OS BENEFÍCIOS DA INTEGRAÇÃO DA ENGENHARIA DE SOFTWARE E DA INTERAÇÃO HUMANO-COMPUTADOR NO DESENVOLVIMENTO DO SOFTWARE CATALÓG

THE BENEFITS OF INTEGRATION OF SOFTWARE ENGINEERING AND HUMAN--COMPUTER INTERACTION IN SOFTWARE DEVELOPMENT CATALÓG

Daniela Gibertoni[1]
Thaís Cristina Casagrande[2]

RESUMO

O objetivo deste artigo é relatar os benefícios da integração da Engenharia de Software (ES) com as práticas de Interação Humano-Computador (IHC) para a construção de um software denominado CataléG (Catálogo de Revistas de Gêneros), que foi desenvolvido pelo Grupo de Pesquisa em Engenharia Software – GPES – da Faculdade de Tecnologia de Taquaritinga. O trabalho traz as fundamentações teóricas acerca das metodologias e processos de software tanto da área de ES quanto de IHC bem como apresenta a contribuição do design participativo para que essas práticas possam ser fundidas a fim de produzir um software que atenda ambas as áreas.

Palavras-chave: Modelos de processos de software. Engenharia de software. Interação humano-computador. Design participativo.

ABSTRACT

This paper aims to report the benefits of the integration of Software Engineering (SE) and the practices of Human-Computer Interaction (HCI) for the construction of a software called CataléG (Catálogo de Revistas de Gêneros), which was developed by Grupo de Pesquisa em Engenharia Software, at Faculdade de Tec-

1. Docente da Faculdade de Tecnologia de Taquaritinga e Coordenadora do Grupo de Pesquisa em Engenharia de Software – GPES. E-mail: daniela.gibertoni@fatectq.edu.br
2. Discente da Faculdade de Tecnologia de Taquaritinga. E-mail: thais.casagrande@fatectq.edu.br

nologia de Taquaritinga. The work presents the theoretical foundations related to the methodologies and processes of SE and HCI , it also shows the contribution of participatory design for the merge of these practices in order to produce a helpful software for both the fields.

Keywords: Software processes models. Software engineering. Human-computer interaction. Participatory design.

INTRODUÇÃO

Duas áreas da Ciência da Computação se destacam no desenvolvimento de sistemas interativos: a Engenharia de Software (ES) e a Interação Humano-Computador (IHC). Entretanto, essas áreas divergem quanto ao foco dado durante o desenvolvimento de tais sistemas. Segundo Brown (1996), enquanto o foco da ES está mais voltado para as funcionalidades, o de IHC está voltado para os aspectos da interação entre o ser humano e a máquina.

Apesar dessas duas áreas divergirem quanto ao foco, ambas propõem o desenvolvimento de sistemas interativos de forma sistemática, definindo modelos de processo, métodos e técnicas. Pode-se citar como modelos de processo propostos pela ES: o Cascata (SOMMERVILLE, 2007), o Incremental, a Prototipação, o Espiral, o *Rapid Application Development* – RAD - (PRESSMAN, 2011), o *Rational Unified Process* – RUP – (KRUCHTEN, 2000) e os métodos ágeis como o *Extreme Programming* (XP). Como modelos de processo propostos pela IHC: o Projeto Centrado no Usuário (PCU) (PREECE et al, 2005), o Estrela (HIX e HARTSON, 1993), a Engenharia de Usabilidade (NIELSEN, 1993) e o Design Participativo (DP) (DIX, 1998).

Embora os modelos de processo, métodos e técnicas expressem o foco da área que a propõe, percebe-se que é possível desenvolver sistemas interativos considerando práticas de ambas as áreas através da consideração dos aspectos relevantes à ES em conjunto aos considerados pela IHC, objetivando o desenvolvimento de sistemas de forma mais abrangente. Logo, considerando os aspectos dessas duas áreas durante o desenvolvimento, é possível construir sistemas que não somente sejam de fácil manutenção, e que satisfaçam o usuário quanto ao prazo de entrega e ao custo, mas que também o tornam mais

confiável e de fácil utilização pelo usuário. Esses resultados são desejáveis, pois o mercado está se tornando cada vez mais competitivo, sendo a interface com o usuário um diferencial de produto (FERRÉ, 2003).

