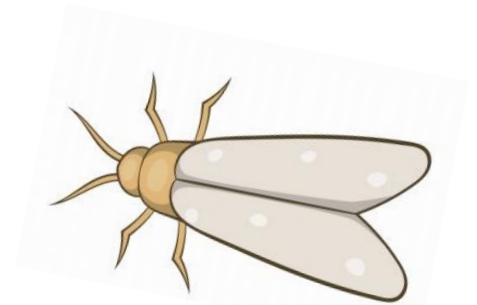




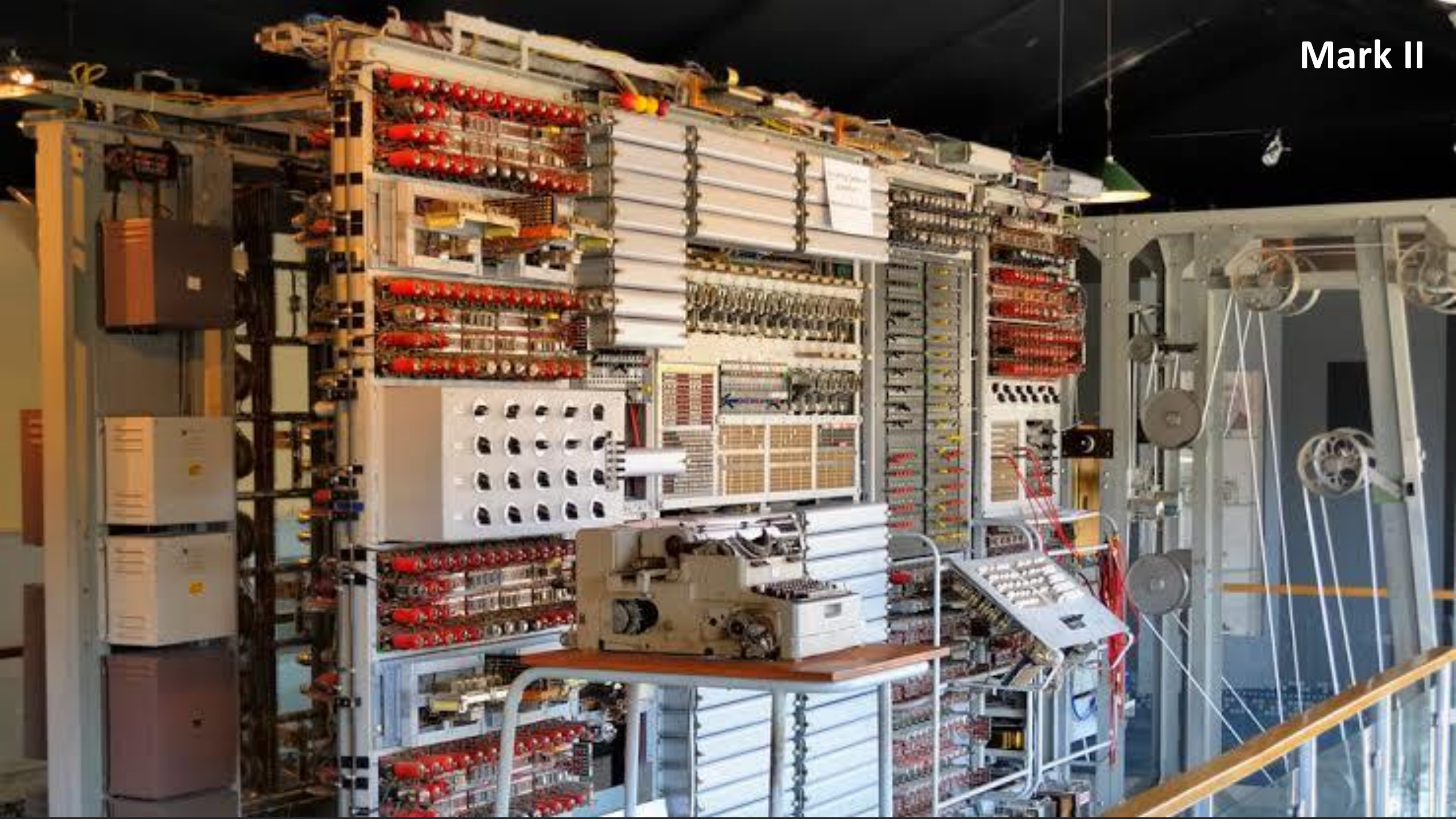
Origem do termo bug

Em **9 de setembro de 1947** o operador **William Burke** encontrou dentro de um Mark II (da marinha dos EUA) uma mariposa que estava presa entre os fios, causando uma falha no funcionamento do computador.

