

Diretrizes e Boas Práticas de Garantia de Qualidade de Software para o Projeto SobreVidas - “Câncer de Boca”

Gabriel F. dos Reis¹, Tallya J. S. Barbosa¹, Leandro Pedrosa², Luiza de O. Costa¹, Renata D. Braga¹, Rejane F. Ribeiro-Rotta³

¹Instituto de Informática – UFG – Goiânia – GO – Brasil

²Programa de Pós-Graduação em Ciências da Saúde
Faculdade de Medicina - UFG - Goiânia - GO - Brasil

³Faculdade de Odontologia – UFG - Goiânia - GO - Brasil

{tallyabarbosa, freitas.gabriel, leandropedrosa, luizaluiza}
@discente.ufg.br, {renatadbraga, rejaneffr}@ufg.br

Abstract. *The increasing integration of technology in healthcare has transformed patient care, making Software Quality Assurance (SQA) essential for the safety and reliability of critical systems, such as screening and monitoring platforms. In the context of the "SobreVidas – Oral Cancer Project," the absence of systematic SQA guidelines increases risks to patient safety, sensitive data protection, and operational efficiency. This paper presents a set of SQA best practices, based on the principles of Evolutionary Architecture, specifically designed for the SobreVidas Project. The study included a systematic diagnosis of current practices, a literature review focusing on agile and DevOps methodologies, and the design of a quality framework to support the system's development, maintenance, and continuous improvement.*

Resumo. *A crescente integração da tecnologia na saúde tem transformado o cuidado ao paciente, tornando a Garantia da Qualidade de Software (SQA) essencial para a segurança e confiabilidade de sistemas críticos, como as plataformas de rastreamento e monitoramento. No contexto do Projeto SobreVidas – Câncer de Boca, a ausência de diretrizes sistemáticas de SQA aumenta os riscos à segurança do paciente, à proteção de dados sensíveis e à eficiência operacional. Este estudo apresenta um conjunto de boas práticas de SQA, fundamentadas nos princípios da Arquitetura Evolutiva e aplicadas ao Projeto SobreVidas. O estudo envolveu um diagnóstico sistemático das práticas atuais, revisão da literatura sobre metodologias ágeis e DevOps e a proposição de um framework de qualidade voltado ao desenvolvimento, manutenção e aprimoramento contínuo do sistema.*

1. Introdução

A crescente integração da tecnologia na área da saúde tem revolucionado o cuidado ao paciente, promovendo inovações que aumentam a precisão diagnóstica, a eficácia terapêutica e a eficiência operacional. Neste cenário em constante evolução, os sistemas de informação em saúde assumem papel cada vez mais central na prestação de serviços de saúde, exigindo soluções confiáveis para lidar com a complexidade do setor [Epizitone et al. 2023].

No Brasil, o câncer de boca representa um problema para a saúde pública. A ausência de estratégias nacionais sistematizadas de prevenção para esse tipo de carcinoma resulta em um cenário no qual a maioria dos casos é diagnosticada em estágios avançados, reduzindo significativamente as chances de cura e a sobrevida dos pacientes. Estima-se que mais de 80% dos casos tratados no país sejam detectados tarde, demandando intervenções invasivas, mutilantes e de alto custo, com sérios impactos na qualidade de vida e na produtividade de indivíduos em idade economicamente ativa. Em contrapartida, quando identificado precocemente, o câncer de boca apresenta taxas de cura superiores a 90%, ressaltando a relevância de projetos voltados à detecção precoce. Nesse contexto, profissionais da Atenção Primária à Saúde e Agentes Comunitários de Saúde desempenham um papel estratégico, dada sua proximidade com a população e sua capacidade de implementar ações de rastreamento e monitoramento [Speight et al. 2017, Ribeiro-Rotta et al. 2022].

É neste cenário que se insere o Projeto SobreVidas – “Câncer de Boca”, concebido como uma plataforma tecnológica para rastreamento e monitoramento da população de risco. Considerando a sensibilidade dos dados de saúde e o impacto direto na vida dos pacientes, a Garantia de Qualidade de Software (SQA) torna-se um requisito imperativo. Softwares desenvolvidos sem padrões adequados podem resultar em erros graves, comprometer a segurança do paciente e reduzir a confiança nos sistemas de saúde [Ronchieri and Canaparo 2023]. Além disso, a conformidade com regulamentações legais, como a Lei Geral de Proteção de Dados (LGPD), é indispensável para assegurar a privacidade das informações sensíveis e prevenir responsabilidades legais e danos reputacionais. A detecção precoce de falhas no desenvolvimento reduz custos e riscos, fortalecendo a confiabilidade das soluções digitais em saúde [Conduah et al. 2025].

