Aplicando Técnicas de V&V na Fase de Projeto do Modelo em Cascata

Naihara Sousa Amorim, Gilleanes Thorwald Araujo Guedes

[1]Universidade Federal do Pampa (UNIPAMPA)
Campus de Alegrete RS – Brasil Curso de Engenharia de Software

naiharamosouza@gmail.com, gilleanesguedes@unipampa.edu.br

Abstract. *The Waterfall Model has well defined steps that occur only once, since it is a non-iterative model. A problem perceived during the Waterfall Model study is that there is no specific activity in the Design stage that verifies the system models created in it, which can lead to the construction of a system with defects. Therefore, the purpose of the article is to integrate Verification and Validation techniques in the Design phase, in order to improve the construction of the system models of the project. In this way, the process will have more technical precision, contributing to the quality of the final product.*

Resumo. *O Modelo em Cascata possui fases bem definidas e que ocorrem apenas uma vez, visto que é um modelo não iterativo. Um problema percebido durante o estudo do Modelo em Cascata é a inexistência de uma atividade específica na fase de Projeto, que verifique os modelos produzidos. Sendo assim, a proposta deste artigo é, integrar na fase de Projeto técnicas de Verificação e Validação, a fim de inspecionar os modelos produzidos durante o projeto. Desta maneira pretende-se que o processo tenha mais precisão técnica, contribuindo para a qualidade do produto final.*

1. Introdução

A Engenharia de Software é uma área de estudo que busca melhorar o desenvolvimento de software com o objetivo de torná-lo mais eficaz, mais robusto e com melhor qualidade. Com este objetivo pesquisadores dessa área propuseram vários processos de desenvolvimento. Um dos primeiros processos propostos foi o Modelo em Cascata.

As fases do modelo em Cascata possuem um encadeamento, sendo que, uma etapa não deve ser iniciada antes que uma etapa anterior seja concluída. Na prática, esses estágios se sobrepõem e alimentam uns aos outros de informações. O modelo em Cascata é também um processo dirigido a planos, ou seja, suas atividades são planejadas antes de serem executadas. A fase de Projeto no Modelo em Cascata, é um processo de vários passos que é centralizado em quatro atributos diferentes do sistema: estrutura de dados, arquitetura do software, detalhes procedais e caracterização das interfaces [Sommerville 2011]. O processo de projeto representa os requisitos de uma forma que permita a codificação do produto (é uma prévia etapa da codificação). Da mesma maneira que a análise dos requisitos, o projeto deve ser documentado. Para atingir este objetivo, o projetista deve mapear as estruturas e funcionalidades identificadas na análise de requisitos dentro do contexto e das restrições da arquitetura, de forma a tornar possível a construção do software, isto é feito principalmente por meio da criação de modelos.

Todavia, Segundo [Sommerville 2011] problemas de modelagem, podem levar a sistemas mal estruturados, quando os problemas de projeto são contornados por artifícios de implementação. Uma das possíveis maneiras de mitigar possíveis defeitos nos artefatos é a aplicação de técnicas de Verificação e Validação(V & V). Estas técnicas visam melhorar a execução do processo e reduzir a quantidade de retrabalho, bem como evitar que soluções inadequadas possam ser implementadas nas etapas seguintes. Assim com o objetivo de aumentar a qualidade do produto, propõe-se integrar ao Modelo em Cascata a utilização de técnicas de V & V, que serão aplicadas nos modelos construídos durante a fase de Projeto.

Este artigo possui a seguinte estrutura: na seção (2) o Modelo em Cascata é apresentado em detalhes com ênfase na fase de Projeto. A seção (3) descreve algumas técnicas de Verificação e Validação que podem ser aplicadas aos modelos de um projeto. Na seção (4) é proposta a inserção de técnicas de Verificação e Validação no modelo cascata durante a fase de projeto. Na última seção são apresentadas as conclusões do artigo.

2. Modelo em Cascata

Um dos Modelos de processo mais tradicionais na Engenharia de Software é o Modelo em Cascata, que tem como principal característica um conjunto de etapas e atividades bem definidas. Este modelo segue um fluxo linear e só avança para a próxima fase quando a fase anterior estiver terminada e revisada [Wazlawick 2012].

