

# XPu: uma extensão do método XP visando à usabilidade

Thiago Rocha Silva e Clarindo I. P. S. Pádua<sup>1</sup>

<sup>1</sup> Universidade Federal de Minas Gerais, Departamento de Ciência da Computação,  
Belo Horizonte, MG - Brasil  
{thiagors, clarindo}@dcc.ufmg.br

**Abstract.** This paper describes a method called XPu. The XPu method was proposed as a XP extension and it has as main goal to deal with software products usability through an agile discipline. Our method was evaluated through a case study and the results suggest a considerable quality improvement in the end interface with a relatively low development cost.

**Keywords:** Agile Methods, eXtreme Programming, Usability Engineering.

## 1 Introdução

Os métodos ágeis de desenvolvimento de software surgiram no final da década de 90 como uma alternativa ao desenvolvimento de software tradicional, até então referência no mercado. Em 2001, um grupo de importantes desenvolvedores de software se reuniu em prol de um Manifesto Ágil [1], formalizando a filosofia de desenvolvimento até então utilizada por eles, com foco no cliente e nas pessoas envolvidas, na entrega constante de software funcional, no desenvolvimento iterativo baseado em ciclos curtos de entrega e na aceitação da constante mudança dos requisitos ao longo do desenvolvimento.

Kent Beck [3] propôs um dos mais importantes conjuntos de práticas e valores relacionados às idéias do movimento de desenvolvimento ágil denominado XP (*eXtreme Programming* ou Programação eXtrema). No entanto, o XP e os demais métodos ágeis não tratam das práticas relacionadas à usabilidade [6]. Lee [17] afirma que para o XP e para os demais métodos ágeis, ainda não está claro como incorporar as práticas de usabilidade sem sacrificar os benefícios providos pelo desenvolvimento ágil de software.

Segundo a norma ISO 9241 [15], a usabilidade é definida como a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso. Já a norma ISO 9126 [14] define a usabilidade como um dos atributos da qualidade de software. Logo, tratar da usabilidade ao longo dos processos de desenvolvimento de software é uma importante atividade que deve ser desempenhada durante toda a execução do processo, visando à qualidade do produto final. Diante desse cenário, torna-se necessário especificar práticas de usabilidade que possam se integrar aos métodos ágeis, sem prejuízo das características que permitem classificá-los como tal.

Este artigo, portanto, apresenta uma extensão do método XP denominada XPu, cujo principal objetivo é propor uma solução ao cenário descrito. A definição do método XPu surgiu da adaptação do processo de usabilidade Praxis-u [24] ao contexto ágil de desenvolvimento de software. A extensão proposta foi inspecionada em relação à manutenção das características ágeis do método XP e seus resultados estão publicados em [29]. Com o propósito de verificar a instanciação do método em um projeto real de desenvolvimento de software, este artigo também apresenta a avaliação do método XPu em um estudo de caso com dois profissionais no desenvolvimento de um sistema interativo de busca de mídias para uma locadora de DVDs.

As seções seguintes desse artigo apresentam os trabalhos relacionados e uma breve descrição do método XPu em suas três visões. Por fim, são apresentados os resultados do estudo de caso realizado, bem como os trabalhos futuros relacionados ao método descrito, alguns deles, inclusive, já em execução atualmente.

## 1.1 Trabalhos Relacionados

O crescente aumento no número de publicações em periódicos e conferências especializadas da área mostra o claro interesse da comunidade por iniciativas no sentido de entender o processo de integração dos métodos ágeis com as atividades de usabilidade. O trabalho de Constantine [6] foi um dos primeiros a iniciar a discussão sobre o tema. Seu trabalho propõe uma variante simplificada de UCD (*Usage-Centered Design*) e discute algumas práticas que podem se integrar aos métodos “leves”, ditos ágeis.

