

A tabela subsequente demonstra a listagem dos riscos de *software* identificados na literatura, fornecendo uma visão abrangente dos elementos que podem influenciar adversamente o desenvolvimento de projetos de *software*, proporcionando informações para a gestão acurada dessas situações, destacando quais autores e fontes identificaram esses riscos.

Tabela 1 – Tabela de Riscos de *Software*

<i>Riscos de Software</i>		
Risco	Descrição	Fonte
Carência de pessoal	Possibilidade de não ter a quantidade necessária de pessoal para executar as tarefas, o que pode afetar o andamento do projeto.	(BOEHM, 1991), (KEIL et al., 1998), (ADDISON; VALLABH, 2002), (SCHMIDT, 1997), (NERUS, s.d.), identificado pela autora
Cronogramas e orçamentos não realistas	Risco relacionado à definição de prazos e orçamentos que não refletem a realidade, podendo levar a atrasos e estouro de custos.	(BOEHM, 1991), (ADDISON; VALLABH, 2002), (SCHMIDT, 1997), (NERUS, s.d.), (UDS, 2023), identificado pela autora
Desenvolvimento de funções e propriedades erradas	Possibilidade de criar funcionalidades ou propriedades que não atendem às reais necessidades do projeto, levando a retrabalho e insatisfação do cliente.	(BOEHM, 1991)
Desenvolvimento de interface de usuário incorreta	Risco de criar uma interface de usuário pouco intuitiva ou de difícil uso, afetando a experiência do usuário final.	(BOEHM, 1991), (ADDISON; VALLABH, 2002)
Adornamento excessivo	Risco de adicionar recursos desnecessários ao projeto, aumentando custos e complexidade sem benefícios significativos.	(BOEHM, 1991), (ADDISON; VALLABH, 2002)
Continuidade de mudanças nos requisitos	Risco associado a alterações frequentes nos requisitos, podendo causar instabilidade no projeto e dificultar o cumprimento de prazos.	(BOEHM, 1991)
Falhas externas	Risco associado à dependência de componentes externos, podendo haver atrasos ou falta de qualidade nesses elementos juntamente com a possibilidade de não atender às expectativas em tarefas realizadas por terceiros, impactando a qualidade global do projeto.	(BOEHM, 1991), (ADDISON; VALLABH, 2002), (SCHMIDT, 1997), (NERUS, s.d.)
Deficiências de desempenho real	O risco de o desempenho do sistema não atender aos requisitos especificados, prejudicando a satisfação do cliente.	(BOEHM, 1991), (SCHMIDT, 1997)

<i>Riscos de Software</i>		
Sobrecarga das capacidades de ciência da computação	Possibilidade de exigir mais recursos de ciência da computação do que disponíveis, levando a atrasos ou falhas na implementação.	(BOEHM, 1991), (KEIL et al., 1998), (ADDISON; VALLABH, 2002)
Falta de monitoramento e controle	Risco de não monitorar e controlar o projeto de forma consistente, prejudicando a identificação precoce de problemas e ajustes necessários.	(NERUS, s.d.)
Falta de sessões de planejamento com o time	Risco de iniciar o projeto sem sessões de planejamento eficazes, levando a falta de entendimento dos papéis, objetivos e colaboração inadequada, e também de não antecipar e gerenciar os riscos do projeto, resultando em falhas e busca por culpados.	(NERUS, s.d.), (UDS, 2023), identificado pela autora
Falta de apoio geral à cultura de projetos	Risco relacionado à falta de apoio da empresa à cultura de projetos, comprometendo o sucesso mesmo com boas ferramentas e controles.	(NERUS, s.d.)
Comunicação ruim	Risco de comunicação inadequada, podendo resultar em problemas e confusões frequentes no projeto, manter o patrocinador pouco informado ou excessivamente envolvido, causando frustrações e dificuldades na execução.	(NERUS, s.d.)
Sobrecarga de trabalho das pessoas envolvidas	Risco de gerar sobrecarga de trabalho, especialmente em empresas de médio e pequeno porte, comprometendo a execução das tarefas.	(NERUS, s.d.)