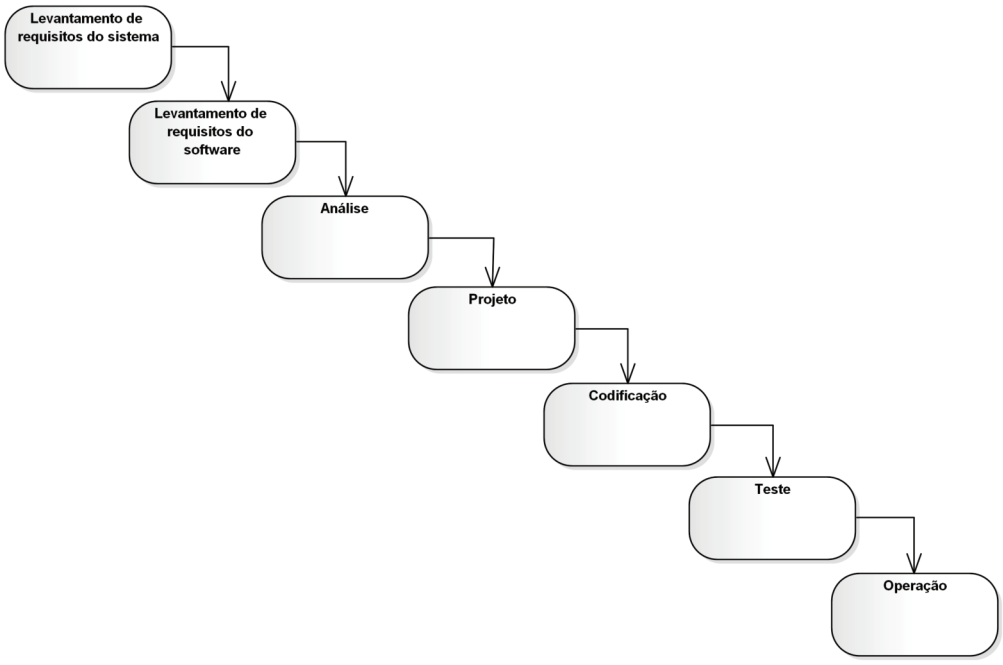

Figura 1. Modelo Cascata [Wazlawick 2012]

O Modelo baseia-se na filosofia BDUF (Big Design Up Front), que propõe que antes de produzir linhas de código deve-se fazer um trabalho detalhado da fase de análise e projeto. A utilização do Cascata sugere que todas as etapas do processo sejam bem documentadas e bem definidas, com destaque para as fases anteriores ao desenvolvimento

de linhas de código. Estas etapas são as que mais produzem artefatos de software e são de extrema importância para que o produto final satisfaça os requisitos do cliente.

Dentro do Modelo em Cascata clássico existem sete (7) fases[Wazlawick 2012]: Levantamento de requisitos do Sistema, Levantamento de Requisitos do Software, Análise, Projeto do Programa, Codificação, Teste, e Operação, conforme pode ser observado na Figura 1.

Na fase de levantamento de requisitos do software as metas e restrições do produto são estabelecidas por meio de consulta aos usuários. A fase seguinte, de levantamento de requisitos do sistema, aprofunda e detalha essas informações, enquanto que a fase de análise verifica inconsistências e conflitos nos requisitos.

Na fase de projeto são identificadas e descritas as abstrações fundamentais do software e seus relacionamentos. Após seu término, na fase de codificação o projeto do software é traduzido em um conjunto de programas ou unidades de programa e, na fase de teste, esse código é testado para determinar se o software funciona adequadamente de acordo com o que foi solicitado. Finalmente na fase de operação, o sistema passa a ser utilizado. Em geral está é a fase mais longa do ciclo de vida.

O enfoque deste trabalho concentra-se na fase de projeto. A figura 2 representa por meio do Modelo BPMN (Business Process Model and Notation), os passos desta etapa como definidos por [Boehm 1981].