A partir do trabalho de Constantine, outros trabalhos foram publicados com objetivos diversos, porém quase sempre complementares. Alguns deles [7, 12, 13, 18, 25, 30, 31, 33] propõem um processo ou atividades isoladas de usabilidade focadas em métodos ágeis. Alguns deles partem de modelos bem definidos, porém sem apresentar claramente a integração das atividades de usabilidade com as atividades de desenvolvimento. Outros não tratam das atividades gerenciais de um processo de usabilidade ou tratam de aspectos específicos, como o trabalho de Meszaros e Aston [21] que está focado na discussão de testes de usabilidade para métodos ágeis.

Outros trabalhos [2, 9, 10, 16, 20] se concentram em discussões mais genéricas sobre como adaptar as práticas e atividades de usabilidade em um contexto ágil, sem propor, de fato, um método ou processo. O trabalho de Lievesley e Yee [19], inclusive, foca a discussão apenas no papel esperado do Projetista de Interação em um método ágil. Um terceiro e último grupo de trabalhos [5, 11, 22, 32] discute estudos de caso na indústria e em projetos específicos sobre a adoção de práticas de usabilidade em equipes ágeis.

O método XPu aqui apresentado se insere no primeiro grupo de trabalhos ao propor uma adaptação do processo Praxis-u ao contexto de desenvolvimento do método XP. Essa característica faz com que este trabalho tenha pontos de convergência com todos os trabalhos deste grupo, no entanto se difere de todos eles ao apresentar uma análise crítica sobre a caracterização de agilidade da proposta [29], confrontando-a com uma experiência prática de desenvolvimento. Além disso, o XPu ao adaptar as atividades do Praxis-u manteve suas atividades gerenciais, característica negligenciada por grande parte dos demais trabalhos desse grupo.

## 2 O método XPu

Esta seção descreve brevemente o método de usabilidade proposto para o XP. O XPu (*eXtreme Programming with usability* ou Programação eXtrema com usabilidade) [29] foi definido a partir do processo de usabilidade Praxis-u, proposto por Pádua *et al.* [24], utilizado para ensino e pesquisa no Departamento de Ciência da Computação da Universidade Federal de Minas Gerais e integrado ao Praxis [26].

O método está apresentado nas subseções a seguir através de um conjunto de visões que abordam aspectos como projeto, iteração e desenvolvimento. Essa subdivisão em visões foi proposta para o XP em [34] e mantida pelo nosso trabalho, adaptando apenas as visões que sofreram algum tipo de modificação ou adaptação devido ao método XPu.

### 2.1 Visão de Projeto

A visão de projeto apresenta as atividades do método em mais alto nível de abstração. Esse nível de abstração corresponde à definição dos *releases* que irão compor o produto a ser desenvolvido.

**Atividades Técnicas.** As atividades técnicas do método XPu se iniciam com a Análise de Usuários que é feita através de Personas [8]. Essa técnica foi desenvolvida por Alan Cooper com o objetivo de caracterizar grupos de usuários finais através da observação direta destes em seus locais habituais de trabalho. Com a utilização de Personas é possível descrever protótipos típicos desses usuários finais.

Paralelamente, a Análise de Tarefas é feita através do uso de Roteiros [27]. Roteiros fazem parte de uma técnica de IHC que propõe uma narrativa textual concreta e rica em detalhes contextuais para descrever a execução de uma tarefa por um perfil de usuário típico. Os Roteiros, portanto, representam situações de uso e são utilizados para conhecimento dos usuários e suas necessidades.

Ao final da Análise de Usuários e da Análise de Tarefas, estará completa a atividade de Análise de Contexto de Uso. A Análise de Contexto de Uso é representada pelas Histórias de Usabilidade. As Histórias de Usabilidade têm um papel similar às Histórias de Usuários já utilizadas pelo método XP. No entanto, há uma diferença importante entre elas. Ao contrário das Histórias de Usuário que são escritas pelo próprio cliente no método XP, as Histórias de Usabilidade são escritas pelo Projetista de Interação (papel sugerido pela segunda edição do método XP [4] e, de fato, incorporado e implantado pelo XPu) com a colaboração do Cliente. O Projetista de Interação é responsável por todas as atividades de usabilidade propostas pelo método.