Baseado neste raciocínio, este artigo tem como objetivo relatar os benefícios da integração dos processos oriundos da ES com os modelos de processo de software da área de IHC para o desenvolvimento do software CatalóG (Catálogo de Revistas de Gênero), bem como os processos e atividades desenvolvidas ao longo do projeto. Este software foi desenvolvido pelo Grupo de Pesquisa em Engenharia de Software – GPES – da Fatec de Taquaritinga. Como um dos grandes desafios da área de desenvolvimento é trabalhar e realizar um produto ou software que atenda a todas as especificações dos usuários e ao mesmo tempo tenha um bom design, este artigo mostra o caminho para superar esse desafio, através da metodologia do design participativo.

1. Modelos de Processos em Engenharia de Software

A ES é uma disciplina de engenharia relacionada com todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação. A ES, juntamente com a Engenharia de Requisitos, tem o objetivo de utilizar as melhores técnicas e métodos para a construção de um software confiável que esteja compatível com as necessidades dos usuários (SOMMERVILLE, 2007).

Portanto, produzir um software é oferecer uma solução viável para um problema existente de forma planejada e válida e que resulte em um produto final satisfatório e de qualidade.

Para que o desenvolvimento de um software seja possível, é necessário que seja planejado (modelagem e projeto de software), executado (geração de código) e avaliado (testes e garantia de qualidade). O planejamento de software envolve modelos de processos de software a serem seguidos para que seja desenvolvido evitando-se custos desnecessários e tempo de desenvolvimento maior que planejado.

Um processo de software é composto por modelos que, por sua vez, envolvem um conjunto de tarefas que tem função de nortear o processo de desenvolvimento. Os objetivos de um processo de software incluem redução de custos e aumento de

qualidade e de produção, portanto um processo de software que não atenda esses objetivos de forma satisfatória, não é um processo viável (MEDEIROS, 2013).

Apesar de os métodos ágeis ganharem cada vez mais repercursão, os modelos prescritivos ainda estruturam vários projetos. Responsáveis por definir um fluxo de trabalho de processo previsível, são denomidados assim por prescreverem um conjunto de elementos de processo (atividades metodológicas, ações de ES, tarefas, produtos de trabalho, garantias de qualidade e mecanismos de controle de mudanças para cada projeto). Todos os modelos de processo de software podem acomodar atividades metodológicas com ênfase em pontos diferentes e definir fluxo de processos e ações de ES (PRESSMAN, 2011).

Os modelos prescritivos contemplam desde o modelo cascata até o modelo evolucionário (prototipação e espiral). Neste artigo é abordado de forma mais precisa os modelos prototipação e incremental, pois juntamente com modelos de ciclo de vida em IHC, que serão analisados mais a frente, proporcionaram a base para o modelo utilizado na construção do software CataláG.

1.1. Prototipação

Segundo Sommerville (2007), um protótipo é uma versão inicial de um sistema de software usada para experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções.

As fases que compõem a prototipação vão desde a definição dos objetivos do protótipo até sua avaliação, passando pela definição das suas funcionalidades e seu desenvolvimento.

Embora a prototipação possa ser utilizada como modelo de processo isolado, é mais comumente utilizada como uma técnica passível de ser implementada no contexto de qualquer um dos modelos de processo (PRESSMAN, 2011).

A prototipação pode ser utilizada como aliada no processo de desenvolvimento pelo fato dos desenvolvedores poderem usá-la como parâmetro para produzir uma versão final de alta fidelidade com os requisitos levantados, além de proporcionar aos usuários um nível de experiência com o uso do software, podendo, ao mesmo tempo, avaliar as experiências dos usuários e contribuir para a construção de um sistema interativo.

1.2. Entrega incremental

Na iteração de processos de ES, existe a entrega incremental, onde o projeto é dividido em etapas e elas são desenvolvidas uma por vez. Os usuários definem as atividades a serem entregues de acordo com o grau de necessidade. Para Pressman (2011), as atividades que compõem esse modelo são: a comunicação, o planejamento, modelagem (análise e projeto), construção (codificação e testes) e o emprego (entrega, realimentação ou feedback).

A entrega incremental do desenvolvimento de software foi formulada por Watts Humphrey como um conjunto de regras para as equipes de gestão do software seguirem durante as fases de especificação de requisitos de projeto de um processo de software (PETER & PEDRYCZ, 2001). É uma abordagem intermediária que combina as vantagens dos modelos evolucionário (prototipação) e o cascata (SOMMERVILLE, 2007).