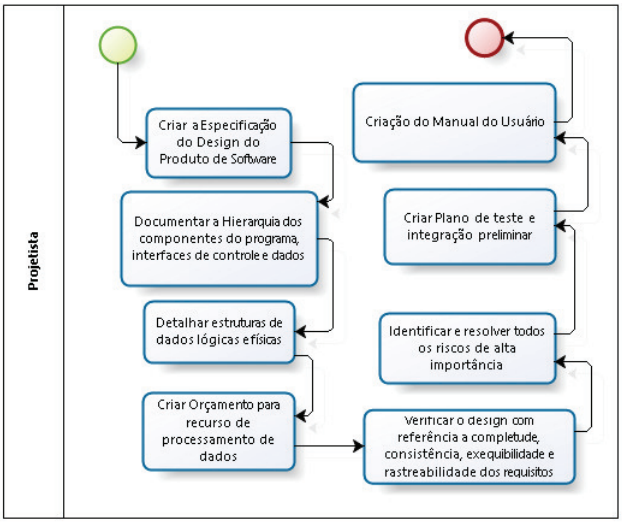

Figura 2. Atividades da etapa de projeto [Boehm 1981]

3. Técnicas de Verificação e Validação(V & V)

A qualidade é hoje um grande motivador em todas as áreas de atividade humana, todos querem oferecer e receber produtos e serviços com qualidade [Tsukumo et al. 1997]. A cada dia é maior o número de empresas que buscam melhorias em seus processos de desenvolvimento de software. Além do aumento da produtividade e da diminuição do retrabalho, elas buscam um melhor relacionamento com seus clientes por meio do melhor planejamento e gestão de suas atividades de desenvolvimento e da diminuição no número de defeitos nos produtos entregues [Jomori et al. 2004].

O processo de Garantia da Qualidade utiliza-se de atividades de V&V, com o intuito de encontrar erros em todas as etapas do desenvolvimento. Essas atividades são muito importantes, pois cuidam de analisar diversos pontos do processo de desenvolvimento, impedindo que um erro se propague para as fases posteriores do projeto [Melo 2009].

Sobre o propósito da V&V, pode-se dizer que o objetivo é confirmar que cada produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados. Já com relação à validação, seu propósito é confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido [Longen 2013].

De acordo com a [IEEE 1990] "verificação e validação(V&V) é o processo que consiste em se determinar: se os requisitos do sistema ou componentes estão completos e corretos; se os produtos de cada fase de desenvolvimento preenchem os requisitos ou condições impostas pela fase anterior; se o sistema final ou componente está de acordo com os requisitos especificados".

Verificação é considerada uma técnica estática que avalia a qualidade do documento e corrige defeitos presentes no documento sem a necessidade de exercitar uma aplicação. Se defeitos não são detectados e corrigidos na fase inicial de desenvolvimento, o custo será muito alto, porque os defeitos serão propagados para a próxima fase do desenvolvimento. Já a validação é considerada uma técnica dinâmica porque envolve exercitar uma implementação. Técnica estáticas podem ser usadas em todos os estágios do processo de desenvolvimento, enquanto técnicas dinâmicas são usadas quando um protótipo ou um programa executável está disponível[Sommerville 2011].

Inspeção de software é um tipo particular de revisão que possui um processo de detecção de defeitos rigoroso e bem definido [Fagan 1976]. È considerada um tipo de verificação estática, que tem o objetivo de detectar defeitos antes que a fase de teste inicie, contribuindo para melhorar a qualidade do software em relação ao custo beneficio [Debaud 1990]. Inspeções foram criadas por Michael Fagan da IBM [Fagan 1976].O processo para inspeção formal contém atividades e papéis bem-definidos, dados do produto, o produto a ser inspecionado e uma infra-estrutura de suporte [NASA 1993].

Segundo o padrão [IEEE 1998], o processo de inspeção contém características que verificam se o elemento de software está conforme o padrão aplicado e identificam desvios de padronização e especificação. Pode ser usada em todos os estágios do desenvolvimento de software e são aplicadas em um artefato em particular, como especificação de requisitos, projeto e código.

Segundo [Bertini 2006] a inspeção é um método que contribui para garantir a qualidade do produto de software. Todas as etapas do processo de desenvolvimento de software são suscetíveis à incorporação de defeitos, que podem ser detectados pela inspeção e posteriormente removidos.

Concluindo de acordo com [Melo 2009] a inspeção de modelo é uma das técnicas que cujo objetivo é encontrar erros a partir da leitura. Além disso, a inspeção é uma técnica usada durante revisões formais. Sua utilização tem como intuito, por exemplo: descobrir erros de função, de lógica ou implementação para qualquer representação do software e também garantir que o software foi representado de acordo com padrões pré-

definidos, garantindo, assim, que o sistema seja desenvolvido de maneira uniforme.

