



XVII COLÓQUIO INTERNACIONAL DE GESTÃO UNIVERSITÁRIA
Universidade, desenvolvimento e futuro na Sociedade do Conhecimento

Mar del Plata – Argentina
22, 23 e 24 de novembro de 2017
ISBN: 978-85-68618-03-5



OS DESAFIOS DE UM GESTOR PÚBLICO NO LEVANTAMENTO DE REQUISITOS DE UM SISTEMA DE INFORMAÇÃO

Alexandre Moraes Ramos¹
Universidade Federal de Santa Catarina
E-mail: amrrms@gmail.com

Rodolfo Amando Schmitz²
Universidade Federal de Santa Catarina
E-mail: rodolfoschmitz@gmail.com

RESUMO

Este artigo tem como objetivo apresentar as dificuldades de um gestor público no levantamento de requisitos visando ao desenvolvimento de um software voltado à admissão de novos servidores de uma Instituição Federal de Ensino Superior. Para tanto, foram relacionados Sistema de Informação, além dos aspectos teóricos iniciais e básicos inerentes ao levantamento de requisitos no desenvolvimento de um software, para logo em seguida apresentar as dificuldades no levantamento de requisitos aliadas à teoria, para então apresentar os métodos consolidados utilizados no levantamento de requisitos. Espera-se que este estudo sirva de fundamento para a utilização dos métodos de levantamento de requisitos apresentados visando à melhoria nos processos de desenvolvimento do software em questão. A metodologia utilizada foi aplicada, descritiva, com observação participante natural e pesquisa bibliográfica.

Palavras chaves: Gestão administrativa, sistema de informação, levantamento de requisitos.

¹Professor efetivo do Departamento de Ciências da Administração do Centro Socioeconômico da Universidade Federal de Santa Catarina – CAD/CSE/UFSC. E-mail: amrrms@gmail.com.

²Mestrando do Programa de Pós-graduação em Administração Universitária da Universidade Federal de Santa Catarina – PPGAU/UFSC; Servidor do quadro técnico-administrativo da Universidade Federal de Santa Catarina – UFSC. E-mail: rodolfoschmitz@gmail.com.

1. CONSIDERAÇÕES INICIAIS

Com a constante evolução tecnológica e com o efeito direto na forma como as informações são tratadas, decorrente da necessidade de efetividade nas relações organizacionais, os processos de trabalho são constantemente adaptados também para que se atinjam os objetivos de forma adequada. Para tanto, a aplicação destes se dá por meio dos Sistemas de Informação (SI) de forma a facilitar, organizar e aumentar a segurança dos dados e informações processados (LAUDON e LAUDON, 2010; MANÃS, 2007). Assim, explorar as potencialidades das novas tecnologias se torna um requisito ao serviço público e às Instituições Federais de Ensino Superior (IFES), visto que estas se inserem como partícipes de uma área estratégica no desenvolvimento do país, por sua alta especificidade nas diferentes áreas do saber (GRILLO, 2001).

No que tange à área de gestão admissional de pessoal, a UFSC não possui SI diretamente voltado a esta finalidade. A elaboração de soluções baseadas em tecnologias de informação e comunicação (TIC) auxilia o gestor no aprimoramento dos processos, visando à sua produtividade, eficiência e eficácia e contribui com a tomada de decisão (O'BRIEN; MARAKAS, 2013). Para tanto, torna-se fundamental a identificação dos requisitos funcionais do SI desse tipo, que, antes de tudo, devem estar adequados à gestão dos processos de admissão de pessoal.

Neste sentido, os servidores da Coordenadoria de Admissões, Concursos Públicos e Contratação Temporária (CAC) da Universidade Federal de Santa Catarina (UFSC) identificaram a necessidade de desenvolver um sistema informatizado no qual os servidores ingressantes na instituição realizassem seu próprio cadastro no sistema de gestão de pessoas. Como benefício, então, haveria o aumento da capacidade mensal de ingresso de novos servidores na instituição devido à eliminação de uma etapa do processo de contratação, que é o cadastramento do ingressante. Ademais, exige-se que formulários sejam preenchidos e entregues à CAC no momento prévio à posse, o que, então, reforça a eliminação de um procedimento e não a transferência para outro usuário deste. Ainda, o ingressante teria acesso à matrícula funcional logo após o seu efetivo exercício, diminuindo as dificuldades burocráticas para o início de suas atividades, fatores que justificaram o início do desenvolvimento de um SI.