As Histórias de Usabilidade, portanto, definem os Requisitos de Usabilidade e os Cenários de Teste que formam os insumos, respectivamente, para as atividades de Planejamento do *Release* (que é uma atividade do XP) e Avaliação de Usabilidade (que é uma atividade do XPu). A Avaliação de Usabilidade no XPu possuem escopo aberto. Os recursos produzidos por eles são os Testes de Usabilidade que podem ser feitas através de inspeções como Avaliações Heurísticas [23] ou Avaliações com Usuários [28], dependendo da abrangência da iteração que está sendo entregue aos Testes de Aceitação.

**Atividades Gerenciais.** As atividades gerenciais do método XPu se iniciam com o Planejamento de Usabilidade que gera um conjunto de Estimativas de Recursos que serão utilizados pelo método. Essa Estimativa de Recursos envolve previsões de tempo e custo envolvidos no Planejamento do *Release*. Ou seja, essa estimativa serve de insumo para o Jogo do Planejamento – atividade de planejamento do método XP – e influenciará diretamente na estimativa do prazo para entrega de *releases* e iterações.

Ao final da iteração, a velocidade do projeto é avaliada e os atrasos ou adiantamentos relacionados à usabilidade que foram observados durante a iteração são incorporados e reavaliados pela atividade de Controle de Usabilidade. Essa atividade tem como principal objetivo o acompanhamento das atividades de usabilidade, comparando o esforço previsto e o realizado para melhorar a qualidade das estimativas da atividade de planejamento para a iteração seguinte. O Controle de Usabilidade é feito pelo Projetista de Interação com a colaboração do *Tracker* do XP.

## 2.2 Visão de Iteração

A visão de iteração apresenta as atividades do método em um nível de detalhamento mais refinado. Esse nível de detalhamento permite abordar as questões específicas das iterações que compõem o *release*.

As atividades do método XPu que são realizadas durante uma iteração se iniciam com a Definição de Requisitos e Metas de Usabilidade. Essa definição permite o estabelecimento de *Benchmarks* que servirão de balizadores para avaliar a qualidade da usabilidade que está sendo entregue ao final de cada iteração. Naturalmente, essa definição influencia no Planejamento da Iteração – atividade do método XP – e, portanto, serve de insumo para ele. A Definição de Requisitos e Metas de Usabilidade é feita na primeira iteração e refinada nas iterações seguintes.

Após definido o Plano de Iteração pelo método XP, a atividade de Definição de Estilos de Interação permite definir um padrão de *Templates*. Os *Templates* são modelos estruturados de interface que permitem definir um guia de estilo padrão e coerente entre todos os artefatos de interação que serão utilizados pelo produto a ser desenvolvido. Esses *templates*, portanto, servem de insumo para o Desenho da Interação que resulta na definição final das interfaces que serão utilizadas ao longo de todo o desenvolvimento do produto.

## 2.3 Visão de Desenvolvimento

A visão de desenvolvimento propõe uma última instância de refinamento. Esse nível de detalhamento permite abordar as questões específicas ligadas ao desenvolvimento de uma tarefa pertencente a uma iteração.

As atividades do método XPu que são realizadas durante o desenvolvimento da implementação de uma tarefa se iniciam com a Prototipação de Interface. A prototipação é a primeira instância do desenvolvimento da interface e compõe o Desenho da Interação. O protótipo (inicialmente simplificado) é refinado à medida que o produto vai sendo desenvolvido, de modo que o Desenho da Interação esteja concluído ao final da iteração correspondente.