Ele relatou em seu diário de bordo um caso real de **"inseto (bug) sendo encontrado"**.



Mark II

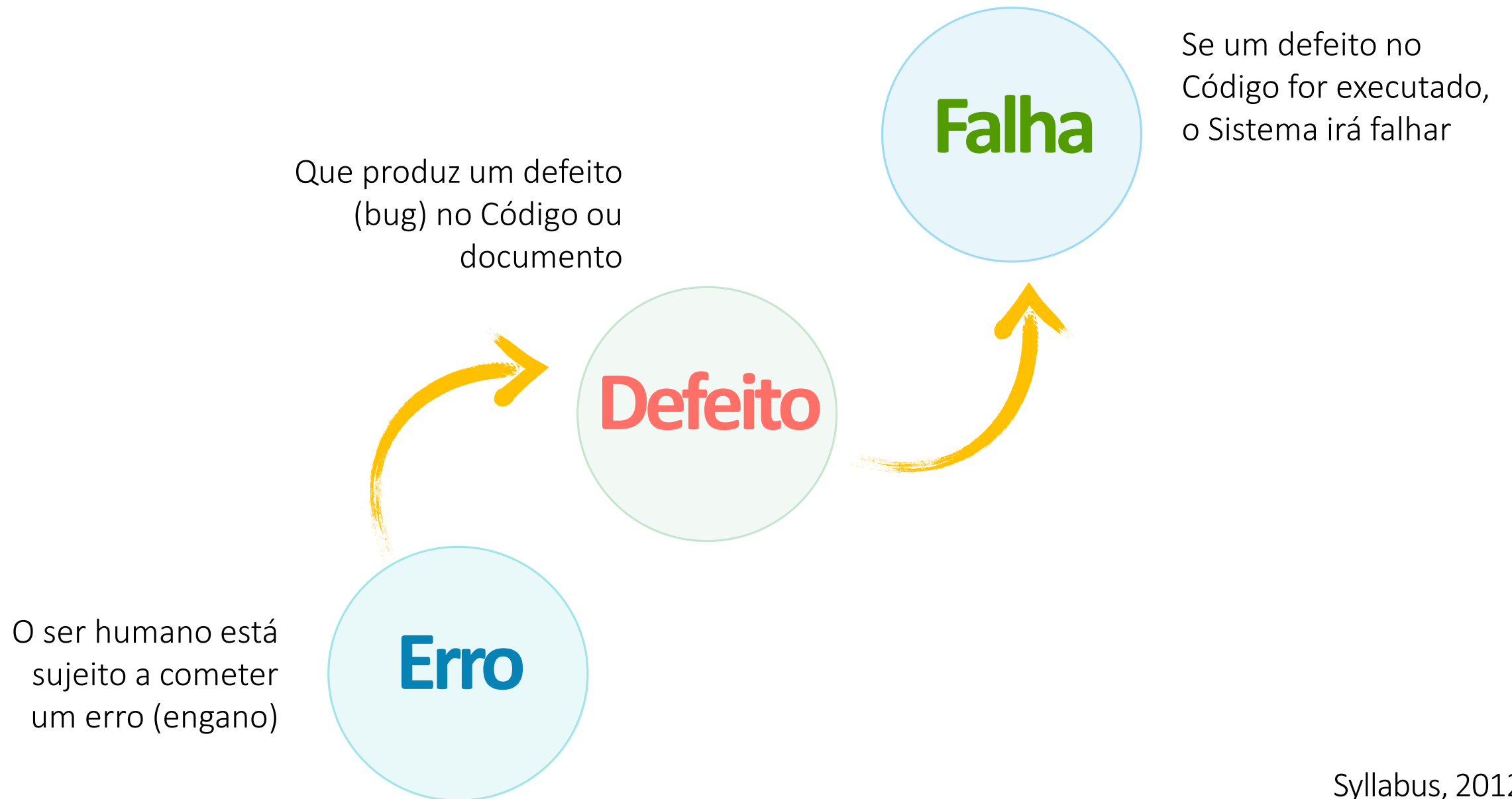


Por que há mais
de 2000 bugs
nessa versão?