Assim, os servidores da CAC demandaram ao setor de sistemas e tecnologia da informação da UFSC, o desenvolvimento de um sistema no qual poderia ser acessado por usuários finais com intuito de que estes realizassem o preenchimento dos seus cadastros, gerando formulários e registrando as informações de forma temporária até que a posse e o exercício do ingressante fossem confirmados. Haveria, então, organização das informações e eliminação de parte do processo de trabalho.

Contudo, os servidores da CAC, que são os próprios usuários e gestores setoriais do sistema que seria desenvolvido, não possuíam conhecimento relativo aos processos de desenvolvimento de um sistema de informação, e assim, enfrentaram dificuldades que poderiam ter sido evitadas caso tivessem conhecimento da literatura voltada ao tema. Este desconhecimento foi percebido quando surgiam as inconsistências do sistema e a sinalização da equipe de desenvolvimento que as necessidades e informações mais completas alterariam substancialmente a estrutura do sistema. Ainda, notaram que houve um levantamento de requisitos incompleto e ineficaz.

Assim, objetivou-se abordar neste estudo algumas dificuldades enfrentadas pelos gestores do setor como participantes no desenvolvimento de um SI.

2. PROCEDIMENTOS METODOLÓGICOS

Partindo da análise bibliográfica e ao vivenciado na prática pelo pesquisador quanto às dificuldades em se obter uma adequada definição de requisitos funcionais e não funcionais, e baseado na técnica de observação (GIL, 2008) realizado pelo pesquisador, buscou-se avaliar as dificuldades na concepção de requisitos que atendam de fato a elaboração de um software, e assim, minimizando os custos com a produção deste.

Assim, a pesquisa se caracteriza como aplicada, tendo em vista que será estudada uma situação real e é motivada pela necessidade de estudar problemas concretos com finalidade prática (VERGARA, 2013). Ainda, este estudo é classificado em relação aos fins como descritivo, conforme proposto por Vergara (2013), com observação participante natural, visto que o pesquisador faz parte da equipe estudada (MARCONI e LAKATOS, 2003). Trata-se, ainda, de uma pesquisa bibliográfica.

3 SISTEMA DE INFORMAÇÃO

O’Brein e Marakas (2013) definem sistema como sendo o conjunto de componentes inter-relacionados que produzem um todo organizado, tendo como funções básicas as entrada, processamento e resultado, ou saída, podendo estas serem suplementadas pelas funções *feedback* e controle. Ainda, apontam que a organização, gestão e manutenção dos dados gerados carecem de desenvolvimento de um SI.

Na mesma linha, Laudon e Laudon (2010) apresentam os Sistemas de Informação (SI) sendo compostos por pessoas, organizações e elementos de tecnologia, conforme diagrama apresentado na figura 1. Ainda, destacam que os SI devem apoiar à decisão nas diferentes atividades de uma organização, além de auxiliar a organização na análise de problemas, visualizar assuntos complexos e criar novos resultados.

Figura 1 – Diagrama de composição de um SI



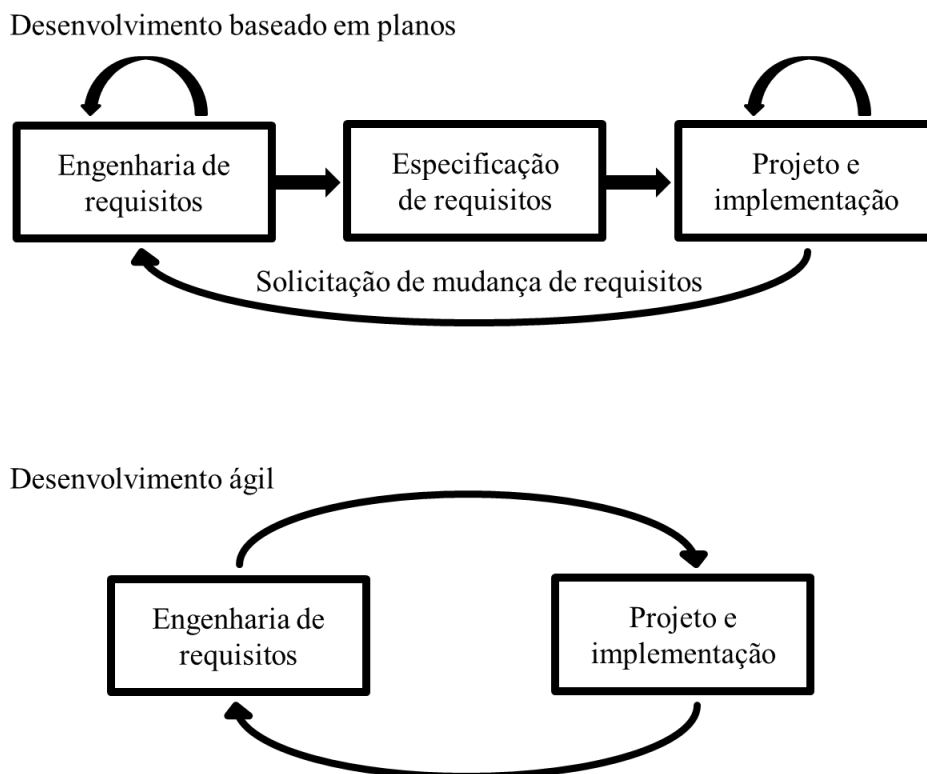
Fonte: Laudon e Laudon (2010).

