

Impacto do Apoio Metodológico Ferramental a Aspectos de Usabilidade no Ensino Prático da Engenharia de Software

Giuseppe A. N. Lima, Yuska P. C. Aguiar, Bernardo Lula Jr

Departamento de Sistemas e Computação – Universidade Federal de Campina Grande
(UFCG)

Caixa Postal 10.106 – 58.109-970 – Campina Grande – PB – Brasil

{giuseppe,yuska,lula}@dsc.ufcg.edu.br

***Abstract.** In the Software Engineering teaching is desirable that students assimilate theoretical knowledge from practical application of a software design process in laboratory. To assure the learning coverage and completeness, this process should be agile, addressing both functional and usability aspects. However, historically the functional software issues are more explored and evidenced, stimulating the development and use of methodologies and support tools to your practice, in detriment of usability aspects. This article describes a solution for this problem showing the results of its application in the Laboratory of Software Engineering discipline at UFCG.*

***Resumo.** No ensino da Engenharia de Software é desejável que os alunos assimilem o conhecimento teórico adquirido com a aplicação prática de um processo de design de software em laboratório. Para garantir a abrangência e completude do aprendizado, o processo deve ser ágil e abordar os aspectos de funcionalidade e de usabilidade. No entanto, historicamente, os aspectos de funcionalidade são mais explorados e evidenciados, estimulando o desenvolvimento e uso de metodologias e ferramentas de apoio à sua prática, em detrimento dos aspectos de usabilidade. Este artigo descreve uma solução para este problema e relata o resultado de sua aplicação na disciplina Laboratório de Engenharia de Software da UFCG.*

1. Introdução

No ensino da Engenharia de Software (ES), é importante aliar a teoria à prática a fim de proporcionar ao alunado uma visão realista do uso e aplicação dos conceitos envolvidos no desenvolvimento de software. O sucesso (ensino/aprendizado) tende a ser mais facilmente alcançado quando se pode fazer uso de um processo de *design* de software (PDS) adequado ao contexto acadêmico. Entende-se por academicamente adequado um PDS que considere: (i) projetos com escopo pequeno ou médio a serem desenvolvidos dentro do período de um semestre letivo; (ii) a presença de clientes reais; (iii) o uso de processos e tecnologias considerados *estado da arte*; (iv) o aprendizado colaborativo; (v) agilidade, como conseqüência direta do tempo limitado, do escopo restrito e do aprendizado colaborativo; (vi) avaliação contínua e iterativa do progresso do projeto

para que o aprendizado dos alunos possa ser avaliado ao longo da disciplina. Além disso, com o advento, o progresso e a popularização (do uso) de novas técnicas de interação homem-computador (IHC), um PDS academicamente adequado também deve: (vii) integrar práticas de usabilidade no seu ciclo de desenvolvimento.

Segundo [Constantine 1999], a ES tem tradicionalmente construído sistemas de software tomando como foco principal suas funcionalidades e a qualidade dos sistemas mensuradas através de fatores como reusabilidade, portabilidade, confiabilidade, modularidade, acoplamento e robustez. Porém, embora um software possa estar correto funcionalmente, seus usuários podem sentir dificuldade em usá-lo, pois o processo interativo estabelecido pela interface do usuário pode não refletir as suas características e necessidades. Sendo assim, é necessário garantir que um software, além de correto funcionalmente, tenha um nível de qualidade satisfatório que colabore com a aceitação do mesmo por seus usuários. Neste caso, os fatores a serem utilizados dizem respeito à facilidade de aprender, de entender e de usar o sistema, em outras palavras, dizem respeito à usabilidade do software [Ferré 2001]. Assim, um PDS deve compreender etapas relativas às funcionalidades e à usabilidade do sistema, indicando técnicas e ferramentas para coleta, especificação e avaliação também de requisitos de usabilidade a fim de: caracterizar os usuários, compreender as tarefas efetuadas pelos mesmos, definir ambientes de trabalho e metas de usabilidade (eficiência, satisfação, aprendizado de uso, entre outras) [Antunes *et al.* 2001].