Por fim, após serem rodados com sucesso todos os testes unitários para um conjunto consistente de tarefas, uma interface funcional daquela iteração é submetida a uma Avaliação de Usabilidade. Idealmente, quando ainda estamos trabalhando com protótipos simplificados, o método de avaliação indicado é a Avaliação Heurística. À medida que tenhamos interfaces próximas da definição final, podemos utilizar avaliações mais abrangentes com usuários finais. Ao final das tarefas daquela iteração, o sucesso nos Testes de Usabilidade é uma das condições necessárias para o sucesso dos Testes de Aceitação, que no método XP são estabelecidos pelo próprio cliente.

### 3 Estudo de Caso

O estudo de caso realizado para validação do método XPu foi conduzido durante uma semana no início do ano de 2009, com uma equipe de dois desenvolvedores recém-graduados em Sistemas de Informação, sendo um deles com vasta experiência em desenvolvimento de software e outro menos experiente. O sistema proposto foi um software interativo de busca de mídias em locadoras de DVDs. O objetivo do sistema era o de auxiliar usuários em terminais de auto-atendimento na busca por vídeos e na locação automática da mídia escolhida, sem a necessidade da intervenção de um atendente. Um dos autores deste trabalho atuou no papel de Cliente durante o projeto que foi rigorosamente desenvolvido segundo o ambiente e as práticas ágeis propostos pelo método XP. Isto é, toda a configuração do ambiente de desenvolvimento ágil foi utilizada, bem como todas as práticas ágeis pertinentes ao escopo do projeto.

O Cliente propôs o desenvolvimento de 2 Histórias de Usuário para o projeto. A primeira delas descrevia que o usuário da locadora deveria ser capaz de realizar uma busca de filmes através de um terminal interativo na locadora. A segunda descrevia que o usuário deveria ser capaz de locar a mídia escolhida através do terminal e receber um comprovante de sua locação contendo os filmes locados, as datas de entrega e o valor a pagar por cada um deles. Para validar o estudo, o projeto foi desenvolvido em duas etapas. Na primeira delas, foi utilizado o método XP tradicional, sem adaptações. Na segunda delas, foi utilizado o método XPu, proposto por nosso trabalho. O objetivo de validação foi comparar a qualidade da interface (medida através da quantidade de erros encontrados) com o custo de tempo de desenvolvimento (medido através da sobrecarga de trabalho imposta ao time pela adição das atividades do método XPu).

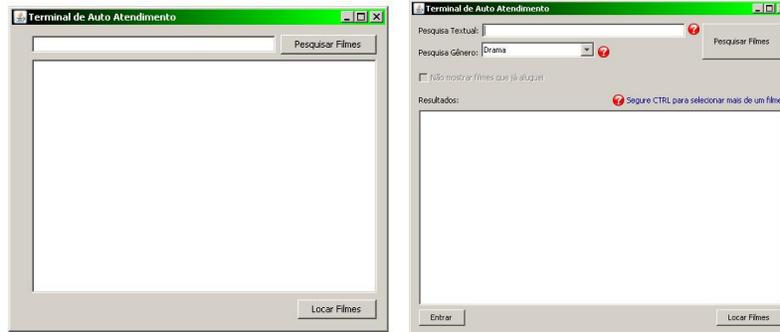
No primeiro desenvolvimento (utilizando o método XP tradicional), os dois participantes do estudo receberam treinamento apenas sobre métodos ágeis (com foco no método XP) e atuaram durante todo o projeto formando um Par de Programação (*Pair Programming*). Como foram propostas apenas 2 Histórias de Usuário, estas naturalmente foram organizadas em um mesmo *release* e em uma mesma iteração. Logo, a atividade inicial de planejamento utilizada foi o Planejamento da Iteração, visto que não faria sentido um Planejamento de *Release*, dado o escopo do projeto. Portanto, durante o Planejamento da Iteração e após receberem as histórias a serem desenvolvidas, o par atuou no mapeamento e estimativa das tarefas daquela iteração utilizando a técnica de *Planning Poker*. Inicialmente, a dupla estabeleceu uma escala

relativa de 1 a 5 para classificar a complexidade das histórias. A primeira história foi classificada com complexidade 5 e a segunda com complexidade 1. Após essa classificação, o tempo estimado para o desenvolvimento de cada uma das tarefas foi registrado através do consenso nas estimativas fornecidas por cada um dos dois integrantes da dupla.