As atividades do processo de inspeção é formal e composto por seis atividades: planejamento; visão geral; preparação; reunião da inspeção; re-trabalho; e acompanhamento.

4. Integração de V&V sob Modelo em Cascata durante a fase de Projeto

A Verificação e Validação sob Modelo Cascata agregam valor ao processo, principalmente pelo fato de que há grande volume de documentação, em destaque nas etapas de análise e de projeto, gerando diversos tipos de artefatos, o que contribui para uma melhor construção do produto final. Como o Modelo preza a qualidade dos artefatos gerados, técnicas de Verificação e Validação potencializam a qualidade do software antes mesmo de ser codificado.[Wazlawick 2012]

4.1. Integração de Modelos na Fase de projeto

Na fase de projeto é onde a maioria de modelos utilizados para a construção do projeto são criados. Os documentos produzidos em cada fase podem ser modificados para refletirem as alterações feitas em cada um deles [Sommerville 2011]. Portanto percebe-se que erros na modelagem do sistema podem ser encontrados nas etapas seguintes, quando teoricamente a etapa de projeto deveria estar revisada e finalizada. Os problemas encontrados causam retrabalho não apenas na fase de projeto, mas também em suas etapas posteriores.

Em nossa proposta pretendemos focar fortemente nas técnicas de V&V. As técnicas serão aplicadas sobre os modelos a fim de identificar possíveis defeitos na modelagem. Caso seja descoberta alguma irregularidade no projeto, o problema será documentado e terá seu impacto mapeado em outros artefatos além de, claramente, ser corrigido pelo projetista. A Figura 3 descreve detalhadamente as etapas da fase de projeto integrada à técnicas de V&V.

A primeira etapa de nossa proposta é a fase de planejamento, na qual são definidos os participantes da inspeção, seus respectivos papéis e responsabilidades. Nessa etapa o material de inspeção é repassado aos inspetores. A atividade de visão geral consiste em uma reunião, durante a qual são fornecidas à equipe de inspeção as informações necessárias à inspeção. A reunião de visão geral ocorrerá se a equipe de inspeção não possuir familiaridade com o produto. Durante a atividade de preparação, os artefatos são verificados e validados com o objetivo de encontrar defeitos e, caso algum seja encontrado, classificá-los de acordo com o tipo de defeito. A atividade de preparação, de acordo com [Cabral 2007], é considerada a mais importante de uma inspeção, porque seu foco principal é a detecção de defeitos.

A fase de preparação na nossa proposta deve ser organizada individualmente. Após isso, durante a atividade denominada de reunião de inspeção ocorre a consolidação de defeitos encontrados durante a fase de preparação. O objetivo da atividade de retrabalho é corrigir os defeitos encontrados, os defeitos classificados com prioridade alta serão corrigidos de imediato, enquanto os de prioridade baixa só o serão caso o custo e o prazo permitirem. Após verificar se todos os defeitos foram corrigidos, o Analista de Modelo deve aprovar o artefato e assim registrar o fim da inspeção e, caso o produto não for aprovado, deverá notificar o projetista.