Diversos modelos e técnicas já existem para o *design* de interfaces do usuário. Entretanto, a sua integração com um PDS ainda não está muito clara nos principais modelos de processo existentes. A maioria dos modelos (Cascata, Espiral, RAD) não explicita em qual fase do processo deve acontecer a interação do *design* da interface do usuário com outras atividades do processo. Além disso, faltam técnicas e formalismos que viabilizem esta integração. Por outro lado, já é perceptível o esforço de se adaptar um PDS a fim de contemplar aspectos de usabilidade, embora ainda distante de atender as características específicas do contexto acadêmico de ensino/aprendizagem da ES ([Anderson *et al.* 2001, [Radle 2001], [Ferre *et al.* 2005], [Constantine 2008]).

O processo de *design* de software **easYProcess (YP)** [Aguiar *et al.* 2004; Vasconcelos *et al.* 2003] foi desenvolvido com o propósito de atender os requisitos supracitados de adequação ao contexto acadêmico e vem sendo utilizado nas disciplinas que envolvem a prática inicial de Engenharia de Software na Universidade Federal de Campina Grande (UFCG). Entretanto, em acompanhamento de sua aplicação, detectou-se certo desconforto por parte dos alunos em relação à elaboração dos artefatos (documentos, arquivos, testes, etc.) de usabilidade definidos pelo método. A relação custo/benefício relacionada ao tempo investido na geração destes artefatos e seu uso efetivo dentro do processo não foi considerada favorável pelos alunos. A “queixa geral” é que a produção dos artefatos de usabilidade torna a utilização de YP mais “pesada” (menos ágil) do que o desejado. Além disso, os alunos relatam dificuldades em formular um modelo consistente do processo a partir da aplicação do método de *design* de interface. Por outro lado, considerando que diversos clientes acompanharam o desenvolvimento de softwares a partir da aplicação do YP com e sem os artefatos de usabilidade, e que os relatos feitos por estes clientes expressaram um índice de satisfação superior com os produtos oriundos de YP com a produção dos artefatos de usabilidade, pode-se concluir que os artefatos de usabilidade agregam valor ao produto

final. Ou seja, a qualidade do produto obtido é satisfatória, mas a agilidade do processo e o nível de aprendizagem proporcionado pelo seu uso deixam a desejar. Assim, fez-se necessário buscar uma solução que mantivesse os ganhos com relação aos aspectos de funcionalidade, mas propiciasse o desenvolvimento teoricamente consistente dos artefatos de usabilidade e que fosse capaz de minimizar o esforço empenhado pelos alunos na construção e manutenção dos mesmos.

Este artigo descreve a solução proposta para este problema e apresenta os resultados de sua aplicação na disciplina Laboratório de Engenharia de Software (LES) da UFCG no semestre letivo 2008.2. Na seção 2 são descritos YP e o contexto metodológico proposto como solução ao problema. Na seção 3 é apresentada uma ferramenta computacional adequada de suporte a este contexto. Na seção 4 são apresentados os resultados de um estudo sobre o impacto/aceitação causado pela adoção da solução proposta em relação às práticas anteriores/atuais, a partir da reação dos alunos a uma série de fatores que determinam a satisfação, o esforço e o aprendizado. Por fim, a seção 5 apresenta os trabalhos ainda a serem desenvolvidos para a consolidação do uso do processo YP na disciplina considerada.

2. YP e o contexto metodológico para usabilidade

O easYProcess (YP) é um processo de *design* de software simplificado, apoiado em práticas do XP (*eXtreme Programm*), RUP (*Rational Unified Process*) e AM (*Agile Modeling*), para propiciar a construção de software de forma ágil. Para a elaboração de YP, as práticas, as regras e os artefatos presentes nas metodologias XP, RUP e AM foram estudados a fim de se identificar quais e como poderiam ser integrados em um PDS com aquela qualidade. YP “herda” de XP os planejamentos de *releases* e iterações (ambos com pequena duração), a especificação de *user stories* e seus respectivos testes de aceitação. Do ponto de vista de implementação, utiliza-se ainda em YP os princípios de refatoramento e integração contínuos, propriedade coletiva de código, uso de padrões de projeto, programação em pares e desenvolvimento dirigido a testes. De RUP, YP emprega o modelo de negócio (de forma simplificada) enquanto um documento de visão, o projeto arquitetural, o modelo de dados, a lista de riscos e a coleta de métricas (*big chart* em YP). AM contribuiu para YP com os princípios de mudanças incrementais e com a idéia de aprendizado colaborativo, onde todos podem aprender uns com os outros.