Em seguida, a dupla optou por utilizar o ambiente NetBeans como plataforma de desenvolvimento em linguagem Java, utilizando o *framework* JUnit como suporte para criação dos testes automatizados. O SGBD de escolha foi o MySQL. Após essas definições, os testes unitários foram escritos para cada uma das tarefas da iteração e logo em seguida o desenvolvimento do código-fonte foi iniciado. Dado o pequeno escopo do projeto, a dupla modelou o sistema através de rascunhos em papel e utilizou a comunicação direta entre eles, dado que não se fazia necessário um Ambiente Informativo para disseminar a informação através de um maior número de colaboradores. Ao longo do desenvolvimento, métricas relativas ao tempo realmente gasto em cada tarefa foram sendo registradas por um dos membros da dupla (no papel de *Tracker*) e ao final do desenvolvimento, o Cliente validou a entrega através dos Testes de Aceitação.

Após a entrega, a dupla iniciou a segunda etapa do estudo que era o desenvolvimento do mesmo produto utilizando o método XPu. Naturalmente, era de se esperar o reuso de código, visto que a aplicação já havia sido desenvolvida anteriormente. Por essa razão, tivemos uma idéia clara da sobrecarga de trabalho imposta pelo método XPu, uma vez que todo o tempo de desenvolvimento gasto nessa segunda etapa seria creditado à adição das atividades de usabilidade do método e, posteriormente, à reescrita de código imposta pelas modificações relacionadas à usabilidade do produto.

Na segunda etapa do estudo, a dupla recebeu treinamento em processos e métodos de usabilidade (com foco no método XPu) e em técnicas de IHC. Após o treinamento, a equipe iniciou o desenvolvimento pela re-estimativa de tempo das tarefas da iteração (Planejamento de Usabilidade). Em seguida, foram construídas as Personas e os Roteiros (Análise de Usuários e Tarefas) para compor as Histórias de Usabilidade. A partir do momento em que os dois membros da equipe observaram as Histórias de Usabilidade, eles perceberam rapidamente a necessidade de redesenhar principalmente a interface de busca de mídias para contemplar os diversos aspectos de usuários e tarefas pertinentes àquele projeto e que não tinham sido observados durante a primeira etapa do estudo (utilizando o método XP tradicional). Baseado nesses aspectos, a dupla rascunhou em papel os *Templates* e Protótipos Simplificados para as novas interfaces e em seguida, iniciaram as atividades de redesenho (Desenho da Interação).



**Fig. 1.** Evolução da interface de busca de mídias. À esquerda, interface gerada pelo método XP. À direita, interface gerada pelo método XPu.

Como é possível perceber na Figura 1, o ganho de qualidade percebido na interface gerada pela equipe com a utilização do método XPu é considerável. Dado o reduzido escopo do projeto, as Avaliações de Usabilidade realizadas em todas as interfaces ao final da iteração (bem como todas as demais atividades de usabilidade) foram feitas pelos próprios desenvolvedores (no papel de Projetista de Interação) utilizando a Avaliação Heurística. A Avaliação Heurística foi realizada por cada um dos membros da equipe separadamente. Por fim, o produto foi entregue ao Cliente para os Testes de Aceitação.

