

Design e Agile: Análise da Metodologia XPlus

Design and Agile: Xplus Methodology Analysis

Filipe V. V. T. Barbosa, Fábio Campos, Dino L. Figueirôa, Walter F. M. Correia

design centrado no usuário, usabilidade, metodologias ágeis.

O design e a usabilidade contribuíram enormemente para o desenvolvimento de software nos últimos 30 anos, agregando valor e melhorando a qualidade dos artefatos produzidos, na medida em que levam em conta as características e necessidades do público-alvo. Mesmo com o surgimento das metodologias ágeis de desenvolvimento de software, o design e a usabilidade continuaram como requisitos secundários e mesmo opcionais nos projetos. Este artigo avalia a metodologia ágil conhecida como XPlus, que introduz técnicas de design e usabilidade em seu processo, analisando de que forma essas técnicas foram inseridas e as adequações propostas.

user centered design, usability, agile methodologies.

The design and usability contributed greatly to the development of software in the last 30 years, adding value and improving the quality of the artifacts produced, as it takes into account the characteristics and needs of the audience. Even with the rise of agile software development, these subjects continued as secondary and disposable requirements on projects. This article seeks to analyze the feasibility and appropriateness of design techniques and usability embedded in an agile software development known as XPlus.

1. Introdução

Design é uma atividade projetual de caráter técnico, criativo e estético que objetiva a concepção de artefatos de consumo, através de um processo que envolve a identificação de um problema ou necessidade e a especificação de requisitos que os atendam, pesquisa e levantamento de informações relevantes ao projeto, geração, seleção e avaliação de alternativas, finalização e acompanhamento.

A engenharia de software como atividade projetual possui fases semelhantes em seu processo de trabalho, mas se diferencia do design por se concentrar na implementação do artefato em si, enquanto o design visa à relação que se estabelece entre o artefato e os atores externos. Caberá ao designer garantir que o diálogo entre o usuário e o sistema ocorra com o mínimo de ruído, identificando o que os usuários esperam do artefato, estudando como realizam suas tarefas, definindo como os usuários irão interagir com este artefato e validando se o que foi projetado realmente atende às necessidades destes, enfim, facilitando a comunicação entre o usuário e o sistema mediada pela interface. Para entender melhor como o design e a usabilidade integram-se ao desenvolvimento de software, faz-se necessário uma rápida contextualização histórica.

2. Modelos e metodologias de engenharia de software

A engenharia de software, cujo surgimento data do início da década de 1970, é a disciplina que visa o desenvolvimento de produtos de software com boa relação custo-benefício, ou seja, entregues dentro do prazo definido, do orçamento estabelecido e que atendam às necessidades dos usuários. A primeira metodologia de desenvolvimento de software ficou conhecida como “Modelo em Cascata”, que adota um viés taylorista de produção em que as atividades são realizadas em um processo linear e sequencial, com longos ciclos de desenvolvimento onde cada ciclo deve ser finalizado para que um novo inicie. Tudo deve ser muito bem analisado, projetado e testado antes de ser desenvolvido, pois mudanças em fases mais adiantadas do projeto são muito onerosas ao projeto. O modelo em cascata define 5 fases em seu ciclo de vida: 1) análise e definição de requisitos, 2) projeto de sistema e software, 3) Implementação e teste de unidade, 4) Integração e teste de sistema e 5) Operação e Manutenção.

Figura 1: Modelo em cascata.



O ciclo de vida da engenharia de usabilidade, proposto por Debora Mayhew (Mayhew, 1999), é o processo de design e usabilidade que melhor se integra ao ciclo de vida do modelo em cascata, pois define conjuntos de atividades de design e usabilidade compatíveis com as fases desse modelo.

