

Atividade Prática – Gerenciamento de Riscos em Projetos de Software

Integrantes : André S., Matheus M., Ruan L., Rayza, Thiago Lazari.

Objetivo

Identificar, analisar e planejar respostas aos riscos em um projeto de desenvolvimento de software.

Exercício 1 – Identificação de Riscos

Liste pelo menos 10 riscos que podem afetar negativamente o projeto considerando aspectos técnicos, humanos, prazos, integração etc. Descreva a causa e a consequência de cada risco.

Exemplo:

Risco: Falta de tempo para concluir os testes.

Causa: Atividades paralelas dos integrantes.

Consequência: Entrega de uma versão com bugs graves.

Risco 1: Falta de domínio em Django

Causa: Pouca experiência da equipe com o framework.

Consequência: Atrasos e retrabalho por decisões técnicas equivocadas.

Risco 2: Integração fraca de front-end e back-end

Causa: Comunicação deficiente entre as camadas do sistema.

Consequência: Falta de informação nas páginas, dificuldade na navegação.

Risco 3: Pouco tempo para testes

Causa: Priorização do desenvolvimento e falta de testes para lançamento em produção.

Consequência: Bugs em produção e baixa confiabilidade do sistema.

Risco 4: Conflitos no versionamento de código

Causa: Falta de disciplina no uso do Git.

Consequência: Perda de código e falhas na integração.

Risco 5: Interface pouco amigável

Causa: Falta de foco em design e UX.

Consequência: Má experiência do usuário e rejeição da ferramenta.

Risco 6: Falhas de segurança na autenticação

Causa: Implementação frágil ou mal validada.

Consequência: Acesso indevido e comprometimento de dados.

Risco 7: Falta de documentação técnica

Causa: Priorização exclusiva no desenvolvimento funcional.

Consequência: Dificuldade de manutenção e expansão futura.

Risco 8: Sobreposição de tarefas entre membros

Causa: Falta de definição de papéis claros.

Consequência: Perda de tempo e retrabalho.

Risco 9: Complexidade excessiva em funcionalidades simples

Causa: Superengenharia.

Consequência: Tempo desperdiçado e sistema difícil de manter.

Risco 10: Problemas com deploy (Railway, Render etc.)

Causa: Configurações erradas ou instabilidade da plataforma.

Consequência: Sistema fora do ar ou com falhas.

Risco 11: APIs externas (ex: dados de ações) fora do ar

Causa: Dependência de serviços de terceiros.

Consequência: Funcionalidade quebrada ou desatualizada.

Risco 12: Falta de validação dos dados de entrada

Causa: Falhas nos formulários e na sanitização.

Consequência: Erros de lógica, vulnerabilidades e falhas críticas.

Exercício 2 – Classificação e Análise Qualitativa

Classifique os riscos com base em Probabilidade e Impacto. Complete a tabela abaixo.

Risco	Probabilidade	Impacto	Prioridade
Integração fraca de front-end e back-end	Alta	Média	Alta
APIs externas (ex: dados de ações) fora do ar	Alta	Alta	Alta
Complexidade excessiva em funcionalidades simples	Baixa	Média	Baixa
Problemas com deploy (Railway, Render etc.)	Média	Alta	Alta
Falta de documentação técnica	Alta	Média	Alta

Probabilidade – Baixa, Média, Alta.

Impacto – Baixo, Médio, Alto.

Exercício 3 – Plano de Resposta aos Riscos

Para os 5 riscos mais críticos, defina uma estratégia de resposta e a ação correspondente.

Risco	Estratégia	Ação prática
Integração fraca de front-end e back-end	Reduzir	Definir escopo claro de interface, definir um diagrama de classe e realizar reuniões semanais de alinhamento. Com definições claras de entrega.
APIs externas fora do ar	Aceitar	Implementar fallback de cache local e usar múltiplas fontes de dados.
Problemas com deploy (Railway, Render etc.)	Reduzir	Utilizar hospedagem local como alternativa e avaliar ferramentas e planos para deploy
Falta de documentação técnica	Evitar	Estabelecer rotina de documentação ao final de cada sprint.
Pouco tempo para testes	Reduzir	Planejar tempo exclusivo para testes em cada sprint com responsável designado.

Estratégia – Evitar, Reduzir, Transferir, Aceitar.

Exercício 4 – Plano de Monitoramento

Responda às perguntas abaixo para definir como os riscos serão monitorados.

1. Quem será responsável por acompanhar os riscos?

Thiago Lazari, Product Owner.

2. Quais indicadores serão usados? (Ex: número de bugs, atrasos, falhas)

Atrasos no projeto e inconsistências nos testes efetuados.

3. Com que frequência os riscos serão reavaliados?

No meio do projeto e ao final para mensurar resultados obtidos.

Exercício 5 – Critérios de Aceitabilidade dos Riscos

Para cada risco identificado, calcular o índice de risco (IR) para definir a aceitabilidade do risco envolvido no projeto. O IR é baseado na matriz de classificação de riscos apresentada em aula. Considere os seguintes critérios de decisão:

- Risco é **aceitável** se a gravidade for **desprezível ou crítica** e a probabilidade for **improvável ou remota** não comprometendo, assim, o projeto.

- Risco é **inaceitável** se tiver a gravidade **crítica ou catastrófica** e a probabilidade for **frequente ou remota** colocando em risco a entrega do produto, além de afetar a qualidade, eficiência e segurança.

Complete a tabela a seguir para os riscos do seu projeto:

Risco	Probabilidade	Gravidade	IR
Falta de domínio em Django	Frequente (5)	Crítica (3)	15
Integração fraca de front-end e back-end	Frequente (5)	Crítica (3)	15
Pouco tempo para testes	Frequente (5)	Crítica (3)	15
Conflitos no versionamento de código	Remota (3)	Crítica (3)	9

Interface pouco amigável	Remota (3)	Crítica (3)	9
Falhas de segurança na autenticação	Remota (3)	Catastrófica (5)	15
Falta de documentação técnica	Frequente (5)	Crítica (3)	15
Sobreposição de tarefas entre membros	Remota (3)	Crítica (3)	9
Complexidade excessiva em funcionalidades simples	Improvável (1)	Crítica (3)	3
Problemas com deploy (Railway, Render etc.)	Remota (3)	Catastrófica (5)	15
APIs externas (ex: dados de ações) fora do ar	Frequente (5)	Catastrófica (5)	25
Falta de validação dos dados de entrada	Remota (3)	Catastrófica (5)	15