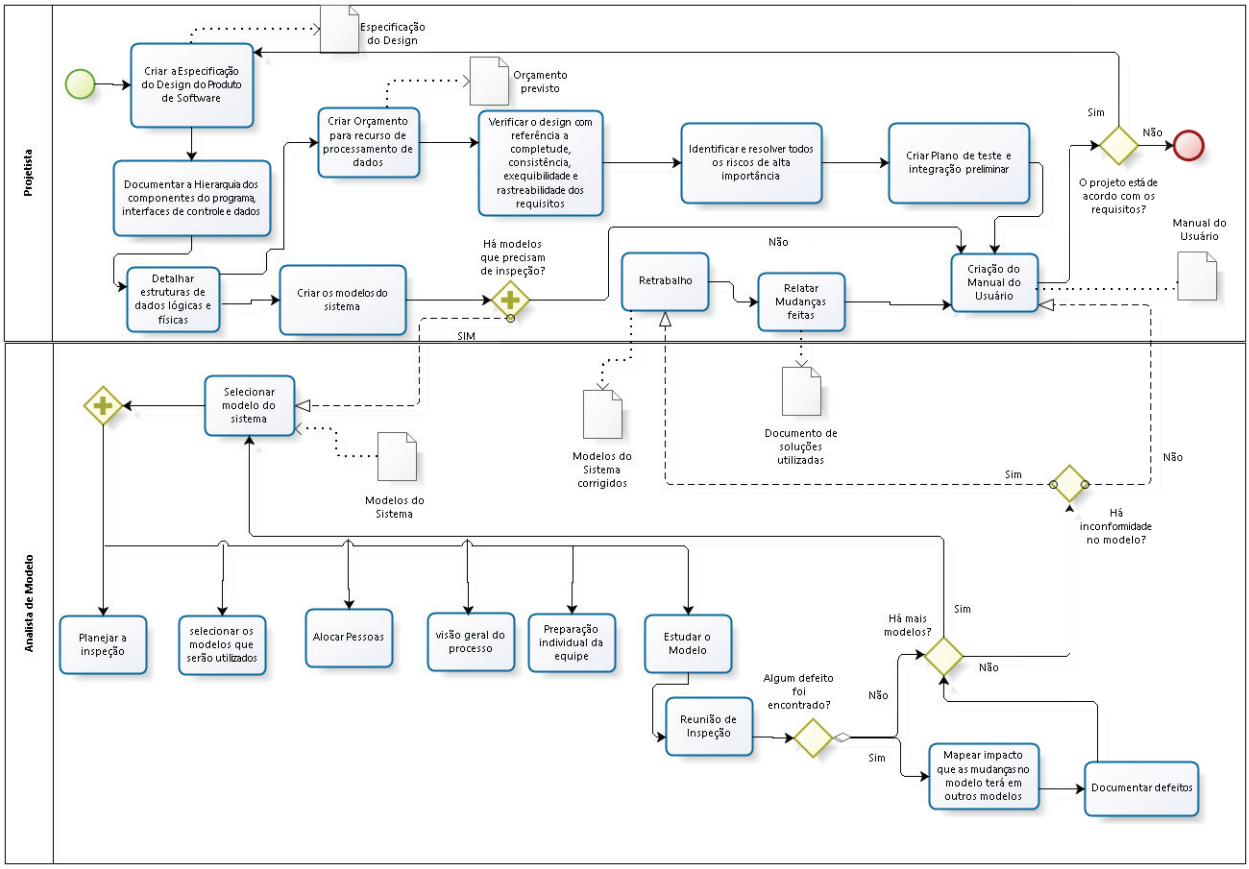

Figura 3. Modelo Cascata com Verificação e Validação na Fase de Projeto

Após as inspeções o projetista deverá ser informado sobre as possíveis inconsistências dos artefatos do sistema e terá a tarefa de corrigi-los. Este processo é reexecutado até que todos os modelos considerados importantes pelo Analista de Modelo sejam verificados. Assim os Modelos de Sistema são refeitos em uma atividade específica, evitando o retrabalho em futuras atividades do processo.

4.2. Analista do Modelo

Para a realização da adaptação do processo é adicionado um novo papel, o Analista de Modelo.

O Analista de Modelo é responsável por selecionar os modelos mais relevantes para o projeto e aplicar a técnica de Inspeção de modelo como as OORT's [Travassos et al. 2002]. Caso seja encontrado algum problema sua responsabilidade é mapear e documentar as possíveis causas dos defeitos e reportar para o Projetista.

O processo de inspeção OORT's de software pode ser aplicado a qualquer tipo de artefato do sistema, incluindo o documento de requisitos. Em relação a Diagramas de Sequência e Casos de Uso, pode-se, por exemplo, verificar se os Diagramas de Sequência descrevem uma combinação apropriada de objetos e mensagens que trabalham em conjunto para capturar a funcionalidade descrita pelo caso de uso. Já com relação a Diagramas de Sequência e Classe é possível, por exemplo, verificar se um determinado Diagrama de Classes descreve as classes e suas associações de forma que os comportamentos espe-

cificados nos Diagramas de Sequência estejam representados corretamente. Mas para que isso possa acontecer, é fundamental o uso de técnicas específicas e adequadas no decorrer de toda Verificação e Validação [Travassos et al. 2002]

5. Conclusão

A atividade de criação de modelos de sistemas não é uma tarefa trivial, ela requer muito cuidado, visto que um defeito produzido em algum modelo pode acarretar em graves problemas na construção do sistema. Acreditamos que técnicas de V&V devem ser aplicadas na fase de projeto com o objetivo de identificar possíveis defeitos de modelagem.