Segundo TELES (2005), os principais problemas do modelo em cascata residem em desconsiderar que projetos de software:

- não são lineares, pois são fruto de trabalho intelectual e criativo, não de trabalho repetitivo e determinístico, característico de outras engenharias;
- necessitam de envolvimento ativo e constante dos usuários para validar requisitos, interfaces e interações propostas pela equipe;
- precisam se adequar às mudanças e que elas ocorrem durante todo o projeto, sendo essenciais para produzir um software útil, relevante e adequado às necessidades dos clientes e usuários.

Devido à esta natureza linear e sequencial do modelo em cascata, qualquer mudança em fases mais adiantadas do projeto, geram impactos e retrabalho em várias as fases, comprometendo o cronograma, os custos do projeto e a qualidade final do produto.

Metodologias Ágeis

A partir do início da década de 1990, surgiram novas metodologias como Scrum, Dynamic Systems Development Method (DSDM), Crystal Methods, Feature-Driven Development (FDD), Lean Development (LD), Extreme Programming (XP) e Adaptative Software Development (HIGHSMITH, 2002), todas baseadas no “Modelo Incremental” de desenvolvimento, cuja abordagem difere do “Modelo em Cascata” por definir que as atividades do processo de software sejam realizadas em paralelo e em pequenos ciclos de desenvolvimento (iterações).

Figura 2: Modelo incremental.



Segundo Somerville (SOMMERVILLE, 2003) o objetivo comum destas metodologias eram:

- reduzir os erros de projeto e o conseqüente retrabalho;
- permitir que os clientes e a equipe adiem decisões importantes de projeto para o momento em que detiverem o conhecimento necessário;
- possibilitar ao cliente e à equipe testarem o sistema aos poucos, desde os estágios iniciais de desenvolvimento;
- encarar as mudanças durante o projeto como algo positivo e necessário para entregar um software de valor para o cliente e os usuários;
- diminuir o risco de fracasso do sistema ou de parte dele e gerenciar melhor os riscos de maneira geral.

Em 2001, durante um workshop realizado em Snowbird, Utah, EUA, um grupo de consultores veteranos da indústria de software sistematizou um conjunto de valores e princípios, fruto de suas

experiências em projetos de software bem sucedidos e produziu um documento que ficou mundialmente conhecido como Manifesto para o Desenvolvimento Ágil de Software ou simplesmente Manifesto Ágil¹, agrupando as metodologias supracitadas sob o rótulo de “Metodologias Ágeis”.

Em um projeto que utiliza metodologia ágil os ciclos de desenvolvimento são curtos, entre 2 e 4 semanas, exigindo mudança significativas nas atividades de design e usabilidade. Segundo Cybis (CYBIS et al, 2010: 145): “A modelagem e o projeto das interfaces devem ser orientados a padrões de projeto. As avaliações ergonômicas e testes de usabilidade devem ser realizados rapidamente, assim como a especificação das revisões necessárias na interface.”

Para tentar mudar este cenário, em 2008 foi proposta uma metodologia revisional intitulada XPlus (GUIMARÃES, 2009), que insere técnicas de design e de usabilidade em determinadas fases do ciclo de desenvolvimento de software. Dado que esta revisão metodológica surgiu sob a ótica da engenharia de software, o presente estudo procura inverter este enfoque, analisando esta proposta sob o ponto de vista do design e da usabilidade.