Contudo, para que um SI seja desenvolvido de forma adequada, é necessário planejamento prévio ao desenvolvimento, sendo estes aspectos tratados pela engenharia de software. Sommerville (2011) define que engenharia de software é uma disciplina de engenharia cujo foco está em cuidar de todas as etapas de desenvolvimento de um software, do estágio inicial à manutenção. A sociedade carece de sistemas de softwares com melhorias na confiança e economicidade no desenvolvimento destes, tendo em vista que a maior parte do custo de um software é relativa à sua alteração após iniciado seu uso (SOMMERVILLE, 2011).

Para a produção de um software, há a utilização de uma abordagem sistemática básica, também chamada de processo de software, para o desenvolvimento do trabalho. Segundo Sommerville (2011, p. 5), “processo de software é uma sequência de atividades que leva à produção de um produto de software”. Além disso, conforme o autor, as quatro atividades fundamentais comuns a todos os processos de software são: 1 – especificação de software, etapa que clientes e engenheiros definem o software e suas restrições; 2 – desenvolvimento de software, etapa em que o software é projetado e programado; 3 – validação de software, na qual se verifica se representa o que o cliente deseja; 4 – evolução de software, caracterizada pela modificação do software para refletir as alterações ambientais.

Entretanto, não há processo ideal, sendo que a maioria das organizações desenvolvem os próprios processos de desenvolvimento de software, havendo, portanto, necessidade de equilíbrio entre métodos ágeis e dirigidos a planos, cuja diferença é caracterizada, principalmente, pela formalidade no desenvolvimento e sequência nos fluxos de trabalho, visto que os requisitos e o projeto são desenvolvidos conjuntamente, conforme se pode observar na figura 2 (SOMMERVILLE, 2011).

Figura 2 – Métodos baseado em planos e métodos ágeis



Fonte: Adaptado de Sommeville (2010, p. 43)

Apesar de não existir processo ideal, Pressman (2010) aponta que uma das etapas fundamentais no desenvolvimento de software é a comunicação no levantamento de requisitos e outras atividades relacionadas entre clientes, interessados e analistas de tecnologia da informação (CARVALHO e CHIOSSI, 2001).

Os requisitos podem ser funcionais, quando voltados à descrição de um serviço ou função, ou não funcionais quando apresentam restrições ao processo de desenvolvimento ou limitação do sistema (CARVALHO e CHIOSSI, 2001). Segundo Martinez e Alves Filho (2011, p.93), “os requisitos funcionais são condições necessárias para a obtenção de certo objetivo, ou para o preenchimento de certo objetivo”, que, em se tratando de SI, caracterizam a funcionalidade do sistema e suas entradas e saídas, isto é, o que o sistema deve fazer, e não como ele deve funcionar. Sommerville (2011) traz que estes requisitos são descritos segundo os usuários, as características organizacionais e o tipo de software a ser desenvolvido. Já os requisitos não funcionais estão ligados às propriedades globais de um software que não são ligadas diretamente aos serviços oferecidos pelo software, como relacionado à segurança, manutenção, desempenho, custo, restrições legais e orçamentárias, interoperabilidade entre sistemas (SOMMERVILLE, 2011; TONSIG, 2008).