Segundo o mesmo autor, o processo incremental possui uma série de vantagens:

1. Os usuários não precisam esperar até a entrega do sistema inteiro para se beneficiarem dele.
2. Os incrementos iniciais podem ser usados como protótipos e servirem para que os usuários ganhem experiência e assim dar informações sobre os próximos requisitos a serem incrementados no sistema.
3. Menor risco de falha geral no projeto.
4. Os serviços mais importantes do sistema recebem mais testes, portanto a possibilidade de encontrar falhas no produto final é menor.

Ao se observar a segunda vantagem apresentada, pode-se dizer que a prototipação e o modelo incremental em determinado momento se fundem, trazendo à equipe de desenvolvimento melhores condições de se apropriar dos reais requisitos que os usuários requerem, bem como aproximá-los por meio do modelo de processo Estrela, apresentado na sequencia.

2. Modelos de processos em Interação Humano-Computador

Segundo Rocha e Baranauskas (2003), a IHC tem foco no projeto, avaliação e implementação de um sistema interativo para o uso humano, além de se preo-

cupar com o estudo dos principais fenômenos ao redor dele, como a área de aplicação e o contexto de uso do usuário.

Dentro da área de IHC, existe o design de interação, que segundo Cybis (1999) pode ser descrito como o design de produtos interativos que fornecem suporte às atividades cotidianas das pessoas, seja no lar ou no trabalho. Uma parte do processo de entender as necessidades do usuário, no que diz respeito a projetar um sistema interativo que as atenda, consiste em ser claro quanto ao objetivo principal. Existem duas metas a serem seguidas para que se possa construir um sistema interativo com alto grau de fidelidade: as metas de usabilidade e as metas decorrentes da experiência do usuário (PREECE *et. al*, 2005).

No que tange às metas de usabilidade, pode-se defini-la como a característica de um sistema ser fácil de usar, além de serem eficientes, seguros, fáceis de serem aprendidos e esteticamente agradáveis. Com relação às metas de experiência do usuário, os sistemas devem ser úteis, motivadores, e atender as necessidades dos usuários de forma satisfatória.

A partir dessas metas, surge a necessidade de se definir a forma de trabalho, os processos e modelos sobre os quais o software será desenvolvido. A escolha de um modelo de desenvolvimento afeta a forma como o software irá trabalhar, bem como o tempo de desenvolvimento, forma como será documentado e recursos disponíveis.

No modelo de design, o software sempre começa a ser desenvolvido de acordo com as necessidades e requisitos levantados pelo usuário, a partir desse ponto ocorre a validação dos mesmos, e logo após ocorre o processo de design, que têm o objetivo de construir designs interativos de acordo com os requisitos levantados. Caso o software não tenha saído de acordo com os requisitos, ou não tenha atendido satisfatoriamente as necessidades dos usuários, ele pode passar por um processo de re-design até atingir seu objetivo. Uma versão interativa é capaz de suprir as necessidades dos usuários e apoiá-los nas atividades rotineiras, facilitando o uso do software. O produto final deve ser proveniente dessas atividades e ao final, passar sempre pela avaliação.

Assim como na ES, o campo da IHC possui seus modelos de ciclo de vida. O modelo Estrela (Star) derivou do trabalho empírico de entender como os designers lidavam com problemas de design em IHC. Por outro lado, o ciclo de vida da Engenharia de Usabilidade apresenta uma abordagem mais estruturada e descende da tradição da própria Engenharia de Usabilidade (PREECE et .al., 2005).

2.1. Modelo estrela

Composto pelas fases de implementação, análise das tarefas/análise funcional, requisitos/especificação, projeto conceitual/ representação formal do design e prototipação, este modelo foi proposto por Hix e Hartson (HIX e HARTSON, 1993). O ciclo de vida Estrela não especifica ordenamento algum das atividades, elas são altamente interconectadas, pode-se passar de uma atividade a outra qualquer desde que se passe primeiro pela atividade de avaliação. No que diz respeito a esse modelo, a avaliação é central, sempre que uma atividade for completada, seu resultado deverá ser avaliado (PREECE et al, 2005).

Para Rocha e Baranauskas (2003) as atividades do modelo estrela são semelhantes às atividades do modelo cascata, porém a avaliação é central e o início do processo pode acontecer em qualquer uma das demais atividades.