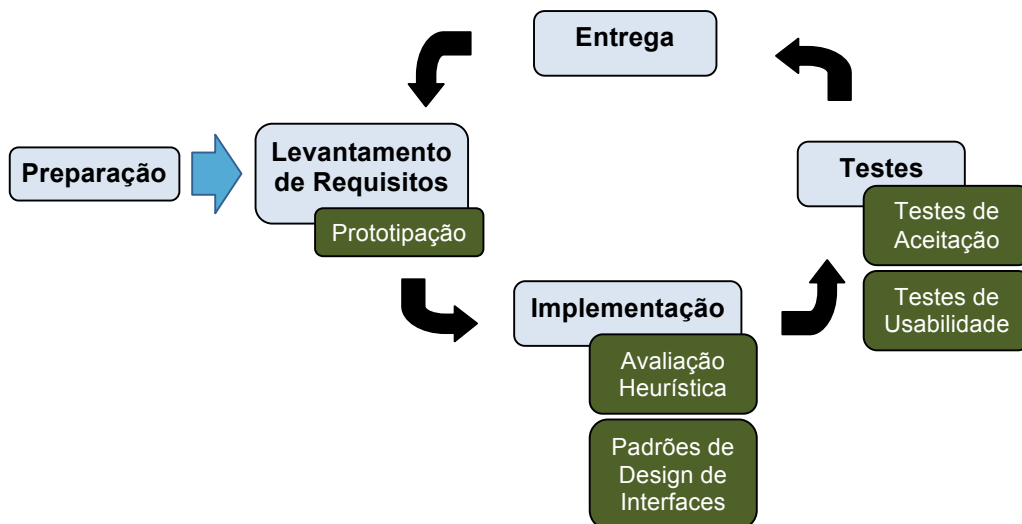
3. Objetivos e Pesquisa

O objetivo deste artigo é fazer uma pesquisa analítica da metodologia XPlus, no que concerne à contribuição do design e da usabilidade em seus processos, identificando quais técnicas de design e usabilidade foram utilizadas, em que fases do processo foram inseridas e qual seu objetivo, que adaptações foram feitas e porquê, quais os resultados esperados e quais foram atingidos em sua aplicação. A metodologia XPlus foi escolhida pelo fato de revisar a metodologia XP (eXtreme Programming), que é bem conhecida entre os times de desenvolvimento ágil.

4. Metodologia XPlus

A metodologia XPlus se propõe a inserir técnicas de design e usabilidade nas principais fases do ciclo iterativo da metodologia XP (eXtreme Programming). A figura abaixo destaca estas técnicas em cada passo do ciclo iterativo:

Figura 3: Ciclo de iterativo da metodologia XPlus.



¹ Disponível em português em: <http://agilemanifesto.org/iso/ptbr/>

A preparação é a fase inicial antes do início das iterações e seu objetivo é estabelecer o primeiro contato entre a equipe e o cliente, que vai expor seu objetivo e suas necessidades em relação ao futuro sistema para a equipe tenha uma visão geral do projeto.

Na fase de levantamento de requisitos os representantes do cliente e a equipe de desenvolvimento se reúnem para definir quais funcionalidades e interfaces de usuário serão implementadas primeiro. Nesta reunião são produzidas as histórias, que descrevem de forma simples o que cada funcionalidade deve fazer. Se a funcionalidade necessitar de interação com o usuário, o designer deve ser esboçar um protótipo de baixa fidelidade em papel que será ajustado em conjunto com o cliente e a equipe até que esteja aprovado. Este protótipo deve detalhar a interface do usuário, como será a interação e o feedback que o sistema deve fornecer.

Ao iniciar a fase de implementação as tarefas são distribuídas entre os desenvolvedores por ordem de prioridade e, caso a funcionalidade possua interface de usuário, o desenvolvedor deve formar uma dupla com o designer (pair programming). O desenvolvedor utiliza o protótipo para implementar a interface e o designer dá as orientações necessárias e indica os padrões de interfaces de usuário a ser utilizados quando necessário. Para estes padrões a metodologia XPlus sugere o trabalho de Jennifer Tidwell como referência (TIDWELL, 2005). Enquanto a funcionalidade está sendo implementada pelo desenvolvedor, o designer realiza uma avaliação heurística do protótipo seguindo as 10 heurísticas propostas por Nielsen (NIELSEN, 1994).

