Problemas, qualidade e processo de desenvolvimento de software

Um sistema entrou em produção. O que aconteceu?

Contexto

Uma empresa de software implantou um sistema de agendamento em uma clínica médica. Após o início da operação, a equipe recebeu um registro de ocorrências dos usuários.

Registro de ocorrências

1. Dois procedimentos que dependem do **mesmo equipamento** puderam ser agendados simultaneamente.
2. Profissionais relatam **dificuldade para localizar** uma operação usada com frequência.
3. Uma regra importante da rotina somente foi esclarecida quando **grande parte do sistema já estava desenvolvida**.
4. A alteração exigiu **modificar partes já produzidas** e repetir testes.

Atividade

Analise o registro ao lado e identifique:

- duas **possíveis causas** dos problemas;
- duas **consequências** para o projeto ou para o uso do sistema.

Não há resposta única. O objetivo é analisar e formular hipóteses antes de avançar.

O problema começou onde?

Compartilhe as hipóteses produzidas com a turma. Use as perguntas abaixo para orientar sua justificativa:

- **O problema surgiu somente na programação?**
- **Que informação relevante pode ter faltado?**
- **Quem precisava participar dessa descoberta?**
- **O que precisou ser refeito depois?**

Uma possível solução

Os problemas observados em produção são consequências de requisitos e necessidades dos usuários identificados ou validados tardiamente. Revisando as questões anteriores, a equipe poderia ter evitado o retrabalho.

1

O problema surgiu somente na programação?

Não. Os registros indicam falhas na descoberta, definição e validação dos requisitos antes e durante o desenvolvimento.

2

Que informação relevante pode ter faltado?

A regra de que determinados procedimentos utilizam o mesmo equipamento e não podem ocorrer simultaneamente, além de informações sobre as operações mais frequentes na rotina dos profissionais.

3

Quem precisava participar dessa descoberta?

Profissionais da clínica (responsáveis pelo agendamento e pelos equipamentos), junto à equipe de produto, design e desenvolvimento.

4

O que precisou ser refeito depois?

Partes do sistema tiveram de ser alteradas, com repetição de testes e validações, gerando retrabalho, aumento de prazo e custo, além de problemas de uso após a implantação.

Falhas podem começar antes do código

Comunicação insuficiente

A equipe precisa interpretar as necessidades do cliente antes de implementá-las. Essa interpretação pode estar incompleta ou equivocada desde o início.

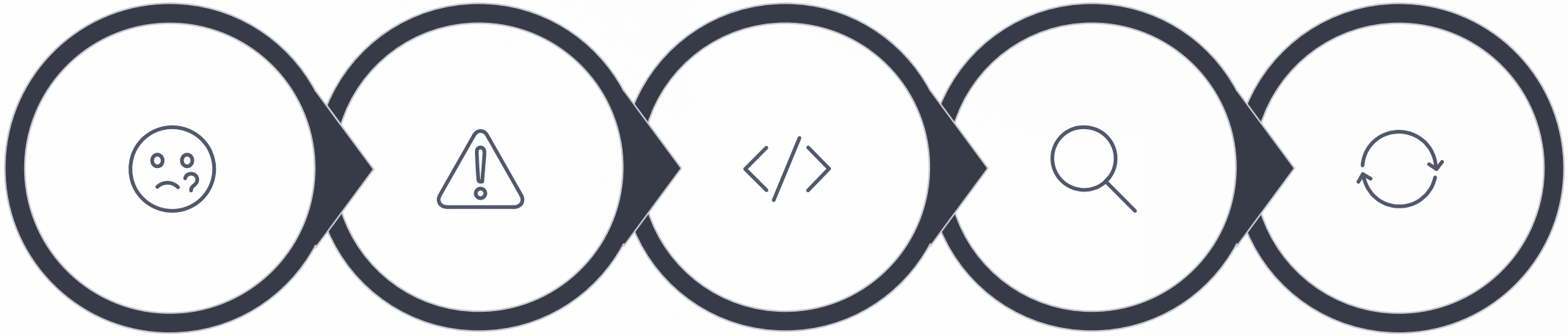
Requisitos insuficientes ou ambíguos

Informações ausentes, vagas ou contraditórias orientam decisões diferentes daquelas esperadas pelos usuários — sem que a equipe perceba.

Mudanças de necessidade

Mudanças podem ser legítimas. O problema não é a mudança em si, mas não reconhecer e analisar seu impacto no que já foi produzido.

Como um problema se propaga



Uma informação incorreta ou incompleta não fica restrita aos documentos — ela pode atingir modelos de dados, código e testes antes de ser detectada.

Funcionar não é o mesmo que atender bem

Funcionar conforme implementado

O software executa aquilo que foi programado. Os testes passam. O sistema não apresenta erros técnicos visíveis.

Atender adequadamente

O comportamento corresponde às necessidades e às condições reais de utilização. O usuário consegue realizar suas tarefas de forma eficaz.

Nesta aula, consideramos especialmente três dimensões da qualidade de software:

Confiabilidade

Comportamento consistente e resultados corretos nas condições previstas de uso.

Usabilidade

Condições para que o usuário compreenda e realize suas tarefas com a menor dificuldade possível.

Adequação ao usuário

Correspondência entre a solução construída e a necessidade que justificou o sistema.

Discussão: Qual dessas dimensões aparece com mais força em cada ocorrência do caso da clínica? Classifique oralmente uma ocorrência e justifique.

Quebra-gelo

Para começar nossa atividade de hoje, vamos descontrair e aquecer as ideias!

Agora você representa o processo

Papel e contexto

Você integra a equipe responsável por documentar como o desenvolvimento deve ser organizado. A empresa precisa tornar o processo mais compreensível após os problemas encontrados no sistema da clínica.