Com relação à usabilidade, inicialmente o YP não sugeria a construção de artefatos que apoiassem o *design* de interfaces do usuário. Os artefatos gerados com a aplicação do YP tinham como foco apenas a parte funcional do sistema. Identificada esta deficiência, a estrutura inicial do YP foi adaptada para incluir em seu fluxo a construção de artefatos de usabilidade, tais como: levantamento do perfil do usuário, objetivos de usabilidade, análise da tarefa, protótipo da interface e testes de usabilidade, de acordo com um método empírico de *design* de interface do usuário, como mostra a Figura 1.

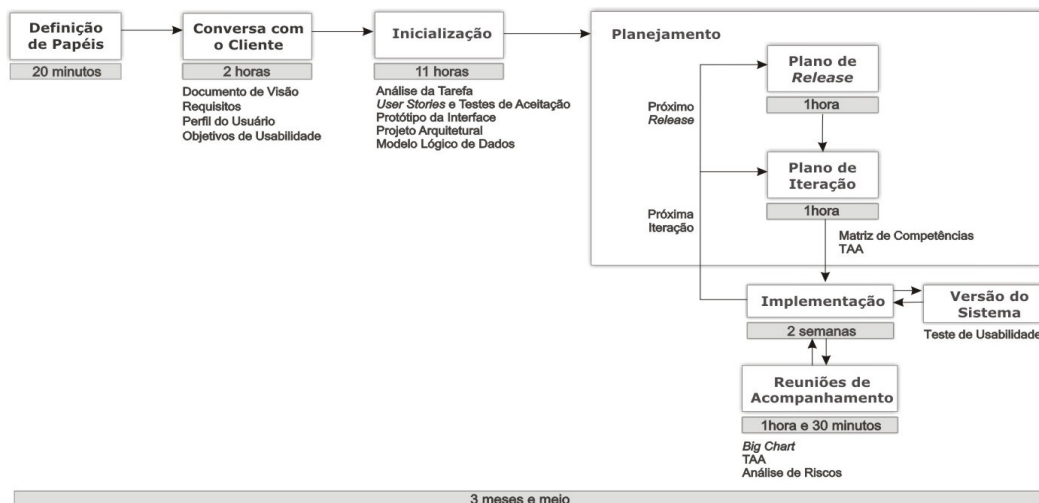


Figura 1. Fluxo do processo de desenvolvimento YP

O método de *design* de interfaces integrado a YP se baseia em uma *abordagem somativa*, em que algo deve ser construído rapidamente (prototipagem rápida), mesmo sem muita fundamentação teórica, e, através de avaliações sucessivas, deve ser refinado até um grau de aceitação satisfatório [Tavares 2002]. O ciclo de desenvolvimento é, portanto, baseado na avaliação. À época, a própria indústria de software conduzia o processo de *design* de interface do usuário dessa forma, baseada em diretrizes, princípios e ferramentas muito pouco fundamentados em teoria. No entanto, embora aparentemente satisfatória com relação aos resultados (produtos) obtidos, para o contexto acadêmico, essa abordagem traz dois problemas: alto custo de tempo e baixa aprendizagem. Os custos de produção de um protótipo inicial são relativamente baixos, mas devem ser somados aos custos de implementação de um software com baixa usabilidade, que deverá ser posteriormente corrigido através de um processo iterativo e custoso de codificação-avaliação. Basicamente, a cada iteração, tem-se um re-projeto. Com relação à aprendizagem, a abordagem é deficiente por não veicular um modelo do processo que explicita e integre os diversos conhecimentos relativos aos aspectos de usabilidade num quadro teórico compreensivo e funcional.