Para isso deve-se exigir mais esforço na etapa de projeto para aplicar técnicas de Verificação e Validação. Essas técnicas estão relacionadas, particularmente, à área de Qualidade de Software, que tem como objetivo auxiliar o time de desenvolvimento a construir produtos com um nível de qualidade aceitável.

De acordo com a nossa visão, essa proposta possui pontos positivos e negativos em relação à integração de Verificação e Validação no modelo cascata.

Os pontos positivos identificados foram cinco, a saber: (1) diminuir a quantidade de retrabalho em etapas seguintes à etapa de projeto; (2) proporcionar maior qualidade na documentação do projeto; (3) mapear problemas detectados com antecedência; (4) garantir maior conformidade dos artefatos; (5) proporcionar maior qualidade no produto final.

Como pontos negativos consideramos que: (1) realizar inspeções sobre o projeto são consideradas caras, embora a longo prazo elas sejam na verdade uma forma de economizar custos de manutenção; e (2) um novo membro da equipe de desenvolvimento deve ser alocado para ser o papel de Analista de Modelo, o que tende a deixar o processo mais custoso em relação ao tempo e orçamento.

Muitas técnicas de Verificação e Validação já são utilizadas nos processos de construção de software. Entretanto, essa proposta tem como objetivo conduzir as técnicas de Verificação e Validação dentro da etapa de Projeto, fazendo com que os modelos de sistema criados nessa fase sejam produzidos de acordo com as boas práticas da Engenharia de Software. Em suma, essa proposta visa reduzir diversos problemas que uma má modelagem pode acarretar em outras etapas do modelo cascata, melhorando a qualidade do processo em geral.

Como forma de validação, pretendemos aplicar essa proposta na Universidade Federal do Pampa (UNIPAMPA) em turmas do curso de Engenharia de Software, particularmente durante as disciplinas de Resolução de Problemas, que são disciplinas em que um problema maior, próximo à realidade do mercado, é apresentado às turmas. Pretendemos tabular os resultados e compará-los com os resultados de turmas que utilizaram o modelo em cascata padrão para determinar se a aplicação do processo de desenvolvimento em cascata com forte ênfase na aplicação de técnicas de verificação e validação na fase de projeto demonstra ser superior ao processo em cascata padrão.

Referências

Bertini, Tereza G. Kirner, M. I. M. I. A. L. (2006). *Técnicas de Inspeção de Documentos de Requisitos de Software: um Estudo Comparativo*. WER06 - Workshop em Enge-

nharia de software.

Boehm, B. W. (1981). *Software Engineering Economics*. Prentice Hall.

Cabral, M. S. (2007). Técnicas de leitura para inspeção da especificação inicial de requisitos.

Debaud, J.M., L. (1990). ”an encompassing life cycle survey of software inspectons”.

Fagan, M. (1976). ”design and code inspections to reduce errors in program development”.

IEEE, C. S. P. (1990). standard glossary of software engineering terminology.

IEEE, C. S. P. (1998). Ieee standard for software reviews and audits.

Jomori, S. M., VOLPE, R., and Zabeu, A. C. P. (2004). Qualidade de software. *Revista TECHOJE,(IETEC-Instituto de Educação Tecnológica), ano XIII. Disponível em:¡ http://www. ietec. com. br/ietec/techoje/techoje/tecnologiadainformacao/2004/07/01/2004_07_01_0001. 2xt/-template_ interna*.

Longen, S. J. (2013). Desenvolvimento de um processo de verificação e validação para produtos de software.

Melo, S. M. (2009). Inspeção de software. *University of São Paulo: São Carlos, SP*.

NASA (1993). Software formal inspections guidebook.

Sommerville, I. (2011). *Engenharia de Software, 9ª edição*. Pearson Education, Inc.

Travassos, G. H., Shull, F., Carver, J., and Basili, V. (2002). Reading techniques for oo design inspections. Technical report.

Tsukumo, A. N., Rêgo, C. M., Salviano, C. F., Azevedo, G. F., Meneghetti, L. K., Costa, M. C., CARVALHO, M. B. d., and Colombo, R. M. (1997). Qualidade de software: visões de produto e processo de software. *II ERI–SBC, Piracicaba, São Paulo, Brasil*.

Wazlawick, R. S. (2012). *Engenharia de Software para Sistemas de informação:Conceitos e Práticas que fazem sentido*. Pearson Education, Inc.