Um bom software
sempre tem bugs!





Origem das Falhas

Falhas de fluxo de dados

Problemas de gerenciamento
(Tempo, escopo, custo)

Falha nos algoritmos

Falha de controle, lógica
ou seqüência

Enganos ou esquecimentos

Falha no detalhamento
das funcionalidades

Falha na descrição das
interfaces

Falta de entendimento do
requisito

Requisitos descritos incorretamente ou
sem informações suficientes

Falha na interface externa

Problemas de comunicação

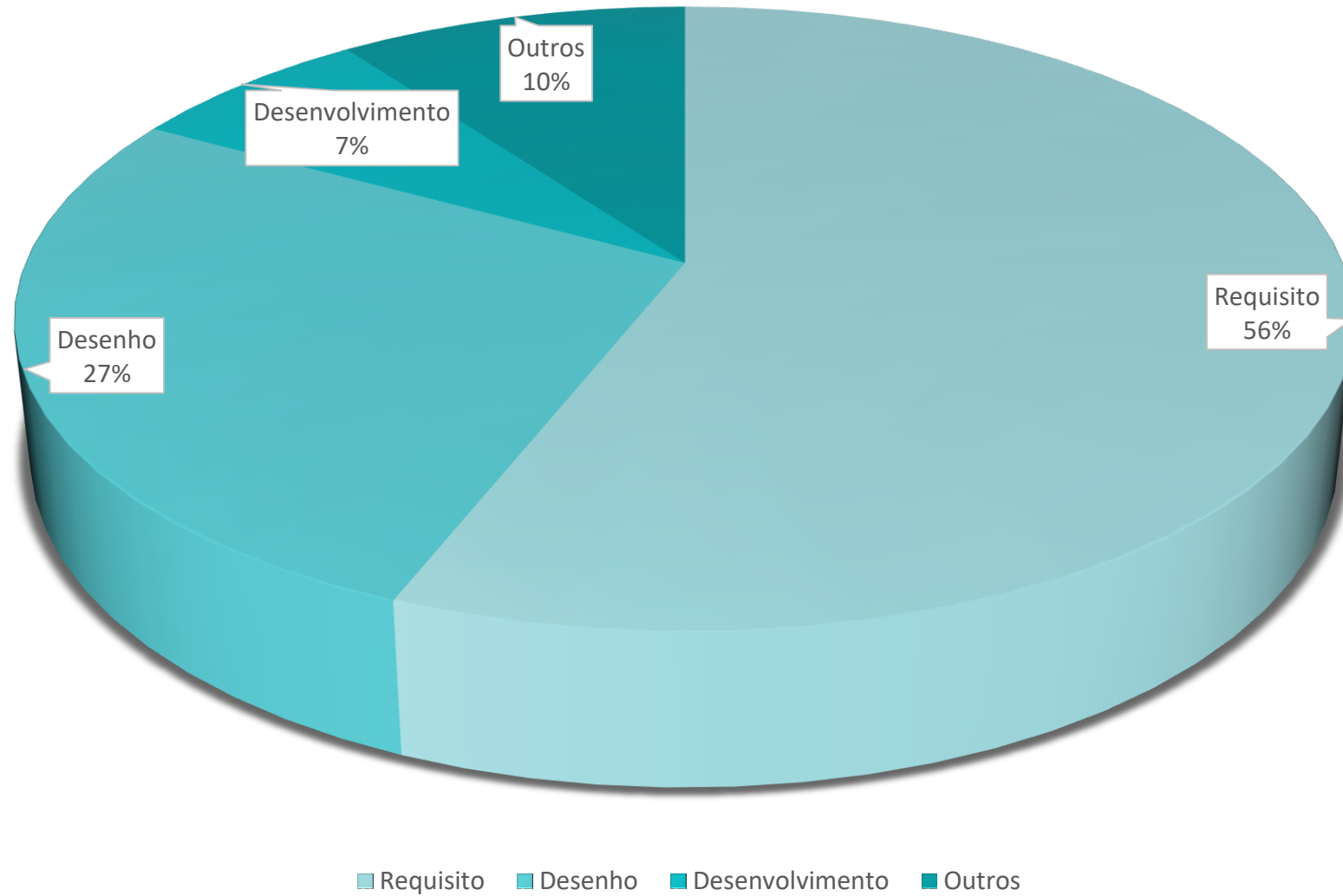
Falha nas estruturas de dados

Recursos não previstos implementados no software

Implementações incompletas

Ausência do critério
de aceitação

Distribuição



Severidade x Prioridade

Severidade

| | | |
|-----------------------------|---|---|
| S1 - Crítica / Show Stopper | Bloqueia o teste de uma função ou feature causa crash na aplicação nos principais casos de uso | <ul style="list-style-type: none">• Botão não funciona, impedindo o uso da feature• Segurança• Feature esperada não aparece na build• Perda grave de dados |
| | Bloqueia release | |
| S2 - Grave | Feature funciona pobremente | <ul style="list-style-type: none">• Input esperados causam crash ou efeitos indesejados• Input incomum causa efeitos irreversíveis |
| S3 - Moderada | Feature não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada | <ul style="list-style-type: none">• Mensagem de erro ou sucesso não é exibida• Input incomum causa efeitos indesejados, mas contornáveis |
| S4 - Pequena | Quase nenhum impacto na funcionalidade, mas ainda é um erro válido | <ul style="list-style-type: none">• Erro ortográfico• Pequenos erros de UI |

Prioridade

| | | |
|--------------|--|--|
| P1 - Crítico | Tem que ser solucionado imediatamente | <ul style="list-style-type: none">• Erros de severidade S1 encontrados produção• Erros de UI que afetam feeling/UX |
| P2 - Alta | Feature não é usável como deveria ser, por problema no código | <ul style="list-style-type: none">• Configuração de dependências• Detalhes da feature não levantados durante seu desenvolvimento• Deadlines do projeto |
| P3 - Média | Problemas que podem ser avaliados para entrar no ciclo de desenvolvimento no futuro; dependendo dos recursos disponíveis | <ul style="list-style-type: none">• Efeitos indesejados quando o sistema é submetido a inputs incomuns ou não-existent no ambiente do usuário |
| P4 - Baixa | Erros que não afetam funcionalidade | <ul style="list-style-type: none">• Pequenas melhorias de UI e UX• Erros de digitação |