### 3.1 Discussão dos Resultados

Baseado nos objetivos do estudo de caso, os resultados a serem discutidos dizem respeito à qualidade da interface gerada pelo método XPu em relação à sobrecarga de trabalho imposta por ele. Para tanto, a interface de busca de mídias (que sofreu o maior número de modificações devido à adoção do XPu) foi inspecionada através da Avaliação Heurística antes e depois da utilização do método proposto para verificar o número de erros encontrados em cada versão. Já os dados relativos ao tempo consumido pelas atividades do método XP e do método XPu foram separadamente isolados e registrados. Para tabular esses dados, a Tabela 1 apresenta o número de erros encontrados nas interfaces geradas pelos dois métodos, bem como os seus respectivos tempos totais consumidos.

**Tabela 1.** Comparativo dos resultados para os métodos XP e XPu.

Resultados	XP	XPu
Erros encontrados	8	4
Tempo consumido	10:30	2:10

A Tabela 1 mostra que as atividades de usabilidade do método XPu sobrecarregaram o trabalho da equipe em 2 horas e 10 minutos e reduziram pela

metade o número de erros encontrados na interface gerada. Ou seja, com cerca de 20% a mais de tempo investido nas atividades de usabilidade, o método XPu dobrou a qualidade da interface produzida e entregue ao usuário final para esse estudo de caso.

No entanto, seria prematuro afirmar que esse quadro de melhoria provido pela adoção do método XPu se repetiria em outros contextos. Sabemos através da literatura que o investimento em atividades de usabilidade comprovadamente melhora a qualidade das interfaces geradas. O desafio aqui, portanto, é analisar a que custo essa melhoria é alcançada através do método XPu. Os dados preliminares apresentados aqui sugerem que esse custo é relativamente baixo, visto que a adoção das atividades de usabilidade consome um tempo relativamente pequeno da equipe de desenvolvimento e ainda mantém o processo com as características que se espera para abordagens ágeis [29]. Além disso, em entrevista com os desenvolvedores após o estudo de caso, eles relataram se sentirem mais seguros e confortáveis com a qualidade do produto que estavam desenvolvendo, após realizarem uma análise mais apurada dos usuários finais e das tarefas que estes realizariam sobre o sistema.

#### **4 Considerações Finais e Trabalhos Futuros**

Esse trabalho propõe uma importante adaptação ao XP do ponto de vista da qualidade de software. O grande desafio até então era encontrar mecanismos para adaptar complexos processos de usabilidade ao contexto ágil de desenvolvimento ou até mesmo propor novos métodos especialmente definidos de maneira ágil. No primeiro caso, a principal questão era avaliar até que ponto atividades de usabilidade poderiam ser adicionadas ao método, sem descaracterizar sua agilidade. No segundo caso, dificilmente encontraríamos parâmetros suficientes para mensurar a qualidade do processo de usabilidade definido especialmente para aquele fim, sem que para isso utilizássemos técnicas já consagradas nos processos tradicionais, incorrendo novamente à necessidade de adaptação.

Nosso trabalho optou por definir o método XPu a partir da primeira opção, haja visto que a comprovada eficiência das técnicas aqui utilizadas contribuem de maneira inequívoca para o sucesso do desenvolvimento centrado no usuário. A questão de avaliar até que ponto o método definido continua mantendo suas características ágeis foi respondida com a inspeção do método segundo critérios técnicos e objetivos definidos na literatura da área [29]. Já a questão da viabilidade de adoção do método proposto no contexto de um projeto real de desenvolvimento de software foi aqui apresentada, mesmo que com resultados limitados. Naturalmente, outros estudos são necessários para avaliar a abrangência dessas atividades e até que ponto elas são factíveis em projetos de maior porte.

Outros trabalhos relacionados também descreveram importantes contribuições para essa integração, porém alguns deles sem a devida discussão de suas características ágeis, outros pautados em processos de usabilidade não tão bem definidos como o Praxis-u e outros bastante generalistas, não apontando soluções para problemas inerentes ao ciclo diário de desenvolvimento de uma equipe ágil.