O desenvolvimento de tecnologias voltadas para a saúde digital, como o Projeto SobreVidas, requer metodologias ágeis e adaptáveis, capazes de responder rapidamente às mudanças nas necessidades dos usuários e atender às exigências regulatórias. Nesse sentido, o Projeto SobreVidas adotou práticas ágeis e de versionamento como fundamentos essenciais para assegurar flexibilidade, rastreabilidade e continuidade no desenvolvimento, reforçando a integração entre equipes multidisciplinares [Desai et al. 2024].

No entanto, nesse contexto de adaptabilidade, assegurar a corretude do sistema anterior as mudanças junto com os incrementos se torna uma tarefa muito custosa para a equipe. Isso aumenta a possibilidade de erros passarem despercebidos para a versão final e afetando de forma, parcial ou total o funcionamento da plataforma. Esses erros quando ocorridos, devem ser resolvidos o mais rápido possível, o que altera o foco da equipe de objetivos futuros para uma solução rápida, muitas vezes exigindo intervenções que são funcionais, mas não ideais para uma manutenção de longo prazo. Uma percepção geral da equipe antes de elaborar e implementar diretrizes de qualidade era de que ao invés de focarem em pesquisa e desenvolvimento, deveriam estar prontos a qualquer momento para reagirem a incidentes, o que minava a confiança na plataforma, ademais, o tempo gasto para assegurar a corretude com testes manuais aumentava proporcionalmente com o incremento do sistema.

Para enfrentar esses desafios e aprimorar as práticas de SQA no Projeto SobreVidas – ‘Câncer de Boca’, este estudo propõe diretrizes baseadas nos princípios da Arquitetura Evolutiva. Esta abordagem assegura que a plataforma seja capaz de evoluir de forma

modular e resiliente, acompanhando mudanças tecnológicas, regulatórias e funcionais, além de incorporar automação e monitoramento contínuo. Nesse contexto, a atuação da equipe de QA e Testes no projeto tem sido determinante, tanto na definição de processos de qualidade quanto na construção de testes automatizados de API que funcionam, também, como testes de regressão. Essa experiência fundamenta as diretrizes apresentadas, com foco em escalabilidade, segurança e confiabilidade [Hauschild et al. 2022].

Espera-se que a aplicação dessas diretrizes fortaleça a segurança, a eficiência, a adaptabilidade e a conformidade da plataforma, assegurando confiança à sociedade, aos trabalhadores da saúde e ao mercado. [Sowunmi et al. 2016]. Assim, este estudo teve como objetivo compartilhar a experiência do desenvolvimento e aplicação dessas práticas no Projeto SobreVidas - "Câncer de Boca", destacando os impactos observados no andamento do projeto.

2. Método

Este estudo adotou uma abordagem metodológica mista, combinando técnicas qualitativas e quantitativas. Inicialmente, foi realizado um mapeamento das práticas existentes no projeto, de modo a identificar quais poderiam ser mantidas, aprimoradas ou descartadas, possibilitando a definição de novas diretrizes de qualidade [Ronchieri and Canaparo 2023].

A priori, realizou-se um diagnóstico sistemático da documentação e dos processos vigentes, incluindo a análise de documentos de requisitos, artefatos arquiteturais e metas estabelecidas. As práticas de revisão de código foram avaliadas criticamente. Em paralelo, conduziu-se uma revisão da literatura científica sobre SQA e Arquitetura Evolutiva, com ênfase em aplicações no setor de saúde digital.

Para incorporar automação ao projeto de desenvolvimento, foi implementada uma infraestrutura de testes automatizados, com o objetivo de validar a corretude de desenvolvimentos passados e novos, além de garantir que os novos não mitigassem os já existentes, também alinhado ao princípio da pirâmide de testes [Cohn 2009]. Testes unitários e de integração foram construídos primeiro pela facilidade e velocidade para serem escritos, pela equipe de back-end, utilizando JUnit e SonarQube para análise estática de código, garantindo assim que classes pudessem ser testadas isoladamente e regras de negócios validadas nos níveis mais baixos, além de garantir que novos desenvolvimentos não quebrassem desenvolvimentos passados. A equipe de QA desenvolveu os testes de API voltados à validação da integração com a interface de usuário (UI). Além disso, foram realizados testes de carga e desempenho. Essa infraestrutura foi acoplada à *pipeline* de Integração Contínua/Entrega Contínua (CI/CD) no Jenkins, garantindo a execução automática e a detecção precoce de falhas [Vaddadi et al. 2023].