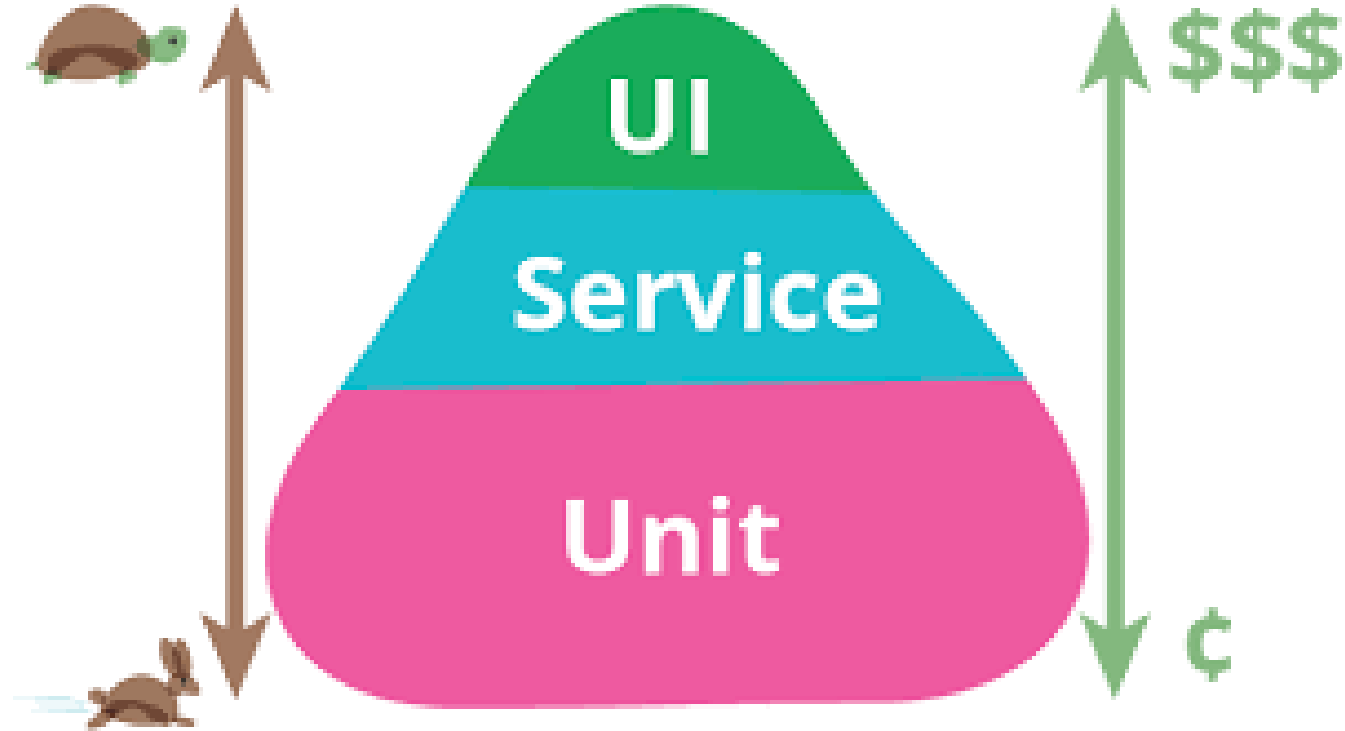
Severidade x Prioridade

A severidade de um defeito define o impacto do defeito no funcionamento da aplicação.

A prioridade indica a ordem de correção do defeito (defeitos com alta prioridade são corrigidos imediatamente ou num curto prazo de tempo).

| PRIORIDADE | | | | |
|------------|---------------------------|------------|----------|--------|
| | | | | |
| | | | | |
| | | | | |
| | Não consegue contornar | 2 | 2 | 1 |
| | Existem alternativas | 3 | 3 | 2 |
| | Consegue Realizar Tarefas | 4 | 3 | 3 |
| | | Raramente | As vezes | Sempre |
| | | SEVERIDADE | | |

Custo



Como prevenir



Defina com clareza o escopo do projeto



Planeje uma resposta adequada e monitore os riscos



Tenha um processo bem definido



Identifique os riscos o mais breve possível



Seja transparente



Documente as decisões



Identifique o sentido de Pronto



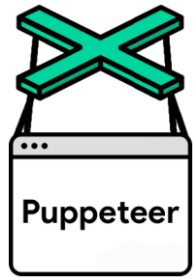
Realize testes unitários, integração e funcionais



