

Desafios sobre Qualidade de Software

Como aumentar a qualidade de software?

Why?



Serasa ser reconhecida como
referência em padrões de
excelência



How?



Identificando pontos fortes e fracos



Problemas a serem solucionados

- Não possuímos padronização de processos e técnicas
- Não adotamos as melhores práticas de desenvolvimento
- Não temos integração contínua e entrega contínua
- Times possuem níveis de maturidade e conhecimentos técnicos discrepantes
- Não temos mapeadas todas as integrações e seus possíveis impactos
- Como garantir que a alteração do produto A da equipe 1 não impactou no produto B da equipe 2 ?
- Cada time tem sua prioridade e cadência

What?



O que precisamos fazer para alcançar padrões de excelência



Secure Software Development Life Cycle

- Definir e implantar um processo de desenvolvimento
- Definir modelo claro de atuação de papéis (Dev, QA, PO, etc...)
- Definir métricas
- Definir e adotar uma estratégia de ambientes e tipos de testes
- Definir e adotar convenções de GIT
- Implantar padronização de processos de CI/CD
- Quality Gates
- Análise de vulnerabilidades
- Implantar Release Process

Resultados que serão obtidos com as mudanças

- Padronização de técnicas e processos de desenvolvimento
- Coleta e análise de dados para melhoria contínua
- Aumentar velocidade e qualidade das entregas
- Integração e entrega contínua
- Processos escaláveis
- Diminuição de bugs em produção
- Cultura de qualidade de software

Modelo de Maturidade de Qualidade

Os indicadores abaixo serão usados para atestar o nível de maturidade das aplicações

- Inicial
- Gerenciado
- Definido
- Maduro
- Otimizado

Inicial

- Processo estruturado de post mortems e acompanhamento das issues geradas

Gerenciado

- Cobertura mínima de 20% nos testes automatizados
- Existem alguns testes funcionais mas não são executados de forma automática
- Realização de revisão de código durante o ciclo de desenvolvimento
- Alguma documentação disponível sobre a aplicação como README, Confluence, etc..

Definido

- Cobertura mínima de 50% nos testes automatizados
- Definição das métricas de qualidade de Código
- Acompanhamento das métricas definidas
- Testes funcionais automatizados como parte do processo de release

Maduro

- Cobertura mínima de 70% nos testes automatizados
- Implementação de diferentes tipos de testes: integração, regressão, E2E, sendo executados no fluxo de CI
- Sonar como Quality Gate das mudanças bloqueando alterações que não atinjam o padrão estabelecido
- Execução da validação de segurança como parte do fluxo de CI/CD

Otimizado

- Cobertura mínima de 90% nos testes automatizados
- Execução de testes de performance como parte do processo de CI/CD

Quais são os próximos passos?

Times avaliam o grau de criticidade das aplicações para a continuidade do negócio

- **Baixo**
- **Moderado**
- **Alto**
- **Crítico**

Aplicações categorizadas como Alto e Crítico,
devem ser compatíveis como os modelos
Maduro e Otimizado.

Após avaliação e categorização, os times criam o plano de ação com as ações e prazos para adequação

OBRIGADO!

Anderson Mann:
anderson.mann@br.experian.com



- **Links dos materiais**



- Software Development Life Cycle
- Modelo de maturidade