Após a consolidação dos testes, foram definidas e aplicadas diretrizes de SQA, contemplando todo o ciclo de vida do *software*: desde a elicitação de requisitos até a validação técnica e de negócios. A avaliação do impacto das diretrizes ocorreu por meio da coleta de *feedbacks* durante as revisões de *sprints*, considerando tanto os resultados práticos obtidos quanto a percepção da equipe sobre os efeitos das práticas no andamento do projeto [Ronchieri and Canaparo 2023].

Com a expansão da equipe, foi instituída a revisão de código por pares (*pair review*). Essa prática possibilitou a identificação de problemas de qualidade, promo-

veu o aprendizado colaborativo e reforçou padrões de excelência no desenvolvimento do código-fonte [Kim et al. 2023].

3. Resultados

Após a revisão, foi constadado que não havia diretrizes claras de qualidade de Software, apenas menções a testes manuais sem critérios bem definidos de aceitação. Ademais, não havia testes automatizados, tanto de funcionalidades já desenvolvidas, quanto de novas funcionalidades para validar se os requisitos foram totalmente atendidos, isso em todas as camadas do *software*. Por fim, também não havia testes para medir a performance da plataforma em situações de grande uso e estresse, o que fez o projeto andar as cegas em relação a como o sistema se comportaria em situações de uso real.

A implementação de testes automatizados e a elaboração de diretrizes de SQA abrangentes foram essenciais para garantir a qualidade e a evolução contínua da Plataforma SobreVidas – “Câncer de Boca”. A aplicação dessas diretrizes desde a concepção até a validação e *deploy* permitiu entregas incrementais e contínuas de funcionalidades, assegurando maior segurança e confiabilidade no ciclo de desenvolvimento.

Os testes automatizados foram desenvolvidos em Python em um repositório de código separado dos demais, simulando as ações de um usuário na interface gráfica por meio da API fornecida do *back-end*. Inicialmente, foram construídos testes para *endpoints* isolados, validando as respostas HTTP em requisições válidas e inválidas. Em seguida, foram implementados testes para validar os fluxos de usos integrados, garantindo a interoperabilidade entre diferentes módulos da plataforma. A etapa seguinte consistiu na aplicação da diretriz de *Shift-left testing* [Vaddadi et al. 2023]: a cada nova entrega contendo interfaces de API, um teste era criado antecipadamente junto ao desenvolvimento, com base nas especificações técnicas fornecidas pelo *Tech Lead* do projeto. Essa prática permitiu a identificação precoce de melhorias, além de proporcionar validação inicial pela execução bem-sucedida dos testes [Hauschild et al. 2022].

Uma automação no ambiente de testes foi implementada para executar todos os cenários diariamente, consolidando relatórios de sucesso, falha ou erro e, por fim, enviá-los ao servidor de comunicação do projeto no Discord. Com o objetivo de informar todos os integrantes do estado de qualidade da Plataforma logo no início do dia, possibilitando análises rápidas e diárias ao analisar os testes que eventualmente tinha o resultado de falha. Os testes automatizados de API também passaram a atuar como testes de regressão contínuos, com o objetivo de garantir maior estabilidade no ciclo de desenvolvimento, apontando se, eventualmente, novos incrementos mitigaram desenvolvimentos de funcionalidades já estabelecidas no sistema. O relatório era compartilhado em formato de mensagem de texto para toda a equipe através de um canal no servidor da plataforma Discord (Figura 1).