Na fase de testes a equipe apresenta o que foi implementado ao cliente para validar as histórias e tarefas desenvolvidas, tanto em termos de funcionalidades quanto de interfaces de usuário. As funcionalidades e interfaces rejeitadas serão objeto de correção pela equipe na iteração seguinte, repetindo todo o ciclo de desenvolvimento até serem revalidadas. Todas as funcionalidades e interfaces validadas nesta reunião devem ser objeto de testes de usabilidade com usuários reais que utilizarão o sistema, utilizando uma versão reduzida da proposta de testes de Nielsen (NIELSEN, 1993).

Análise da Metodologia Xplus

Durante a preparação do projeto, a metodologia XPlus não estabelece a atividade de levantamento de informações para o projeto de design (pesquisa de mercado, público-alvo, produtos concorrentes ou similares, briefing, análise SWOT etc), ou para a pesquisa de usabilidade (perfil dos usuários, contexto de uso, personas, cenários, pesquisa etnográfica etc). Estas informações serviriam de suporte para as etapas de prototipagem e geração de alternativas de design, bem como para as avaliações heurísticas e testes de usabilidade, previstas nas fases de levantamento de requisitos, desenvolvimento e testes.

Na fase de levantamento de requisitos a metodologia propõe a construção coletiva dos protótipos, em um processo coletivo que envolve representantes do cliente, designers e desenvolvedores, que deverá ser permanentemente melhorado durante todo o projeto. Esta prática é importante, pois permite a todos que, ao visualizarem o protótipo, possam amadurecer o entendimento sobre as funcionalidades a serem desenvolvidas, sugerir mudanças e ter novas idéias, identificar funcionalidades inter-relacionadas, antecipar possíveis necessidades, problemas, riscos e desafios de projeto e desenvolvimento. Também permite que a equipe avalie as sugestões do cliente de modo a evitar excessos como:

- interfaces e implementações complexas demais;
- erros de interação da interface;
- funcionalidades ou partes delas que não serão utilizadas na prática.

A armadilha mais comum na fase de levantamento de requisitos é que os clientes estejam representados apenas por gerentes e supervisores, excluindo usuários. Colaboradores de nível gerencial são importantes porque detém o conhecimento dos processos de negócio e possuem uma visão mais sistêmica da organização, essenciais para que a equipe entenda corretamente os conceitos e objetivos gerais que o sistema deve contemplar, bem como as regras de negócio que devem ser implementadas. No entanto, estes representantes normalmente não serão os usuários finais destas funcionalidades, especialmente das atividades do nível operacional da organização, ignorando detalhes importantes de sua rotina e realização diária. É importante que os usuários

sejam incorporados ao processo de levantamento de requisitos de modo que o sistema reflita adequadamente suas necessidades, vocabulários e fluxo de execução de tarefas.

Na fase de desenvolvimento, a implementação das interfaces de usuário utilizando protótipos e em padrões de interface de usuário é uma prática que gera bons resultados, mas são insuficientes se não puderem ser validados com os usuários através de técnicas e pesquisas de design e usabilidade. Assim, corre-se o grave risco de implementar funcionalidades que atendem aos objetivos e regras de negócio, mas que são inadequadas ao uso.

As histórias e protótipos especificados precisam atender às suas necessidades de uso e expectativas dos usuários, ou seja, devem agregar valor à atividade, facilitando e simplificando a realização do trabalho, de modo eficaz, eficiente e satisfatório. Essa participação precoce dos usuários é vital para evitar a reconstrução total ou parcial de uma funcionalidade quando o sistema já estiver funcionando, que é um procedimento mais demorado e dispendioso de realizar do que se fossem feitas nas fases iniciais do projeto.

Para isso diversas técnicas pesquisa e geração de alternativas poderiam ser aplicadas na fase de especificação do softwares como: personas, cenários, contexto de uso, pesquisa etnográfica, card sorting, mapas mentais, testes de usabilidade com protótipos de papel, protótipos navegáveis etc (TEIXEIRA, 2011). Obviamente que nem todas as técnicas cabem no cronograma e/ou orçamento de um projeto, são adequadas e possíveis de serem aplicadas em todos os projetos, ou mesmo necessárias num contexto de desenvolvimento ágil, mas geram um feedback importante para a equipe. Também é importante atentar ao fato de que estas técnicas devem gerar uma documentação mínima, resumida e acessível dos resultados, para possam ser rapidamente analisados pela equipe, confirmando ou refutando a especificação feita com os colaboradores de nível gerencial.