Diversos trabalhos futuros podem advir do método XPu e dos resultados preliminares de avaliação aqui apresentados. Os primeiros deles certamente estão relacionados a novas avaliações e validações pelas quais o método poderia passar em

diferentes contextos. Outro trabalho que vem sendo desenvolvido no momento é a modelagem do XPu em um sistema BPMS (*Business Process Management System*). Como em qualquer processo modelado em um sistema BPMS, o XPu poderá ser executado através de uma ferramenta que permite definir papéis, fluxos e responsabilidades ao longo de todo o ciclo de vida de desenvolvimento do produto. Iniciativas como essa podem, inclusive, contribuir de maneira positiva para a aquisição de agilidade por parte do processo.

## Referências Bibliográficas

1. Agile Manifesto. (2001) “Agile Manifesto”. Disponível em: <http://www.agilemanifesto.org>.
2. Armitage, J. (2004) “Are Agile Methods Good for Design?”, In: Interactions, Volume: 11, Issue: 1, Making Scents: Aromatic Output for HCI, Column: Design, Pages: 14 – 23, ACM Press, NY, USA.
3. Beck, K. (1999) “Extreme Programming Explained: Embrace Change”, Addison-Wesley Professional.
4. Beck, K.; Andres, C. (2004) “Extreme Programming Explained: Embrace Change”, Addison-Wesley Professional, 2<sup>nd</sup> edition.
5. Broschinsky, D.; Baker, L. (2008) “Using Persona with XP at LANDesk Software, an Avocent Company”, In: Agile 2008 Conference, IEEE Computer Society.
6. Constantine, L. L. (2001) “Process Agility and Software Usability: Toward Lightweight Usage-Centered Design”, In: Information Age, Volume: 8, Número: 2.
7. Constantine, L. L.; Lockwood, L. A. D. (2003) “Usage-Centered Software Engineering: An Agile Approach to Integrating Users, User Interfaces, and Usability into Software Engineering Practice”, In: Proceedings of the 25<sup>th</sup> International Conference on Software Engineering (ICSE’03), IEEE Computer Society.
8. Cooper, A.; Reimann, R. M. (2003) “About Face 2.0: The Essentials of Interaction Design”, Wiley Publishing, Inc., Indianapolis, Indiana, USA.
9. Detweiler, M. (2007) “Managing UCD within Agile Projects”, In: Interactions, Volume: 14, Issue: 3, Business Leadership and the UX Manager, Special Issue: UX Management, Pages: 40 – 42, ACM Press, NY, USA.
10. Ferreira, J.; Noble, J.; Biddle, R. (2007) “Agile Development Iterations and UI Design”, In: Proceedings of Agile Conference, Pages: 50 – 58, IEEE Computer Society.
11. Fox, D.; Sillito, J.; Maurer, F. (2008) “Agile Methods and User-Centered Design: How These Two Methodologies Are Being Successfully Integrated In Industry”, In: Proceedings of Agile 2008 Conference, IEEE Computer Society.
12. Hodgetts, P. (2005) “Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes”, In: Proceedings of the Agile Development Conference, Pages: 235 – 242, IEEE Computer Society.
13. Holzinger, A.; Errath, M.; Searle, G.; Thurnher, B.; Slany, W. (2005) “From Extreme Programming and Usability Engineering to Extreme Usability in Software Engineering Education (XP + UE = XU)”, In: Computer Software and Applications Conference - COMPSAC, 29th Annual International, Volume: 2, Pages: 169 – 172, IEEE Computer Society.
14. ISO/IEC 9126-1 (2001) “Software Engineering - Product Quality - Part 1: Quality Model”, International Standards Organization.
15. ISO/IEC 9241 (1997) “Ergonomic Requirements for office work with visual display terminals (VDTs)”, International Standards Organization.