A *Squad* de desenvolvimento *back-end* ficou responsável pela construção de testes unitários usando JUnit, para validar isoladamente as lógicas de cada classe do código e pela análise da cobertura e qualidade de código por meio do SonarQube, uma ferramenta que inspeciona continuamente o código produzido pela equipe, fornecendo métricas do ecossistema Java. Essa solução viabilizou um acompanhamento contínuo da qualidade do código-fonte, com relatórios sistemáticos sobre a segurança, confiabilidade, manutenibilidade e a porcentagem de código que possui cobertura de testes unitários (Figura 2).

```

-----
Ran 77 tests in 95.429s
FAILED (failures=5)

Test Results:
Total run time: [94.996s]
test_atendimento_get_by_id ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.483s]
test_atendimento_nova_decisao_clinica ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.413s]
test_atendimento_novo_acompanhamento ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.353s]
test_atendimento_novo_resultado ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.382s]
test_delete_atendimento_acompanhamento ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.356s]
test_delete_atendimento_decisao_clinica ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.378s]
test_delete_atendimento_resultado ( testcase_atendimento.AtendimentoTestCase ) ... ok [0.369s]
test_fator_risco_all_get ( testcase_fator_risco.FatorRiscoTestCase ) ... ok [0.005s]
test_fator_risco_crud ( testcase_fator_risco.FatorRiscoTestCase ) ... ok [0.032s]

```

Figura 1. Notificação de resultado da execução dos testes automatizados da Plataforma Sobrevidas - “Câncer de Boca”.

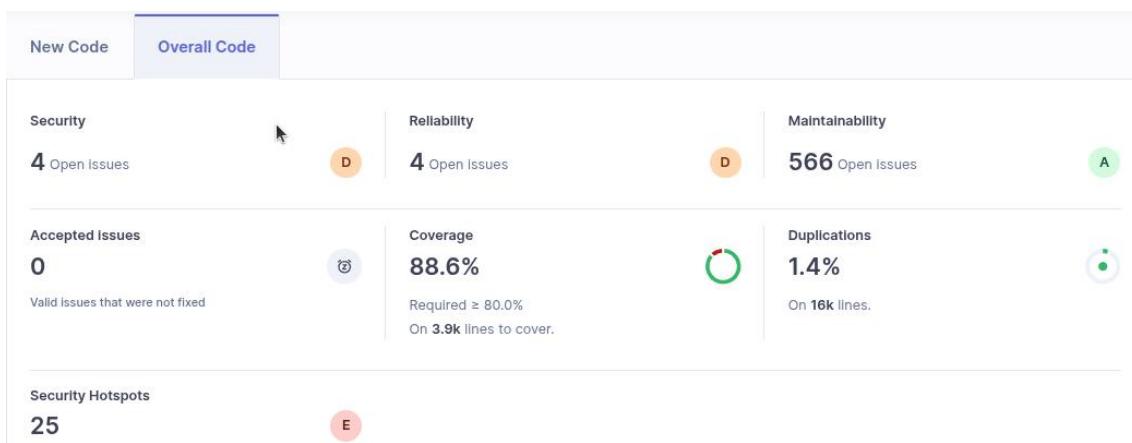


Figura 2. Relatório da análise de cobertura de testes unitários e qualidade de código feito pela ferramenta SonarQube.

Além disso, foram conduzidos testes de carga e desempenho com *framework* LoadRunner, simulando ambientes de uso de acordo com as expectativas de implementação. Os relatórios em tempo real permitiram monitorar requisições por segundo, tempo médio de resposta e número de usuários simulados, garantindo que a solução estivesse preparada para atender volumes significativos de acessos simultâneos (Figura 3). Essa etapa foi essencial para assegurar a escalabilidade da Plataforma [Obigbesan et al. 2024, Usman et al. 2018].

O refinamento de requisitos ocorreu de forma colaborativa entre a *Squad* de Negócio, o *Tech Lead* e os stakeholders, reduzindo ambiguidades e acelerando o desenvolvimento. A *Squad* de QA estabeleceu e decomentou critérios de aceitação específicos para cada *squad*, contribuindo para aumentar a rastreabilidade e a consistência das entregas entre elas:

- **Squad Back-end:** Desenvolvimento guiado por *TDD* (*Test Driven Development*), cobertura mínima de 80% em testes unitários no *SonarQube*, ausência de *warnings* críticos, validação das regras de negócio em testes automatizados e documentação de novos ou modificados *endpoints* no *Swagger*.
- **Squad Frontend:** Interfaces em conformidade com os protótipos do *Figma*, responsividade garantida para múltiplas resoluções, usabilidade intuitiva, ausência de



Figura 3. Estatísticas de desempenho da plataforma durante os testes de carga e desempenho feitos.

hard code na *UI*, delegação de processamentos pesados ao *back-end* e validação via testes funcionais *end-to-end*.