O objetivo do processo engenharia de requisitos é levar todas as pessoas envolvidas a um entendimento comum, seja por meio de cenários de usuários, listas de funções e de características, modelos de análise ou outra especificação. Assim, se torna possível uma visão ampla do que será produzido, permitindo que os analistas o contexto do trabalho de software a ser realizado, assim como as necessidades específicas do projeto a serem satisfeitas e, ainda, as prioridades que estabelecem a ordem na qual o trabalho deve ser completado e quais características terão impacto profundo no projeto implantado (PRESSMAN, 2010).

De tal modo, baseado nas rápidas alterações do ambiente externo às organizações, e considerando que os softwares são desenvolvidos visando à atuação em atendimento a estas novas demandas, verifica-se a necessidade do desenvolvimento e entrega mais rápidos destes softwares. Estas mudanças ambientais demandam alterações de requisitos durante o desenvolvimento e implantação do software, visto que é complexo que os clientes ou usuários finais verifiquem como esta nova ferramenta afetará os processos de trabalho de forma detalhada, sua interação com outros sistemas e quais operações devem ser automatizada antes da entrega (SOMMERVILLE, 2011).

Os processos de desenvolvimento ágeis de software são concebidos para produzir softwares úteis, sendo este desenvolvido em várias unidades, como incrementos que incluem novas funcionalidades ao sistema, com a disponibilização de várias versões aos usuários finais que podem fornecer *feedback* rápido sobre a evolução dos requisitos (PRESSMAN, 2010). Sommerville (2011, p. 40) destaca que “métodos ágeis, universalmente, baseiam-se em uma abordagem incremental para a especificação, o desenvolvimento e a entrega do software”. Ainda, traz que o foco está no desenvolvimento do software, objetivando a redução da burocracia da documentação, muitas vezes desnecessária em situações futuras (SOMMERVILLE, 2011).

Dentre às diversas abordagens, Sommerville (2011) destaca algumas características fundamentais: 1 – A especificação, projeto e implementação são intercalados. Não há especificação detalhada do sistema, a documentação do projeto é reduzida ou originada durante a programação do sistema e o documento de requisitos dos usuários apenas definem as características mais importantes; 2 – O sistema é desenvolvido em versões e há a

participação dos *stakeholders*, podendo estes propor alterações dos requisitos em versões futuras; 3 – A interface de usuário do sistema é elaborada em sistemas interativos de desenvolvimento, trazendo rapidez e com possibilidades de implantação em web (SOMMERVILLE, 2011).

Sommerville (2011) define que apesar desses métodos ágeis serem baseados em desenvolvimento e entrega incrementais, eles compartilham um conjunto de princípios, que são: envolvimento do cliente; entrega incremental; pessoas, não processos; aceitar as mudanças; manter a simplicidade.

3.1 Dificuldades no levantamento de requisitos

O levantamento de requisitos consiste em etapa fundamental no desenvolvimento do software, sendo inequívoca a necessidade de clareza, completude, consistência e fácil compreensão por todos os envolvidos no desenvolvimento do software, sejam usuários, clientes ou desenvolvedores. Contudo, esta plenitude na obtenção de requisitos adequados não é simples de se obter, visto que os envolvidos possuem interpretações diferentes sobre os mesmos aspectos (SOMMERVILLE, 2011).

É no levantamento de requisitos que os usuários exploram as suas reais necessidades, diferenciando o que querem do que realmente precisam, incluindo-se as limitações de programação, complexidade e tecnologia disponíveis. Assim, em um trabalho conjunto entre usuários e desenvolvedores, é possível amadurecer os requisitos para que se chegue a um projeto de qualidade (CARVALHO e CHIOSSI, 2001).

Idealmente, o levantamento de requisitos descreve somente o comportamento externo do sistema, sem detalhar a sua arquitetura ou programação, e assim, deve-se utilizar linguagem natural com informações intuitivas no levantamento de requisitos dos usuários (SOMMERVILLE, 2011). Problemas de comunicação decorrentes de vivências e formações distintas podem gerar falhas no desenvolvimento do software, caso não se utilize linguagem clara entre usuários e desenvolvedores (CARVALHO e CHIOSSI, 2001). Na perspectiva prática vivenciada pelo pesquisador, notou-se a utilização ambígua termos, como, por exemplo, servidor, termo atribuído ao trabalhador servidor público ou à máquina servidor de dados. Notou-se, ainda a errônea simplificação e junção de alguns termos que juridicamente são distintos, como nomeação, convocação, posse e efetivo exercício. Isto gerou dificuldades de análise dos requisitos pelos analistas, que não estavam familiarizados com os termos jurídicos e administrativos, e de dificuldades de entendimento pelos usuários do sistema, que desconheciam os jargões utilizados pelos desenvolvedores.