O acompanhamento da aplicação de YP evidenciou, portanto, como a principal causa dos problemas relacionados ao seu uso, a abordagem, e por conseguinte, o método adotado para o *design* de interfaces do usuário. Como solução, outro contexto metodológico para o *design* de interfaces, baseado em outra abordagem (*abordagem formativa*), foi, então, considerado, para se obter ambas, agilidade na construção dos artefatos de usabilidade e facilidade de compreensão da teoria subjacente ao processo. Numa abordagem formativa o objetivo é partir dos requisitos do sistema e, através de instrumentos de modelagem, chegar ao *design* da interface e, conseqüentemente, aos protótipos iniciais do sistema. Assim, os critérios levantados para a escolha do novo contexto metodológico a ser integrado a YP foram os seguintes:

- (i) existir correlação entre os artefatos definidos por YP e os artefatos definidos pela nova metodologia (compatibilidade);
- (ii) fazer uso de modelos conceituais para a compreensão dos processos envolvidos (diversos estudos da psicologia cognitiva já têm mostrado o papel fundamental do uso

de modelos para o aprendizado, para a concepção e para a operação de dispositivos interativos complexos);

(iii) existir mapeamento entre os elementos dos modelos utilizados, evidenciando a correlação entre os artefatos a serem construídos ao longo do processo e permitindo a construção de ferramentas computacionais capazes de automatizar a geração dos artefatos; e,

(iv) possibilitar a utilização de diferentes técnicas de prototipagem de modo a que usuários, clientes e projetistas possam identificar e corrigir antecipadamente problemas ocorridos no projeto.

Dentre outras possíveis, a metodologia MEDITE [Oliveira *et al.* 2007; Rodrigues *et al.* 2005; Suárez *et al.* 2004] foi selecionada, por ser baseada em modelos e múltipla prototipagem e ser compatível com o *framework* de referência *Cameleon* [Calvary *et al.* 2003], arcabouço teórico que define os níveis de abstração do processo de *design* de interface do usuário em abordagens baseadas em modelos. A Figura 2 descreve as fases, modelos e artefatos de MEDITE.

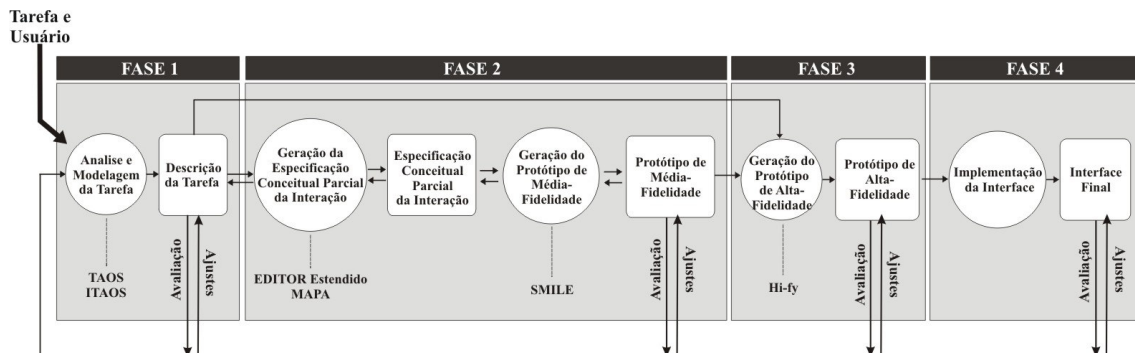


Figura 2. Fluxo, fases e artefatos de MEDITE

A metodologia MEDITE apresenta, portanto, um quadro teórico bem definido e consistente para o *design* de interfaces do usuário e que é compatível com o ciclo de desenvolvimento do YP. A agilidade do processo é garantida pela utilização da ferramenta *FastInterface* que apoia a metodologia.

3. *FastInterface*

As metodologias baseadas em modelos, como MEDITE, possibilitam a transformação evolutiva dos elementos de um modelo em elementos do modelo seguinte de seu fluxo. No entanto, embora a aplicação dos mapeamentos seja facilmente realizada manualmente em sistemas pouco interativos, a sua aplicação em sistemas mais complexos (mesmo os acadêmicos) exige um investimento de tempo e esforço não compatível com o desenvolvimento ágil desejado. Assim, utilizando a facilidade de automatização proporcionada pela característica baseada em modelo de MEDITE, foi desenvolvida *FastInterface* [Oliveira *et al.* 2007], ferramenta destinada a dar suporte computacional à metodologia, tanto na construção dos artefatos como na automatização das transformações entre modelos, provendo (i) agilidade à fase de Inicialização de YP, (ii) modelagem assistida da interface e (iii) obtenção de artefatos consistentes.