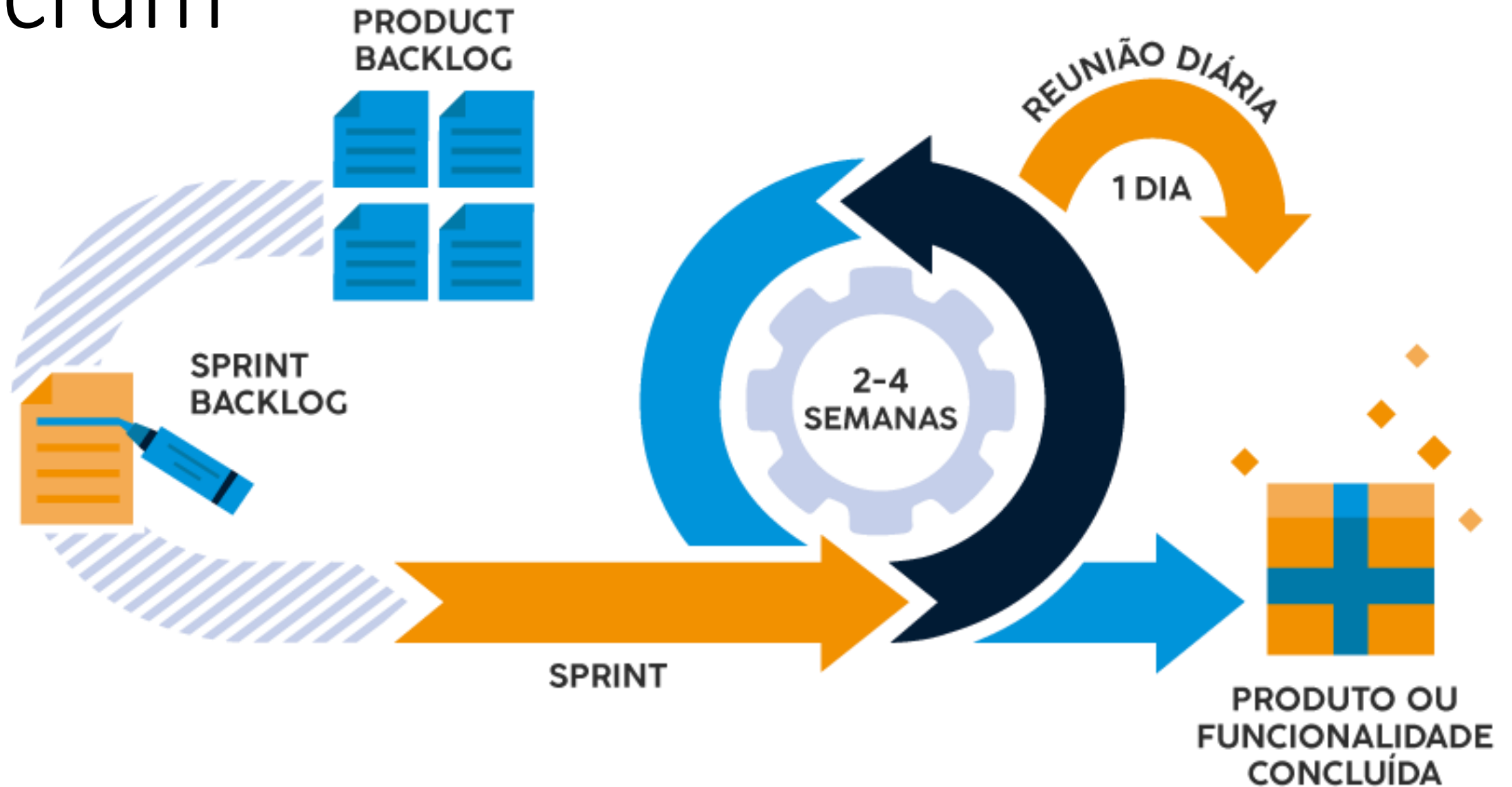
Realize automatização de testes



Seja Criterioso



Scrum



Valores



BUGS HAVE FEELINGS TOO

IF YOU FIND A BUG:
REPORT IT

BUGS DON'T LIKE
TO BE FORGOTTEN



IF YOU FIND A BUG:
GET TO KNOW THEM

BUGS LIKE TO BE
UNDERSTOOD



This ladybird
has 3 spots

IF YOU FIND A BUG:
TAKE A PHOTO

BUGS LIKE TO KEEP MEMORIES
OF THE OCCASION



IF YOU FIND A BUG:
GET TO KNOW THEIR MATES

BUGS ARE SOCIALITES



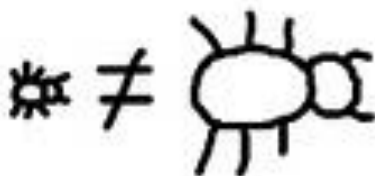
IF YOU FIND A BUG:
REPORT IT QUICK