<i>Riscos de Software</i>		
Envolvimento insuficiente dos <i>stakeholders</i>	Risco de não envolver adequadamente os <i>stakeholders</i> no processo de desenvolvimento, levando a falta de <i>feedback</i> e compreensão insuficiente dos requisitos.	(ADDISON; VALLABH, 2002), (NERUS, s.d.), identificado pela autora
Falta de resiliência a mudanças	Risco de não ser resiliente a mudanças, especialmente em <i>startups</i> e empresas juniores, onde a agilidade e adaptação rápida podem melhorar a flexibilidade e eficiência do processo de desenvolvimento.	Identificado pela autora
Falta de gestão do conhecimento	Risco de não documentar de forma adequada o processo de desenvolvimento, códigos e decisões tomadas, dificultando a manutenção e compreensão futura.	Identificado pela autora
Dependência de indivíduos chave	Risco de depender de indivíduos-chave, tornando o projeto vulnerável a problemas caso esses membros-chave estejam ausentes ou indisponíveis.	Identificado pela autora
Falta de investimento em capacitação	Risco de não investir adequadamente na capacitação contínua da equipe, resultando em defasagem de habilidades e tecnologias ultrapassadas.	Identificado pela autora
Ignorar <i>feedback</i> do usuário	Risco de ignorar o <i>feedback</i> do usuário, especialmente em empresas de pequeno porte, resultando em produtos que não atendem às expectativas e necessidades reais dos clientes.	(UDS, 2023), identificado pela autora
Entendimento falho	Como os detalhes técnicos do projeto ainda não existem, o entendimento da equipe sobre como cada função do <i>software</i> deve se comportar certamente terá falhas e divergências frente ao que o cliente deseja.	(UDS, 2023), (BOEHM, 1991)

<i>Riscos de Software</i>		
Risco	Descrição	Fonte
Dificulta melhorias ou inovações no <i>software</i>	Em projetos de escopo fechado, é difícil aproveitar vantagens pela necessidade de renegociar o investimento e prazo de entrega. Isso dificulta que o <i>software</i> seja o melhor que poderia ser.	(UDS, 2023)
Aumento do escopo e novas negociações	Qualquer item diferente do descrito no contrato inicial que for identificado no detalhamento das funções, será incrementado no escopo do projeto e é muito provável de acontecer, pois seria raro que cada característica do projeto fosse pensada previamente.	(UDS, 2023)
Comprometimento da qualidade	O custo do fornecedor para cumprir com alterações em um projeto de custo e prazo fixos pode comprometer a qualidade dos processos e profissionais envolvidos na produção.	(UDS, 2023)
Prazos estourados	Qualquer divergência nas necessidades do cliente ou atividades do projeto (altamente provável) exigirá que os prazos sejam renegociados.	(UDS, 2023), identificado pela autora
Linha de produção “fordista”	O cliente paga por atividades que não agregam valor ao negócio, já que todas elas foram inicialmente planejadas e “devem” ser entregues, mesmo que não façam mais sentido ao longo do processo de desenvolvimento.	(UDS, 2023), identificado pela autora

Referências

ADDISON, T.; VALLABH, S. Controlling software project risks: an empirical study of methods used by experienced project managers. In: PROCEEDINGS of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology. 2002. P. 128–140.

BOEHM, B. Software risk management: principles and practices. **IEEE Software**, v. 8, n. 1, p. 32–41, 1991. DOI: 10.1109/52.62930.

KEIL, M. et al. A Framework for Identifying Software Project Risks. **Communications of the ACM**, v. 41, p. 76–83, nov. 1998. DOI: 10.1145/287831.287843.

NERUS. **Exemplos de Riscos em Projeto de Software**. Disponível em: <<https://nerus.com.br/blog/processos/exemplos-riscos-projeto-software/>>.

SCHMIDT, R. C. Managing Delphi surveys using nonparametric statistical techniques. **decision Sciences**, Wiley Online Library, v. 28, n. 3, p. 763–774, 1997.

UDS. **7 boas práticas para gerenciar desenvolvimento de software**. 2023. <https://uds.com.br/blog/boas-praticas-para-gerenciar-desenvolvimento-de-software/>. Accessed: 2023-11-06.