2.2. Ciclo de vida da Engenharia de Usabilidade

Para Rocha e Baranauskas (2003), Engenharia de Usabilidade é o termo que se usa para definir o processo de design de sistemas computacionais que objetivam a facilidade de aprendizado, de uso, e que sejam agradáveis para as pessoas. Para as mesmas autoras, o ciclo de vida da Engenharia de Usabilidade é composto por quatro fases que incluem o pré-design, o design inicial, desenvolvimento iterativo e pós-design.

Para Mayhew (1999) *apud* Preece (2005), a Engenharia de Usabilidade compreende três fases: análise de requisitos, projeto/teste/desenvolvimento e instalação desdobrando-se em estágios de identificação de requisitos, projeto, avaliação e construção de protótipos.

2.3. Design participativo no processo de desenvolvimento e levantamento de requisitos

O Design Participativo (DP) foi desenvolvido inicialmente na Escandinávia e provê uma série de técnicas para conduzir o design com os usuários e não para o usuário. A abordagem proposta no DP enfatiza a importância da democracia no ambiente de trabalho para aprimorar os métodos de trabalho, a eficiência no

processo de design (através da experiência e comentários dos usuários), aprimoramento da qualidade do sistema, e a condução das atividades formativas. Em DP estes objetivos são atingidos através da interação direta dos usuários com designers durante todo o ciclo de desenvolvimento, e pelo controle do usuário sobre as decisões de design.

As atividades conduzidas durante o DP visam o desenvolvimento de uma visão compartilhada da tecnologia e da organização, explorando novas estruturas organizacionais, requisitos dos sistemas e protótipos de novos sistemas.

A colaboração dos usuários durante todo o processo de design provê as informações necessárias para os designers. Além disso, as atividades de DP aprimoram a qualidade do sistema resultante por compreender melhor o trabalho desenvolvido pelo usuário e combinar diferentes conhecimentos dos participantes durante o processo de design (BRAA, 1996).

No design participativo, diferente do design contextual, os usuários são ativamente envolvidos no desenvolvimento (PREECE et. al., 2005). A intenção consiste em fazer com que se tornem um parceiro como os outros na equipe de design, projetando o produto em cooperação com os designers.

Esta colocação é muito importante dado que, para o desenvolvimento do software Cataló G foi utilizado os princípios do DP, ou seja, a equipe responsável pelo software foi composta por três usuários do sistema, um analista de sistemas, um administrador de banco de dados, dois programadores e um designer. Está descrito no próximo item como ocorreu a integração bem como o trabalho foi realizado.

3. Integração de Engenharia de Software e Interação Humano-Computador para o Software Cataló G

O Cataló G tem como base uma pesquisa científica para auxiliar sua professora pesquisadora em seu projeto. Por se tratar de um tema de pesquisa na área de estudos de gênero é que o Grupo de Pesquisa em Engenharia de Software – GPES - da Faculdade de Tecnologia de Taquaritinga desenvolveu e nomeou o software em Cataló G: Catálogo Digital de Revistas de Gênero.

Desta forma, o software Cataló G tem como objetivo mapear os diferentes estudos realizados pela comunidade científica nacional na área de estudos de gênero e feministas nos últimos 20 anos.

O software CataléG foi construído com base no modelo evolucionário, especificamente a prototipação, – segundo Pressman (2011), essa técnica pode ser implementada em qualquer modelo de processo - utilizando do desenvolvimento exploratório, que tem o objetivo de explorar os requisitos e assim contribuir com a evolução do sistema até chegar a uma versão final. O uso desse modelo foi fundamental para que a equipe técnica pudesse entender mais claramente os requisitos propostos pelos usuários, uma vez que a área estudada foi Ciências Sociais, em Estudos de Gênero. Desta forma, a prototipação aliada ao DP promoveu uma maior facilidade para a obtenção de requisitos.

A partir desse modelo de processo, o GPES pôde se adaptar e construir seu próprio processo de desenvolvimento, de acordo com as características dos usuários e dos desenvolvedores. Pode-se dizer então, que o CataléG se beneficiou de práticas da ES e da IHC no seu processo de definição de escopo e desenvolvimento.

Esse modelo de processo foi escolhido de acordo com a necessidade de entregar um software que proporcionasse uma experiência inicial aos usuários, que até então faziam suas pesquisas sem uso de software, de forma manual. Através da utilização da versão inicial, mais requisitos foram levantados e validados e depois implementados e testados novamente pelos usuários.