A metodologia também propõe que:

“A validação em relação às funções devem ocorrer semelhante ao processo de XP. O desenvolvimento deve ser Orientado a Testes. Portanto, o designer de interação também deverá se atentar ao código, realizando o trabalho tradicional do par no desenvolvimento que é propor testes e refatorações, auxiliando o desenvolvedor a implementar soluções simples e bem descritas.”. (GUIMARÃES, 2009, p. 53)

A tarefa destacada na citação acima se revela inviável de ser realizada na prática, pois esbarra na limitação técnica de designers e engenheiros de usabilidade cuja área de estudos não inclui atividades de programação e arquitetura de software.

5. Conclusões

Apesar da metodologia XPlus representar um avanço em relação a outras metodologias, reconhecendo a importância do design e da usabilidade no desenvolvimento de software, a mesma representa apenas o início de um processo a ser seguido em revisões metodológicas posteriores, pois apenas sugere, de maneira muito sucinta, a aplicação das técnicas, de forma muito simplificada e isolada, destinando tempo insuficiente para a realização de técnicas e análise dos dados obtidos.

Segundo NIELSEN (2009), a própria natureza das metodologias ágeis de planejar minimamente o que será desenvolvido na próxima iteração, não deixa tempo suficiente para pensar no projeto todo, obrigando o profissional a analisar e projetar considerando apenas, aumentando o retrabalho com o projeto e aumentando as chances do produto final se parecer com uma colcha de retalhos, à medida que não se tem a visão completa do produto na hora do seu desenvolvimento.

Já MELCHER e MON'T ALVÃO (2011), em seu artigo intitulado Proposta metodológica para aplicação de avaliações de usabilidade otimizadas em produtos desenvolvidos com métodos ágeis: Estudo de caso da Globo.com, abrem o caminho propondo uma abordagem mais adequada para a aplicação de técnicas de usabilidade sucintas sem comprometer a análise do dados e a qualidade do produto final.

A conclusão que se chega é que as revisões metodológicas propostas até o momento surgiram como iniciativas positivas da engenharia de software, no sentido de reconhecer da importância e o impacto das disciplinas de design e de usabilidade no desenvolvimento de artefatos de software. No entanto, as técnicas foram incorporadas às metodologias por profissionais da área de ciência da computação e afins, sem que houvesse a devida avaliação e adaptação necessárias para

atingir os objetivos que as técnicas visam, desconsiderando o processo de trabalho existente e a viabilidade de aplicação de tais técnicas.

Tanto MELCHER e MON'T ALVÃO (2011) como CYBIS (2010) indicam que o caminho mais acertado para a inserção de técnicas design e usabilidade deve originar-se das áreas de design e engenharia de usabilidade e não o inverso, já que possuem o corpo de conhecimento adequado para selecionar, propor e adequar tais técnicas às metodologias ágeis, atingindo resultados satisfatórios sem comprometer os resultados das técnicas e, conseqüentemente, a qualidade final do artefato.