- **Squad IA:** Validação de modelos em *dataset* acordado, documentação completa do processo de criação ou uso de modelos existentes e proposição de casos de uso.

Adicionalmente, as tarefas técnicas passaram a adotar revisão de código por pares (*peer review*), o que proporcionou uma segunda camada de avaliação técnica, favoreceu o compartilhamento de conhecimento e consolidou uma cultura de excelência no desenvolvimento. Como resultado, as diretrizes implementadas permitiram que todos os integrantes compartilhassem a responsabilidade pela qualidade, tanto técnica quanto de processo, reforçando a confiabilidade das entregas e mitigando riscos para o usuário final.

4. Discussão

A adoção de diretrizes de qualidade no Projeto SobreVidas - "Câncer de Boca" demonstrou ser uma estratégia efetiva para garantir segurança, manutenção e evolução constante da Plataforma, aspectos essenciais no contexto da saúde. A aplicação das diretrizes em todas as fases do desenvolvimento permitiu entregas incrementais de requisitos, funcionalidades e melhorias, o que não só assegurou maior confiabilidade nas novas demandas, mas também facilitou a detecção precoce de erros, bugs e gargalos de desempenho [Hauschild et al. 2022].

O monitoramento contínuo da qualidade e da cobertura de testes realizados pelo SonarQube e pela infraestrutura de testes automatizados foi fundamental para construir

confiança no código existente e nas modificações subsequentes. Esse monitoramento permitiu que novas funcionalidades não comprometessem funcionalidades já implementadas, por meio de testes de regressão automatizados, garantindo que a Plataforma mantivesse a sua integridade, mesmo com a integração entre API e UI.

Ademais, esse monitoramento constante viabilizou a identificação de falhas e pontos de melhoria em estágios iniciais de desenvolvimento, evitando que problemas chegassem ao ambiente de produção e impactassem os usuários finais. Assim, a equipe pôde focar mais em inovação e melhorias do que em correções emergenciais, adotando um perfil mais proativo do que reativo frente a falhas.

A exigência dos critérios de aceitação, com aprovação pela squad de QA somente após o cumprimento integral desses critérios, foi determinante para evitar o acúmulo de débito técnico, minimizar gargalos e assegurar boa documentação e rastreabilidade. Essa prática favoreceu a passagem de conhecimento entre membros da equipe e facilitou a integração de novos pesquisadores no projeto.

Os testes de carga e desempenho foram cruciais para determinar os limites operacionais da Plataforma. Em simulações de uso incremental, identificou-se que, a partir de cerca de 50 usuários simultâneos, o tempo médio de resposta por endpoint ultrapassava 20 segundos, implicando em degradação linear nos fluxos de uso. Com essa informação, foram iniciadas discussões sobre otimizações infraestruturais e arquiteturais para suportar cargas maiores, visando implantação em localidades maiores, como municípios da região metropolitana de Goiânia-Goiás.

Por fim, a implementação iterativa das diretrizes de SQA favoreceu a adoção crescente por parte da equipe, promovendo uma mudança de mentalidade: a qualidade deixou de ser vista como uma etapa final ou parcela isolada do processo e passou a ser integrada ao fluxo contínuo de desenvolvimento. Os resultados positivos iniciais, especialmente a segurança decorrente dos testes automatizados, contribuíram para que a equipe valorizasse as práticas de qualidade como parte intrínseca do processo, não como acréscimo, de forma que nas documentações do projeto constam as diretrizes aplicadas para seguir o processo de promover a cultura de qualidade. [Alami and Krancher 2022].

5. Conclusões

O desenvolvimento e a implementação de diretrizes de qualidade mostraram-se essenciais para o andamento eficaz do Projeto Sobrevidas - “Câncer de Boca”, garantindo a sua evolução segura e confiável. A combinação dessas diretrizes favoreceu uma melhor interação entre as squads, sustentada por processos de revisão de código colaborativos, avaliação dos critérios de aceitação pela squad de QA e pela atuação integrada do Tech-lead com a squad de Negócio e os stakeholders, resultando em requisitos mais claros e em documentação técnica consistente para implementação e desenvolvimento da Plataforma.

A utilização das ferramentas Python, Locust, JUnit e SonarQube, em conjunto com processos bem definidos de desenvolvimento e aprovação, trouxe segurança para a equipe, possibilitando entregas assertivas, seguras e confiáveis, tanto no aspecto técnico quanto no alinhamento às regras de negócio.