OTHERWISE BUGS SETTLE IN AND
MAKE A HOME FOR THEM SELVES



IF YOU FIND A BUG:
BE HONEST

BUGS DON'T LIKE
GOSSIPS



IF YOU FIND A BUG:
NOTE HOW YOU
MEET THEM

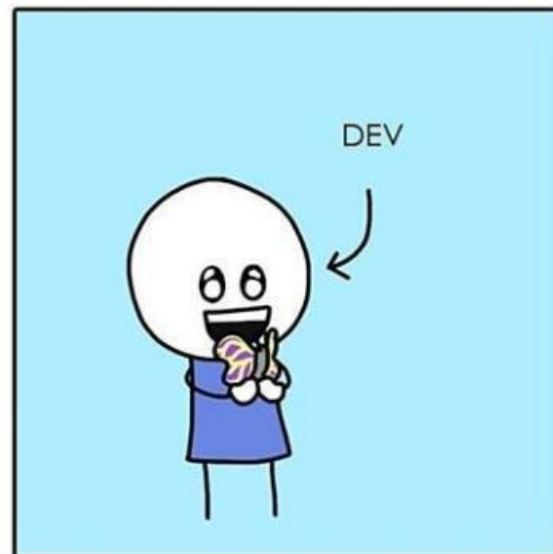
BUGS ARE ROMANTICS



IF YOU FIND A BUG:
DON'T IGNORE IT

BUGS CAN BITE IF
NOT APPRECIATED





“Testers don’t like to break things; they like to dispel the illusion that things work.”

“Os testadores não gostam de quebrar coisas; eles gostam de dissipar a ilusão de que as coisas funcionam. ”

([Kaner, Bach, Pettichord](#))