Outra dificuldade é a falta de conhecimento do desenvolvedor do domínio do problema, trazendo tomada de decisão de forma equivocada. É sugerido que os desenvolvedores busquem entender os processos de trabalho e consultem os usuários a fim de evitar tais equívocos (CARVALHO e CHIOSSI, 2001; SOMMERVILLE, 2011). Esta busca ao entendimento dos processos demanda que os desenvolvedores tenham acesso ao mapeamento, descrição ou algum tipo de documentação que registre os processos de trabalho, contudo, estes inexistem no caso prático. Assim, foi necessário explicar oralmente toda a atividade aos desenvolvedores, o que demandou mais tempo, e teve como consequência a inexistência das informações e conflitos posteriores na estruturação e desenvolvimento do sistema.

Ademais, a falta de domínio do processo de levantamento de requisitos pelos desenvolvedores pode fazer com que os usuários participem com menor efetividade no processo, e que posteriormente, os requisitos estejam incompletos e forcem mudanças e demoras que poderiam ser evitados (CARVALHO e CHIOSSI, 2001). Ainda, a falta de registros dos encontros entre usuários e desenvolvedores não gerou memória para verificação e controle posterior das atividades, agravando ainda mais a dispersão dos requisitos e dificultando o acompanhamento das tarefas dos envolvidos, seja por parte dos desenvolvedores ligados ao processo de levantamento ou dos usuários com suas tarefas de pesquisa sobre questões legais ou operacionais.

Igualmente, a complexidade dos processos de trabalho que serão informatizados faz com que os usuários tenham dificuldades em tomar decisões (CARVALHO e CHIOSSI, 2001). A decisão tomada de forma equivocada pode interferir futuramente em processo de outro setor, e ao invés de ter um software que ajude a execução da tarefa, haverá mais uma atividade que não contribui com o trabalho de forma plena.

3.2 Métodos para levantamento de requisitos

Visando a diminuir as dificuldades apresentadas no processo de desenvolvimento de um sistema informatizado, cabe ao gestor do setor e aos desenvolvedores aplicar métodos consolidados para o processo de levantamento de requisitos. Esta descoberta de requisitos é o procedimento para coletar informações sobre os sistemas existentes e o sistema requerido, separando os requisitos entre funcionais e não funcionais. Podem ser utilizadas técnicas como observação, entrevistas, criação de cenários e protótipos para ajudar os *stakeholders* a entenderem como o sistema será (SOMMERVILLE, 2011).

As entrevistas tem papel fundamental na descoberta de requisitos, sendo este o momento em que os *stakeholders* são questionados pela equipe de desenvolvimento sobre o sistema pretendido e a utilização dos sistemas atuais. A entrevista pode ser formal ou informal, e aberta ou fechada, sendo que a principal diferente entre estas é a forma de condução das perguntas aos entrevistados (SOMMERVILLE, 2011). Contudo, a entrevista requer que o entrevistador desenvolva habilidades sociais, ouça, e conheça de táticas de entrevista, além de estarem abertos a novas ideias e estimularem os entrevistados a participarem das discussões visando ao trabalho conjunto de um protótipo (CARVALHO e CHIOSSI, 2001; SOMMERVILLE, 2011).

A criação de cenários, também conhecidos como casos de uso, mostra como o sistema será utilizado pela perspectiva do usuário. Assim, é possível que os usuários compreendam e critiquem um cenário de interação com um software, sendo o debate útil no detalhamento de uma descrição geral de requisitos. Assim, pode-se esperar que os cenários englobem uma descrição do que o sistema e usuários esperam quando o cenário se iniciar, a descrição fluxos dos eventos propostos e as tratativas dos possíveis erros gerados pelo sistema (PRESSMAN, 2010; SOMMERVILLE, 2011).