16. Kane, D. (2003) "Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet", In: Proceedings of the Agile Development Conference, Pages: 40 – 46, IEEE Computer Society.
17. Lee, J. C. (2006) "Embracing Agile Development of Usable Software Systems", In: Conference on Human Factors in Computing Systems – CHI'06 Extended Abstracts on Human Factors in Computing Systems, Pages: 1767 - 1770, ACM Press, NY, USA.
18. Lee, J. C.; McCrickard, D. S. (2007) "Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development", In: Proceedings of Agile Conference, Pages: 59 – 71, IEEE Computer Society.
19. Lievesley, M. A.; Yee, J. S. R. (2006) "The Role of the Interaction Designer in an Agile Software Development Process", In: Conference on Human Factors in Computing Systems – CHI'06 Extended Abstracts on Human Factors in Computing Systems, Pages: 1025 – 1030, ACM Press, NY, USA.
20. McInerney, P.; Maurer, F. (2005) "UCD in Agile Projects: Dream Team or Odd Couple?", In: Interactions, Volume: 12, Issue: 6, Pages: 19 – 23, ACM Press, NY, USA.
21. Meszaros, G.; Aston, J. (2006) "Adding Usability Testing to an Agile Project", In: Proceedings of the Agile Development Conference, Pages: 289 – 294, IEEE Computer Society.
22. Najafi, M.; Toyoshiba, L. (2008) "Two Case Studies of User Experience Design and Agile Development", In: Agile 2008 Conference, IEEE Computer Society.
23. Nielsen, J. (1994) "Heuristic Evaluation", In: Mack, R. & Nielsen, J. (eds.) *Usability Inspection Methods*. New York, NY: John Wiley & Sons, 25-62.
24. Pádua, C.; Pimentel, B.; Paula Filho, W. P.; Machado, F. T. (2006) "Transitioning model-driven development from academia to real life", In: Proceedings of Educators' Symposium of the ACM / IEEE, 9th International Conference on Model Driven Engineering Languages and Systems, Pages: 61 – 77, Gênova, Itália.
25. Patton, J. (2002) "Hitting the Target: Adding Interaction Design to Agile Software Development", In: Conference on Object Oriented Programming Systems Languages and Applications, OOPSLA Practitioners Reports, ACM Press, NY, USA.
26. Paula Filho, W. P. (2003) "Engenharia de Software: Fundamentos, Métodos e Padrões", LTC, 2ª Edição.
27. Rosson, M. B.; Carroll, J. M. (2002) "Usability Engineering: scenario-based development of human-computer interaction", San Francisco, CA: Morgan Kaufmann.
28. Rubin, J. (1994) "Handbook of Usability Testing: how to plan, design, and conduct effective tests", John Wiley and Sons.
29. Silva, T. R.; Pádua, C. I. P. S.; Resende, R. S. F. XPu: caracterização de agilidade para uma abordagem ágil visando à usabilidade. In: III Workshop de Desenvolvimento Rápido de Aplicações, 2009, Ouro Preto/MG. Anais do VIII Simpósio Brasileiro de Qualidade de Software. Porto Alegre/RS : Sociedade Brasileira de Computação, 2009.
30. Singh, M. (2008) "U-SCRUM: An Agile Methodology for Promoting Usability", In: Agile 2008 Conference, IEEE Computer Society.
31. Sousa, K.; Furtado, E.; Mendonça, H. (2005) "UPi – A Software Development Process Aiming at Usability, Productivity and Integration", In: Proceedings of the Latin American Conference on Human-Computer Interaction, ACM International Conference Proceeding Series, Volume: 124, Pages: 76 – 87, ACM Press, NY, USA.
32. Ungar, J. (2008) "The Design Studio: Interface Design for Agile Teams", In: Agile 2008 Conference, IEEE Computer Society.
33. Wolkerstorfer, P.; Tscheligi, M.; Sefelin, R.; Milchrahm, H.; Hussain, Z.; Lechner, M.; Shahzad, S. (2008) "Probing an Agile Usability Process", In: Proceedings of CHI 2008 - Case Studies, Florence, Italy.
34. XP. (2006) "Extreme Programming: a gentle introduction". Disponível em: <http://www.extremeprogramming.org>.