A estrutura de *FastInterface* congrega todo o processo definido por *Cameleon* e é derivado da integração de 4 módulos de apoio às fases de MEDITE: (i) iTAOS

[Medeiros *et al.* 2002], para construção e análise do modelo da tarefa (Tarefa e Conceitos/Fase 1 de MEDITE); (ii) MAPA [Rodrigues *et al.* 2005], algoritmos de transformação e manutenção da consistência (e rastreabilidade) entre o modelo da tarefa e modelo de interface abstrata do usuário (IAU/Fase 2 de MEDITE); (iii) SMILE [Aguiar *et al.* 2007], para edição e simulação do modelo de interface abstrata do usuário em forma de protótipos de média fidelidade (IAU/Fase 2 de MEDITE); e, (iv) Hi-Fy [Oliveira *et al.* 2007], para geração e edição do modelo de interface concreta do usuário sob a forma de um protótipo de alta fidelidade (ICU/Fase 3 de MEDITE) e para geração de interface final do usuário (IFU/Fase 4 de MEDITE). A Figura 3 apresenta a correspondência entre os níveis de abstração do processo de desenvolvimento proposto em *Cameleon* e os módulos que compõem *FastInterface* e as fases correspondentes de MEDITE.

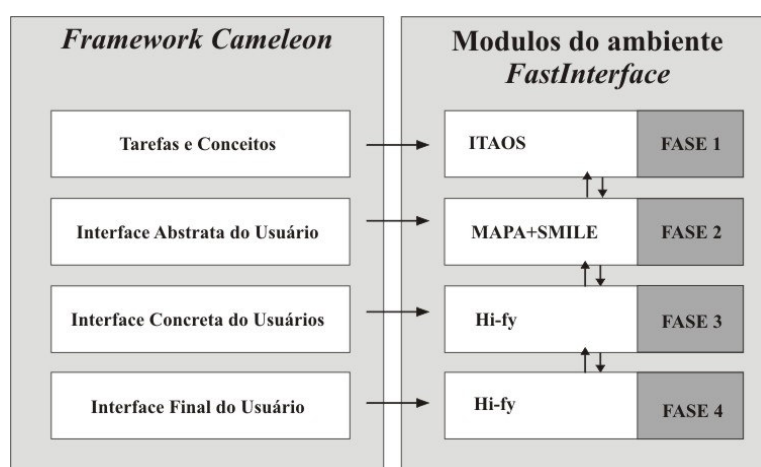


Figura 3. Módulos de *FastInterface* e fases de MEDITE sob o prisma de *Cameleon*.

4. Validação da Abordagem Metodológica Ferramental Proposta

O propósito da disciplina Laboratório de Engenharia de Software (LES) do curso de Bacharelado em Ciência da Computação da UFCG é o ensino de procedimentos e técnicas consolidadas de *design* de software, em que o aprendizado é reforçado pela vivência real de concepção, desenvolvimento, produção e validação de um produto de software. Em geral, esses produtos possuem a característica de serem utilizados no atendimento às demandas autênticas de software, possuindo, portanto, clientes e usuários reais que interagem diretamente com os alunos. O perfil dos alunos de LES é o de estudantes de 3º grau que possuem um conhecimento teórico inicial de técnicas e métodos da ES. O componente principal que orienta todo o laboratório de software é o processo YP. As demandas de software são distribuídas entre grupos de alunos que devem desenvolver cooperativamente todas as etapas recomendadas pelo processo, desempenhando, rotativamente, entre as iterações previstas, papéis de gerência e de desenvolvimento, durante o período de um semestre letivo.

A fim de verificar se o uso do YP diante da proposta ora descrita poderia ser realmente produtora, no sentido de ampliar o conhecimento, a capacidade e a habilidade dos alunos em lidar com questões de usabilidade em seus projetos de software utilizando YP, foi realizado um estudo com os alunos de LES do semestre

letivo 2008.2. As hipóteses levantadas para fundamentar o estudo e nortear o formato da metodologia de aplicação do experimento foram de que o apoio metodológico ferramental proposto: (i) proporciona maior facilidade de produção dos artefatos de usabilidade nas práticas propostas (agilidade); (ii) leva a uma maior compreensão da importância da produção dos artefatos no processo de desenvolvimento (fortalece aspectos teóricos de ES relativos à funcionalidade e usabilidade); (iii) auxilia na melhoria da comunicação entre a equipe de desenvolvimento (alunos de ES) e os clientes e usuários dos produtos em desenvolvimento (aprendizagem colaborativa); (iv) proporciona uma maior satisfação com a aplicação do processo nas práticas propostas (motivação e habilidade); e, (v) traz uma maior confiança na qualidade dos artefatos produzidos (usabilidade).