A entrega das funcionalidades foi feita de forma incremental, assim os usuários não precisavam esperar que o sistema estivesse terminado para testar os incrementos adicionados (SOMMERVILLE, 2007), isso contribuiu para a fase de testes, pois a cada novo incremento disponibilizado, novos testes de uso eram feitos pelos próprios usuários, e os erros reportados e sanados antes desse incremento partir para a versão final.

Além dos modelos de processos de ES, o sistema foi desenvolvido com base no modelo de design de interação proposto por Preece (2005) para garantir que o software fosse interativo e auxiliasse as atividades da pesquisadora de forma eficiente.

Como pode ser observado na Figura 1, o CataléG começou a ser construído a partir da identificação dos requisitos propostos pelos usuários. O levantamento de requisitos aconteceu por meio de reuniões, onde os mesmos expunham suas atividades e o que esperavam de um software que contribuísse para automatização dos processos de trabalho.

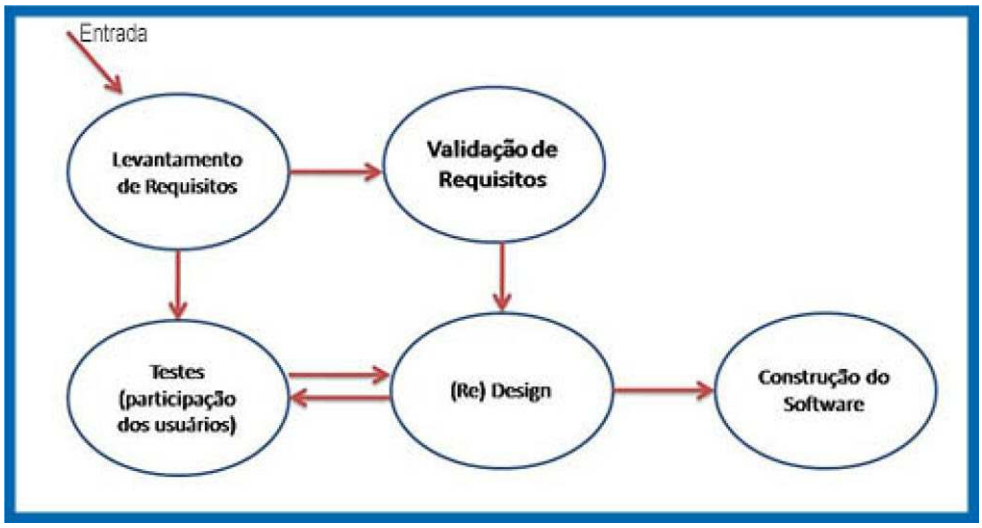

Figura 1. Processo de construção do Software CatalóG.
Fonte: Elaborado pelos autores.

Após o levantamento, os requisitos passavam por um processo de validação. O objetivo principal da validação é avaliar até que ponto o requisito proposto é viável e como deve ser implementado.

No momento em que os requisitos são validados, eles passam para o processo de design, onde o software é desenhado de forma interativa, com a participação dos usuários no processo de testes. Se um requisito não foi bem compreendido ou foi implementado de maneira errada, é nessa etapa que os erros são corrigidos por meio de um processo de re-design e possivelmente um requisito novo é levantado ou a especificação de um existente é avaliado e compreendido com a ajuda dos usuários.

Desta forma, percebe-se que os usuários se mantiveram ativos durante todo o processo de levantamento de requisitos e de avaliação das funcionalidades. A participação deles foi fundamental para que o software fosse construído com base em suas reais necessidades e pudesse oferecer informações confiáveis baseadas nos dados cadastrados. Isto foi possível em decorrência do design participativo, pois consiste em envolver o usuário e trabalhar com ele como parceiro ao longo de todo ciclo de design e desenvolvimento do projeto a fim de se obter um software usável e de acordo com as especificações do usuário.

Essa metodologia foi escolhida com base nas áreas envolvidas no projeto, pois havia necessidade de abstrair conhecimentos de sociologia para a área

de ES para que o desenvolvimento fosse possível. Foram utilizadas técnicas como *brainstorming* para o levantamento de requisitos e reuniões para manter os usuários do software ativos durante o processo de desenvolvimento.

Depois de seguidas essas etapas começaram-se a codificação e documentação do software. Esse processo foi seguido pelos desenvolvedores do CatalóG, para oferecer como resultado final além das funcionalidades requeridas, um software interativo e margem de erros próxima ao zero.