Referências

- CYBIS, W., BETIOL, A. H., FAUST, R. 2010. *Ergonomia e usabilidade: conhecimentos, métodos e aplicações*. 2 ed. São Paulo: Novatec Editora.
- GUIMARÃES, C. P. 2009. *Integrando o Design de Interfaces centrado na Experiência do Usuário ao processo de Desenvolvimento de Software com Extreme Programming*. Monografia (Graduação) – Departamento de Ciência da Computação, Instituto de Matemática, Universidade Federal da Bahia. Salvador.
- HIGHSMITH, JAMES A. 2002. *Agile software development ecosystems*. 2 ed. São Paulo: Novatec Editora.
- MAYHEW, D.J. 1999. *The usability engineering lifecycle: a practitioner's handbook for user interface design*. San Francisco: morgan Kaufman.
- MELCHER, C., MON'T ALVÃO, C. 2011. Proposta metodológica para aplicação de avaliações de usabilidade otimizadas em produtos desenvolvidos com métodos ágeis: Estudo de caso da Globo.com. *Anais da 3ª Conferência Latino Americana de Design de Interação*. 1 ed. Belo Horizonte.
- NIELSEN, J. 1993. *Usability Engineering*. San Francisco: Morgan Kaufman.
- _____, J. 1994. *Ten Usability Heuristics*. Disponível em: http://www.useit.com/papers/heuristic/heuristic_list.html. Acesso em: 02/04/2012.
- _____, J. 1994. *Heuristic evaluation*. San Francisco: Morgan Kaufman.
- NIELSEN, Jakob; NODDER, Chris. 2009. *Agile Usability: Best Practices for User Experience on Agile Development Projects*. 2 ed, 119p. Fremont, CA: Nielsen Norman Group Report.
- PREECE, J.; ROGERS, Y.; SHARP, H. 2005. *Design de interação: além da interação homem-computador*. Trad. Viviane Possamai. 2 ed. São Paulo: Bookman.
- SOMMERVILLE, I. 2003. *Engenharia de Software*. 1. ed. São Paulo: Addison Wesley.
- TELES, V. M. 2005. Um Estudo de Caso da Adoção das Práticas e Valores do Extreme Programming. Dissertação (Mestrado) – Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro. Disponível em: <http://improveit.com.br/xp/dissertacaoXP.pdf>. Acesso em: 28/03/2012.
- TEIXEIRA, F. 2011. Os entregáveis da Arquitetura de Informação. Disponível em: <http://arquiteturadeinformacao.com/2011/06/09/entregaveis-de-arquitetura-de-informacao/>. Acesso em: 02/04/2012.

Sobre os autores

Filipe Virgínio Vital Torres Barbosa, graduado em Sistemas para Internet pela Faculdade Marista do Recife (2012), com experiência em design de interfaces digitais, usabilidade, acessibilidade, metodologias de desenvolvimento ágil e gerência de projetos. Possui artigo publicado nos anais do 10° P&D Design – Congresso Brasileiro de Pesquisa e Desenvolvimento em Design.

<filipe.virginio.torres@gmail.com>

Fabio Ferreira da Costa Campos, doutor em Ciências da Computação pela UFPE (2004) onde é Professor Adjunto. Revisor de periódico da IET Radar, Sonar & Navigation. Experiência em Design de Produto. Atuando nos seguintes temas: Combinação de Evidências, Probabilidade Imprecisa, Representação do Conhecimento, Teoria de Dempster-Shafer, Teoria Matemática da Evidência.

<fc2005@gmail.com>

Dino Lincoln Figueirôa Santos, primeiro doutor em design do Brasil formado por uma Universidade Federal (2012). Professor em programas graduação e pós-graduação em Design, Engenharia de Software. Mestre em Design pela UFPE (2009). Possui experiência nas áreas de: Design de Artefatos Inteligentes/Artefatos Digitais, Metodologia de Design, Usabilidade, Interface Homem-Máquina.

<dinolincoln@gmail.com>

Walter Franklin Marques Correia, doutor em Engenharia de Produção pela UFPE, professor no Programa de Pós-Graduação em Design. Ampla experiência nas áreas de Design de Produto, atuando principalmente nas áreas de Segurança do Produto, Design, Usabilidade e Prototipagem Rápida. Possui bolsa de Produtividade em Pesquisa Nível 2 pelo CNPq.

<ergonomia@terra.com.br>

[Artigo recebido em dezembro de 2012, aprovado em dezembro de 2012]