4.1. Descrição do Estudo

O estudo foi realizado em uma turma de 36 alunos, onde foram selecionados e distribuídos seis projetos de software, replicados entre doze equipes de desenvolvimento independentes, em que metade executou projeto com YP em seu formato atual e outra metade trabalhou com a proposta aqui apresentada. Todos os participantes possuíam conhecimento teórico de ES e foram, inicialmente, apresentados ao processo YP. Apenas as equipes que utilizaram *MEDITE/FastInterface* foram treinadas com relação aos seus procedimentos. As equipes foram acompanhadas e avaliadas em três momentos: (i) planejamento, (ii) fechamento de iteração e (iii) fechamento de *release*, sendo os dois últimos recorrentes de acordo com o fluxo de YP. Durante o planejamento, fase de inicialização, as equipes elaboraram os artefatos, os validaram a partir de apresentações aos condutores do estudo e aos clientes/usuários, e os refinaram de acordo com os *feedbacks* recebidos.

Com o objetivo de avaliar o impacto do apoio metodológico ferramental proposto para YP, os alunos responderam a um questionário baseado no modelo para montagem de questionários USE, utilizado para sondagem indireta da satisfação subjetiva de indivíduos [Queiroz 2001]. O questionário buscou medir fatores relevantes, tais como: (i) esforço e satisfação no aprendizado e construção dos artefatos de usabilidade; (ii) atendimento dos artefatos de usabilidade às expectativas dos clientes e usuários; (iii) aprendizado colaborativo fruto da melhor interação entre desenvolvedores e clientes/usuários; e, (iv) correlação entre artefatos gerados (sejam estes funcionais ou de usabilidade). Os respondentes do questionário podiam assumir posição positiva, neutra ou negativa às afirmativas/questionamentos que compunham o questionário.

4.2. Resultados

A análise das respostas fornecidas pelas equipes que aplicaram o YP com apoio metodológico ferramental atual e por aquelas que utilizaram *MEDITE/FastInterface* resultou no Quadro 1a, composto pelos itens avaliados e a relação da avaliação positiva dos dois grupos em relação aos mesmos. Os itens 1, 2, 4, 5 e 7 apresentaram índices de concordância maior para os alunos que aplicaram a nova abordagem de apoio ao YP em comparação aos alunos que aplicaram o YP em sua forma atual, considerando valores percentuais de 67, 67, 87, 67 e 100 em contraste, respectivamente, com 17, 50, 17, 50 e 67. Para o item 3 houve uma melhor aceitação por parte dos usuários do YP em sua forma atual (65% contra 0%). No item 6 houve igualdade na avaliação realizada por

ambos os grupos. Ainda com base na aplicação do questionário, definiu-se uma correlação entre os itens que compuseram o instrumento de coleta de dados (aspectos investigados) e as hipóteses inicialmente levantadas. O resultado consiste no Quadro 1b, que evidencia quais itens confirmam quais hipóteses.

Quadro 1: Resultados do comparativo da aplicação do YP nas duas abordagens (a) e da relação entre itens avaliados e hipóteses levantadas para o estudo (b)

Id	Itens avaliados/Abordagens	a		b				
		Atual	Proposta	i	ii	iii	iv	v
1	Facilidade de aprendizado do formalismo para construção dos modelos		X	X			X	
2	Facilidade de reprodução dos requisitos de análise do sistema sobre os modelos		X	X	X		X	
3	Facilidade de aprendizado da ferramenta utilizada	X		X	X		X	
4	Facilidade de edição dos elementos da interface		X	X	X	X	X	X
5	Obtenção do protótipo da interface a partir do modelo de tarefas		X	X			X	X
6	Satisfação dos clientes/usuários em relação ao protótipo da interface	X	X			X		X
7	Influência dos artefatos de usabilidade na geração de artefatos funcionais		X		X	X	X	

Legenda: (i) Agilidade; (ii) Aspectos teóricos de ES; (iii) Aprendizagem colaborativa; (iv) Motivação e habilidade; e, (v) Melhoria no uso.