CONCLUSÃO

Este artigo expôs os benefícios da integração dos modelos de processos de software oriundos da ES e os processos de IHC no desenvolvimento do software CatalóG. Um dos aspectos que deve ser ressaltado é que esta integração não é uma tarefa fácil embora traga benefícios ao processo de desenvolvimento, principalmente devido às divergências de focos e de formação dos profissionais. Para transpor essa dificuldade, é necessário que ocorra uma identificação mais detalhada das tarefas a serem realizadas durante o processo de desenvolvimento, não se limitando somente a responsabilizar o engenheiro de software pela parte tecnológica e o especialista em IHC pela interação e interface com o usuário. Essa divisão não é suficiente, pois aspectos de interação e tecnologia se influenciam mutuamente.

Outra questão a ser considerada na integração entre a ES e a IHC é a necessidade de uma boa comunicação entre os especialistas das duas áreas durante o desenvolvimento do sistema. A falta de comunicação influencia no produto construído, como por exemplo, a adição de funcionalidades não necessárias, código de difícil compreensão e manutenção.

A utilização do modelo de prototipação atrelado à prática de design participativo trouxe benefícios tanto para os usuários do software quanto para os desenvolvedores. Em relação aos usuários, essas práticas proporcionaram maior grau de envolvimento no processo de desenvolvimento do software, possibilitando assim a construção de um software de acordo com suas necessidades. Com relação ao desenvolvimento, essas práticas possibilitaram maior compreensão dos requisitos por parte dos desenvolvedores, além de permitir testes constantes sempre que uma funcionalidade é disponibilizada. As entregas foram feitas de

forma incremental, disponibilizando para uso as funcionalidades mais importantes em primeiro lugar, e assim sucessivamente até se chegar a uma versão final com todas as funcionalidades constantes nos requisitos.

Conclui-se, portanto, que apesar de serem tratadas como áreas diferentes, a Engenharia de Software e a Interação Humano–computador se trabalhadas em conjunto podem trazer vários benefícios como ganho de tempo, produtividade, maior grau de abstração de requisitos como constatado durante o desenvolvimento do CatalóG.

REFERÊNCIAS

BRAA, K. Influencing qualities of informantion systems – Future challenges for participatory design. In: **PDC´96** Proceedings of the Participatory Design Conference, 163-172, 1996.

BROWN, J. **Methodologies for the Creation of Interactive Software**. Victoria University of Wellington, Wellington, Nova Zelândia, Tech. Rep. CS-TR-96/1, May. 1996.

CYBIS, W. D., PIMENTA, M. S., SILVEIRA, M. C., GOMEZ, L. **Unicamp,** [s.d.]. Disponível em: <http://www.unicamp.br/~ihc99/Ihc99/AtasIHC99/AtasIHC98/Cybis.pdf>. Acesso em: 03 jun. 2014,

DIX, A. J., FINLAY, J. E., ABOWD, G. D., BEALE, R. **Human-Computer Interaction**. 2.ed. England: Prentice Hall Europe, 1998.

FERRÉ, X. **Approaches to HCI Integration into Software Emgineering Processes**: How much we still need to do. In: Proc. Workshop WIHC-ES. In CLICH, 2003.

HIX, D., HARTSON, H. R. **Developing User Interface**: Ensuring Usability Through Product & Process. USA: Wiley & Sons Inc., 1993.

KRUCHTEN, P. **The Rational Unified Process – An introduction**. 2.ed. Nova Jersey: Addison-Wesley Pub. Co., 2000, p. 320.

MEDEIROS, H. **Introdução aos Processos de Software e o Modelo Incremental e Evolucionário**. Disponível em: <http://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>. Acesso em: Acesso em: 17 ago. 2014.

NIELSEN, J. **Usability Engineering**. California: Academic Press, 1993, p. 362.

PETER, J. F., PEDRYCZ, W. **Engenharia de Software Teoria e Prática**. Rio de Janeiro: Campus, 2001.

PREECE, J., ROGERS, Y., SHARP, H. **Design de Interação**: além da interação homem-computador. Bookman, 2005.

PRESSMAN, R. S. **Engenharia de Software**: uma Abordagem Profissional. Porto Alegre: AMGH Editora Ltda, 2011.

ROCHA H.V., BARANAUSKAS. C. **Design e Avaliação de Interfaces Humano-Computador**. São Paulo: Unicamp, 2003.

SOMMERVILLE, I. **Engenharia de software**. Addison Wesley, 2007.