Desafio

Construa um **fluxograma simplificado** do processo de desenvolvimento de software.

Entrega – o fluxo deve conter:

- as principais etapas;
- conexões entre elas;
- uma explicação breve da função de cada etapa;
- pelo menos uma representação de documentação, comunicação, participação do cliente ou retorno entre atividades.

O que o fluxograma precisa comunicar?

Utilize os critérios abaixo como referência durante a produção do seu fluxograma:

1. **As etapas essenciais estão presentes?**
2. **As conexões formam um processo compreensível?**
3. **A função de cada etapa está explicada brevemente?**
4. **Comunicação, documentação e participação do cliente aparecem em mais de um momento?**
5. **Existe retorno quando uma nova informação exige revisão?**

Não é necessário representar um modelo específico de desenvolvimento — como Cascata, Scrum ou qualquer outro. O objetivo é uma representação conceitual justificada.

Revise como uma equipe de desenvolvimento

Troque o fluxograma com um colega. Analise o artefato recebido com base nas perguntas abaixo:

- **Existe alguma etapa importante ausente?**
- **Alguma conexão gera interpretação incorreta?**
- **As funções de cada etapa estão compreensíveis?**
- **O fluxo parece rígido demais – sem possibilidade de retorno?**
- **Comunicação, documentação ou participação do cliente aparecem no fluxo?**
- **Uma mudança de necessidade poderia ser incorporada ao fluxo representado?**

Ao final: indique uma melhoria concreta ao fluxo do colega.

Principais atividades do processo de desenvolvimento

01

Identificação da necessidade

Compreender o problema, a oportunidade e o contexto de uso.

02

Levantamento de requisitos

Identificar necessidades, funcionalidades, restrições e expectativas dos usuários.

03

Análise e planejamento

Definir a solução, prioridades, recursos, arquitetura e forma de execução.

04

Desenvolvimento

Projetar e implementar o software.

05

Testes e verificação

Avaliar funcionamento, qualidade e aderência aos requisitos.

06

Implantação

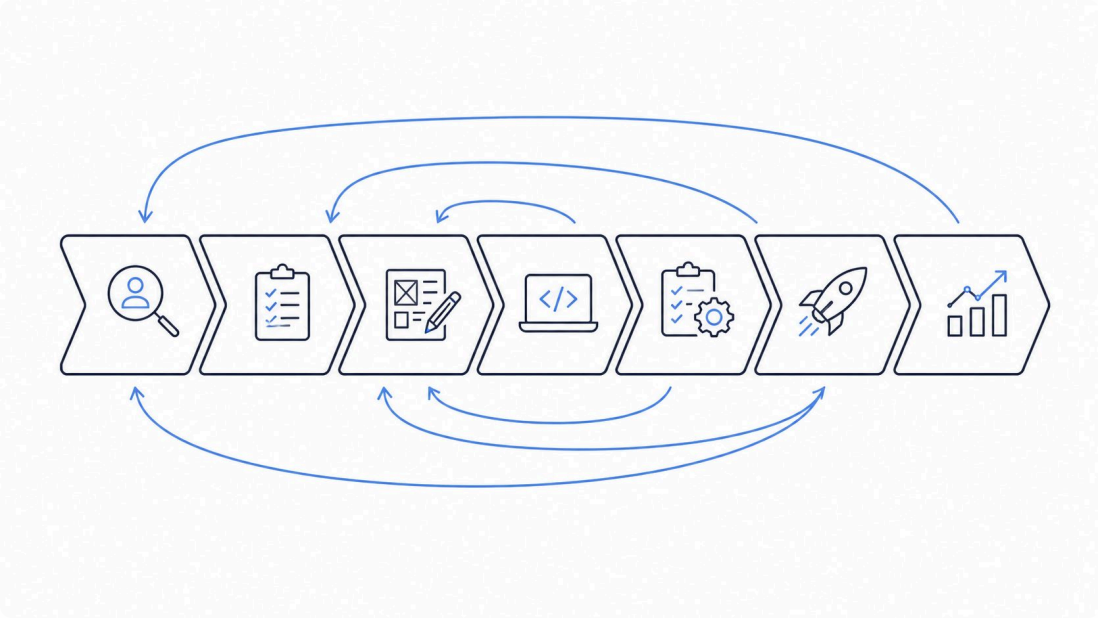
Disponibilizar a solução para uso em ambiente real.

07

Operação e evolução

Acompanhar o uso, corrigir problemas e realizar melhorias.

O processo também precisa permitir retornos



Mudança não significa necessariamente erro. Necessidades podem mudar legitimamente. O problema é não compreender e controlar o impacto dessa mudança.

Documentação, comunicação, acompanhamento e participação do cliente **não pertencem a uma única etapa**. Eles ocorrem ao longo de todo o processo e ajudam a:

- registrar decisões e confirmar interpretações;
- acompanhar mudanças e compreender seus impactos;
- verificar se a solução continua coerente com a necessidade.

Em que o processo nos ajuda?

Compreender antes de construir

Falhas de comunicação e requisitos frágeis podem se propagar do levantamento até o código e os testes — silenciosamente.

Qualidade é adequação

Funcionar tecnicamente não garante que o sistema atenda ao usuário. Confiabilidade, usabilidade e adequação precisam ser consideradas desde o início.

Descobertas tardias aumentam o impacto

Decisões já incorporadas ao sistema precisam ser revistas quando a informação chega tarde — gerando retrabalho em múltiplas etapas.

Processo organiza a evolução

Atividades, documentação, comunicação e participação dos usuários ajudam a compreender e controlar mudanças — sem transformá-las em crise.

Obrigado e até a próxima!

Uma ótima volta para casa!