4.3. Análise dos Resultados

O comparativo entre a aplicação do YP com as duas abordagens mostra (Quadro 1a) um nível de satisfação maior por parte dos usuários que utilizaram o contexto MEDITE/*FastInterface*. Entre os 7 itens analisados, 5 foram favoráveis, 1 neutro e 1 desfavorável (uso de ferramenta), sendo este último justificável (*i*) pela liberdade dos alunos que utilizaram a abordagem atual em escolher ferramentas de prototipagem já conhecidas e (*ii*) por problemas enfrentados pelos usuários de *FastInterface* com seu mecanismo de ajuda. O impacto do uso do contexto MEDITE/*FastInterface* foi considerado positivo pois, de acordo com o Quadro 1b, houve confirmação das hipóteses levantadas o que evidencia um ganho na aplicação da solução proposta: maior agilidade em YP, melhor compreensão dos fundamentos teóricos e práticos da ES, maior participação colaborativa de usuários e clientes e uma maior motivação por parte dos alunos na sua utilização.

A observação do experimento permitiu que o empenho dos alunos fosse acompanhado (níveis de interesse, entusiasmo e de comprometimento com a aplicação do YP) a fim de se chegar a um comparativo, subjetivo e inicial, entre aqueles que utilizaram a abordagem atual e os que utilizaram a abordagem proposta. Embora as equipes de ambos os grupos tenham executado todas as fases do YP elaborando os artefatos requeridos a fim de obter o produto em desenvolvimento, foi perceptível o entusiasmo dos usuários de YP/MEDITE/*FastInterface* na construção dos artefatos em relação ao demais participantes do experimento. No entanto, ainda se faz necessário realizar alguns refinamentos na ferramenta *FastInterface* a fim de melhorar seu nível de usabilidade e os recursos de ajuda disponibilizados.

Outro aspecto observado diz respeito à participação dos clientes/usuários no processo de concepção do produto em desenvolvimento (projeto participativo). O uso de

múltipla prototipagem favoreceu a inclusão dos clientes/usuário no processo colaborativo de definição da interface do usuário. A discussão sobre modelo da tarefa foi decisiva para a definição dos requisitos do sistema, assim como a apresentação dos protótipos permitiu clientes/usuários preverem a interface final do sistema, antecipando as sugestões de melhoria e adaptação.

5. Conclusão

A motivação para a realização deste trabalho originou-se da observação do ensino prático da Engenharia de Software na UFCG diante da utilização do YP para concepção de produtos reais. Embora o YP se encaixasse como academicamente adequado, sua aplicação revelou certo desconforto por parte dos alunos na elaboração dos artefatos de usabilidade definidos pelo método, fazendo com que os mesmos classicassem o processo como “pesado”. O estudo realizado atribuiu tal desconforto a abordagem mais somativa (“o que fazer”) que formativa e crítica (“porquê/como fazer”) vinculada ao YP. Em se tratando do primeiro contato dos alunos com a aplicação prática dos conceitos de ES, a ausência de uma abordagem mais formativa poderia dificultar o processo ensino/aprendizagem.

A solução apresentada para o problema observado, se deu a partir da associação de uma abordagem formativa ao processo YP, fornecida a partir do contexto metodológico ferramental *MEDITE/FastInterface*. Tal abordagem auxilia a aplicação do YP na construção dos artefatos de usabilidade baseada em modelos e múltipla prototipagem. Estas características possibilitam que os alunos absorvam a idéia da relação existente entre os diferentes artefatos elaborados consolidando na prática o conhecimento de ES, assim como estimula a participação ativa dos clientes/usuários no desenvolvimento do produto de forma colaborativa. O uso de modelos proporciona ainda a automatização do mapeamento entre os elementos dos diversos modelos envolvidos o que traz praticidade e agilidade à construção e manutenção dos artefatos elaborados.