Este estudo reforça a importância da aplicação de diretrizes de qualidade em projetos na área da saúde, contribuindo para a construção de um caminho sólido para a evolução

contínua das soluções digitais. No contexto do setor público, tal prática é ainda mais relevante, considerando os desafios de escalabilidade, confiabilidade e segurança. O Projeto SobreVidas - “Câncer de Boca” configura-se, assim, como um exemplo aplicável a outras iniciativas em saúde digital, demonstrando como boas práticas de SQA podem apoiar o desenvolvimento de tecnologias voltadas à detecção precoce do câncer de boca e ao fortalecimento da saúde pública no Brasil.

6. Agradecimentos

Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento desta pesquisa, concedido através da Chamada nº 21/2023 – Estudos Transdisciplinares em Saúde Coletiva.

Referências

- Alami, A. and Krancher, O. (2022). How scrum adds value to achieving software quality? *Empirical Software Engineering*, 27(7):165.
- Cohn, M. (2009). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional.
- Conduah, A. K., Ofoe, S., and Siaw-Marfo, D. (2025). Data privacy in healthcare: Global challenges and solutions. *DIGITAL HEALTH*, 11:20552076251343959.
- Desai, M., Tardif-Douglin, M., Miller, I., Blitzer, S., Gardner, D. L., Thompson, T., Edmondson, L., and Levine, D. M. (2024). Implementation of agile in healthcare: methodology for a multisite home hospital accelerator. *BMJ Open Quality*, 13(2):e002764.
- Epizitone, A., Moyane, S. P., and Agbehadji, I. E. (2023). A systematic literature review of health information systems for healthcare. *Healthcare*, 11(7).
- Hauschild, A. C., Martin, R., Holst, S. C., Wienbeck, J., and Heider, D. (2022). Guideline for software life cycle in health informatics. *iScience*, 25(12):105534.
- Jamieson, T., Mamdani, M. M., and Etchells, E. (2019). Linking quality improvement and health information technology through the qi-hit figure 8. *Applied Clinical Informatics*, 10(3):528–533.
- Kim, J. Y., Boag, W., Gulamali, F., Hasan, A., Hogg, H. D. J., Lifson, M., Mulligan, D., Patel, M., Raji, I. D., Sehgal, A., Shaw, K., Tobey, D., Valladares, A., Vidal, D., Balu, S., and Sendak, M. (2023). Organizational governance of emerging technologies: Ai adoption in healthcare. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’23, page 1396–1417, New York, NY, USA. Association for Computing Machinery.
- Noël, R., Taramasco, C., and Márquez, G. (2022). Standards, processes, and tools used to evaluate the quality of health information systems: Systematic literature review. *J Med Internet Res*, 24(3):e26577.
- Obigbesan, O., Graham, K., and Benzies, K. M. (2024). Software testing of ehealth interventions: Existing practices and the future of an iterative strategy. *JMIR Nursing*, 7:e56585.

- Ribeiro-Rotta, R. F., Rosa, E. A., Milani, V., Dias, N. R., Masterson, D., da Silva, E. N., and Zara, A. L. d. S. A. (2022). The cost of oral cancer: A systematic review. *PLOS ONE*, 17(4):1–26.
- Ronchieri, E. and Canaparo, M. (2023). Assessing the impact of software quality models in healthcare software systems. *Health Systems*, 12(1):85–97.
- Sowunmi, O. Y., Misra, S., Fernandez-Sanz, L., Crawford, B., and Soto, R. (2016). An empirical evaluation of software quality assurance practices and challenges in a developing country: a comparison of nigeria and turkey. *SpringerPlus*, 5:1921.
- Speight, P. M., Epstein, J., Kujan, O., Lingen, M. W., Nagao, T., Ranganathan, K., and Vargas, P. (2017). Screening for oral cancer-a perspective from the global oral cancer forum. *Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology*, 123(6):680–687.
- Usman, O., Oshiro, C., Chambers, J. C., Tu, S. W., Martins, S., Robinson, A., and Goldstein, M. K. (2018). Selecting test cases from the electronic health record for software testing of knowledge-based clinical decision support systems. *AMIA Annual Symposium Proceedings*, 2018:1046.
- Vaddadi, S. A., Thatikonda, R., Padthe, A., and et al. (2023). Shift left testing paradigm process implementation for quality of software based on fuzzy. *Soft Computing*.