A etnografia na engenharia de requisitos, por sua vez, é caracterizada como uma técnica de observação que visa ao mapeamento de requisitos implícitos executados nos processos reais e que verificam como as pessoas realmente trabalham. Por meio deste tipo de levantamento, é possível verificar a relação de trabalho entre diversos trabalhadores distintos, incluindo-se fatores sociais e organizacionais em riqueza de detalhes, podendo este tipo de informação ser percebida somente por um observador imparcial (SOMMERVILLE, 2011).

Salienta-se que os métodos para levantamento de requisitos não são mutuamente

excludentes, podendo ser utilizados de forma conjunta visando ao aprofundamento dos detalhes dos requisitos em busca do aprimoramento da construção adequada do sistema pretendido, sendo aplicadas concomitantemente com a implementação do projeto em se tratando de desenvolvimento ágil.

4 CONSIDERAÇÕES FINAIS

Este artigo abordou a caracterização ampla inicial do desenvolvimento de um software e as dificuldades enfrentadas na prática por um gestor público no levantamento de requisitos, apresentando alguns dos métodos comumente utilizados no levantamento destes. Para que este objetivo fosse alcançado, utilizou-se uma abordagem de pesquisa bibliográfica caracterizando SI, engenharia de software, engenharia de requisitos e desenvolvimento ágil e voltado a planos. Os aspectos teóricos apresentados serviram como base para compreensão da necessidade de que se estruturam de forma adequada os requisitos de um software e de quais formas é possível aprimorar estes levantamentos.

Assim, foi possível analisar algumas causas das falhas cometidas no levantamento de requisitos e no desenvolvimento conjunto do software em questão, geradas pelo desconhecimento de alguns dos envolvidos no desenvolvimento do software. Como principais causas de falhas, foram apontadas a falta de comunicação adequada, sem utilização de linguagem natural e sem informações intuitivas no levantamento de requisitos. Ainda, a falta de conhecimento dos desenvolvedores relativos aos processos de trabalho e a falta de detalhamento destes processos gerou a necessidade de explicação oral aos desenvolvedores, o que causou a inexactidão das informações e conflitos posteriores no desenvolvimento do sistema. Ademais, a falta de registro dos encontros entre usuários e desenvolvedores não gerou memória para verificação e controle posterior das atividades, ocasionando a dispersão do foco do desenvolvimento. Outro fator apontado é a dificuldade de os usuários na tomada de decisão pela falta de conhecimento do impacto do sistema em outros setores.

Como proposta de redução destas dificuldades, foram apresentados métodos consolidados no processo de levantamento de requisitos, como entrevistas, observação, criação de cenários e etnografia, visando ao desenvolvimento adequado de software, seja pelo desenvolvimento ágil ou dirigido a planos, podendo estas técnicas serem utilizadas concomitantemente.

Portanto, a melhoria nos processos de levantamento de requisitos constituem pontos fundamentais no desenvolvimento de software, que deve ser desenvolvido em conjunto entre clientes, usuários e desenvolvedores para que se obtenha efetividade na utilização deste e aprimore os processos de trabalho requeridos.

REFERÊNCIAS

CARVALHO, Ariadne Maria Brito Rizzoni; CHIOSSI, Thelma Cecília dos Santos.

Introdução à engenharia de software. Campinas: Editora UNICAMP, 2001.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Atlas, 2008.

GRILLO, Antônio Niccoló. **Gestão de pessoas: princípios que mudam a administração universitária.** Florianópolis: UFSC, 2001

LAUDON, Keneth; LAUDON, Jane. **Sistemas de informações gerenciais**. São Paulo: Pearson Prentice Hall, 2010.

MAÑAS, Antonio Vico. **Administração de sistemas de informação: como organizar a empresa por meio dos sistemas de informação**. 7. Ed., São Paulo, Érica, 2007.

MARCONI, Marina de Andrade, LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. São Paulo: Atlas 2003.

O'BRIEN, James A.; MARAKAS, George M. **Administração de sistemas de informação**. Porto Alegre: AMGH, 2013.

PRESSMAN, Roger S. **Engenharia de Software**. 6. Ed. Porto Alegre: AMGH, 2010.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. Ed. São Paulo: Pearson, 2011.

TONSIG, Sérgio Luiz. **Engenharia de software – Análise e Projeto de Sistemas**. 2 Ed. Rio de Janeiro: Ciência Moderna, 2008.

WAZLAWICK, Raul Sidnei. **Engenharia de software: conceito e práticas**. Rio de Janeiro: Elsevier, 2013.