Os resultados obtidos apresentaram-se favoráveis para a solução proposta assim como evidenciaram a necessidade de realização de trabalhos futuros: (i) avaliação da usabilidade das interfaces concebidas a partir da realização de testes de usabilidade em laboratório mediante a observação de usuários reais interagindo com o produto desenvolvido; (ii) ajustes na ferramenta *FastInterface* a fim de melhorar os mecanismos de ajuda oferecidos; (iii) realização de outros experimentos a fim de coletar uma massa de dados mais significativas que possibilite uma análise estatística dos resultados alcançados com a abordagem utilizada.

Referências

- Aguiar, Y. P. C.; Lima, A. H. G.; Leite, F. L. J., *et al.*(2004) “easYProcess: Um Processo de Desenvolvimento para Uso no Ambiente Acadêmico”. In XII WEI - XXIV Congresso da Sociedade Brasileira de Computação, Salvador.
- Aguiar, Y. P. C.; Lula Jr, B; Lima, G. A. N.; *et al.* (2007) “SMILE: uma ferramenta computacional baseada em modelos para geração e manipulação de esboço”. Revista Princípa, Ano 11, Nº 15, páginas 30-39.

- Anderson J.; Fleek, F.; Garrity K.; and Drake, F. (2001) "Integrating Usability Techniques into Software Development". IEEE Software, vol.18, no.1, pp. 46-53.
- Antunes, H.; Seffah, A. and Djouab, R. (2001) "Comparing and Reconciling Usability-Centered and Use-Case Driven Requirements Engineering Processes". In IEEE: 2nd Australian User Interfaces Conference, Jan/Feb 01, Queensland, Australia.
- Calvary, G. ; Coutaz, J.; Thevenin, et. al. (2003) "A Unifying Reference Framework for Multi-Target User Interfaces". In *Interacting with Computer*, pp. 289–308.
- Constantine, L. L and Lockwood, L. A. D. (1999) "Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design". Addison –Wesley.
- Constantine, L. L. (2008) "Process Agility and Software Usability: Toward Lightweight Usage-Centered Design". Disponível em: <http://www.foruse.com/articles/agiledesign.htm>, último acesso em: Abril de 2008.
- Ferré, X. (2001) "Incorporating Usability into a Object Oriented Development Process". In INTERACT 2001. 8th IFIP, Shinjuku Tokyo, Japan, July 9-13.
- Ferré, X; Juristo, N and Moreno, A. M. (2005) "A.M.: Framework for Integrating Usability Practices into the Software Process". In: PROFES 2005. Proc. of 6th Int. Conf. on Product Focused Software Process Improvement.
- Medeiros, F.P.A; Lula, Jr. B. and Cordeiro, P. B. (2002) "A Graphical Tool to Support Task Description Using TAOS Formalism for UI Design". In Proc. of the 7th ERCIM Workshop, p. 45- 51.
- Oliveira, K. M. A.; Aguiar, Y. P. C.; B. L. Jr *et al.* (2007) "O uso de Modelos e Múltiplos Protótipos na Concepção de Interface do Usuário". Revista Princípios, Ano 11, Nº 15, pp. 15-29.
- Queiroz. E. R. de (2001) "Usabilidade de Interfaces com o Usuário". Tese de Doutorado, COPELE/CCT/UFPB, Campina Grande, Junho.
- Radle, K. and Young, S. (2001) "Partnering Usability with Development: How Three Organizations Succeeded". IEEE Software, vol.18, no.1, pp. 38-45, January/February.
- Rodrigues, C. E. C. L; Lula, B. Jr. and Suárez, P. R. (2005) "Using a script model to preserve the consistency within an UI design environment". In Proceedings of the 4th International Workshop on Task Models and Diagrams.
- Suárez, P. R.; Lula, B. Jr. and Barros, M. A. (2004) "Applying knowledge management in UI design process". In Proceedings of the 3rd annual conference on Task Models and Diagrams, ACM International Conference Proceeding Series; Vol. 86.
- Tavares, T.A and Leite, J.C., (2002) "ARFDIU-Um método para integrar análise de requisitos funcionais com o desin de interfaces de usuário usando UML e outros formalismos". In Proceedings of IHC 2002.
- Vasconcelos, C. R.; Garcia, F. P. and Turnell, M. F. Q. (2003) "Integrando Usabilidade e Engenharia de Software: um modelo para o desenvolvimento de sistemas centrado no usuário". In: WIHC-ES2003 - I Latin American Conference on Human-Computer Interaction, CLIHC, Rio de Janeiro.