

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

IMPACTO DO SUPORTE METODOLÓGICO E FERRAMENTAL
ORIENTADO A ASPECTOS DE USABILIDADE NO ENSINO PRÁTICO DA
ENGENHARIA DE SOFTWARE

GIUSEPPE ANTHONY NASCIMENTO DE LIMA

CAMPINA GRANDE – PARAÍBA
DEZEMBRO 2010

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**IMPACTO DO SUPORTE METODOLÓGICO E FERRAMENTAL ORIENTADO A
ASPECTOS DE USABILIDADE NO ENSINO PRÁTICO DA ENGENHARIA DE
SOFTWARE**

GIUSEPPE ANTHONY NASCIMENTO DE LIMA

Dissertação submetida à Coordenação de Pós-Graduação em Informática, do Centro de Engenharia Elétrica e Informática, da Universidade Federal de Campina Grande, como requisito parcial para obtenção do grau de mestre em Ciência da Computação (MSc).

PROF. BERNARDO LULA JÚNIOR, DSc.
(ORIENTADOR)

ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO
LINHA DE PESQUISA: ENGENHARIA DE SOFTWARE

CAMPINA GRANDE – PARAÍBA
DEZEMBRO 2010

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

- L732i Lima, Giuseppe Anthony Nascimento de.
Impacto do Suporte Metodológico e Ferramental Orientado à Aspectos de Usabilidade no Ensino Prático da Engenharia de Software / Giuseppe Anthony Nascimento de Lima. – Campina Grande, 2011.
149f. : il. color.
- Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientador: Prof. Dr. Bernardo Lula Jr.
Referências.
1. Engenharia de Software - Interfaces. 2. Ensino da Engenharia de Software. 3. Integração da Engenharia de Software com Interface Homem-Computador. 4. Metodologias de Interface Homem-Computador. 5. Ferramentas para Projeto de Interface. I. Título.

CDU – 004.41:004.5 (043)

**"IMPACTO DO SUPORTE METODOLÓGICO E FERRAMENTAL ORIENTADO EM
ASPECTOS DE USABILIDADE NO ENSINO PRÁTICO DA ENGENHARIA DE
SOFTWARE"**

GIUSEPPE ANTHONY NASCIMENTO DE LIMA

DISSERTAÇÃO APROVADA EM 20.12.2010



**BERNARDO LULA JUNIOR, Dr.
Orientador(a)**



**JOSÉ EUSTAQUIO RANGEL DE QUEIROZ, D.Sc
Examinador(a)**



**FRANCILENE PROCOPIO GARCIA, D.Sc
Examinador(a)**



**EVANDRO DE BARROS COSTA, D.Sc
Examinador(a)**

CAMPINA GRANDE - PB

***"O IMPORTANTE DA EDUCAÇÃO NÃO É O
CONHECIMENTO DOS FATOS, MAS DOS VALORES."***

(DEAN WILLIAM R. INGE)

***DEDICO ESTE TRABALHO AOS CURIOSOS,
PERSEVERANTES E PACIENTES.***

AGRADECIMENTOS

Parece difícil ter que apreender a criar novos vínculos e a pensar diferente, em especial, quando se vai para um novo lugar, para aprender a fazer novas coisas, com novas pessoas ao redor. Portanto, gostaria de agradecer a todos da UFCG pelo auxílio e esforço para o meu melhoramento pessoal e intelectual, em especial:

Ao mestre **Bernardo**, pelas sugestões e constante envolvimento desde a idealização à realização deste trabalho. Não posso me esquecer de agradecê-lo, também, pelos maduros conselhos e encorajamento no início de minha jornada como profissional.

Ao professor **Eustáquio Rangel**, que desde o início foi de uma solicitude incrível, esclarecendo dúvidas, disponibilizando subsídios que me auxiliaram nessa jornada de mestrado, além de seus oportunos “avisos”, quando mais precisei.

À professora **Francilene Garcia** meu muito obrigado, pelo interesse e confiança depositados, disponibilizando pacientemente e sem restrições seu ambiente de aula para realização das pesquisas. Se não fosse por seu comprometimento não seria possível a realização deste trabalho.

À professora **Fátima Vieira**, pelo jeito atencioso de tecer considerações sobre procedimentos de usabilidade e por sempre me deixar “uma vaga” quando precisei usar o LIHM.

À **Karol, Márcia e Yuska** pelo incentivo e total apoio desde a graduação.

Aos meus familiares de Campina Grande (primo **Sérgio** e tio **Cláudio**) por cederem “um canto” para eu me instalar na cidade. Todo dia era uma casa e comida diferente!

A toda a **galera do LIA, LES e LIHM**, por sempre me receberem com atenção e consideração: altas conversas, dicas e bons momentos de descontração.

Germano, obrigado por sempre abrir a porta dos laboratórios quando eu me esquecia das chaves. **Aninha**, não existe pessoa mais afetuosa e eficiente que você! Obrigado por tudo!

Aos colegas da Unidade Acadêmica de Informática, do Instituto Federal da Paraíba (IFPB), em especial ao professor **Fausto Véras** e ao professor **Lafayette Melo**, pelos incentivos. Também, meus sinceros agradecimentos aos **meus alunos** que sempre me apoiaram quando precisei me ausentar.

Ao meu **pai Marcos** e à **tia Helena** pelo amparo total, principalmente nas "horas críticas".

À Prof. **Claudiana**, por me assessorar em parte da execução dos experimentos e na compilação/levantamento de dados.

O presente trabalho de dissertação foi finalizado em cooperação com a **Hewlett-Packard Brasil Ltda.** e com recursos provenientes da Lei de Informática (Lei nº 8.248, de 1991).

RESUMO

Tradicionalmente, a Engenharia de Software (ES) tem priorizado o desenvolvimento e aplicação de metodologias e ferramentas com maior ênfase em aspectos funcionais e de planejamento do software. Esta conduta tem se refletido no ensino da ES, em que aspectos relativos à usabilidade do software, geralmente, são abordados em segundo plano. Apesar dos esforços de integração da ES com a área de Interface Homem-Computador (IHC), no sentido de ofertar procedimentos que contemplem o mínimo de recursos para projeto de usabilidade de interface, não é raro encontrar alunos que relatem dificuldades em produzir, compreender e utilizar tais recursos adequadamente. Sendo desejável que o laboratório de ES seja completo, abordando tanto aspectos funcionais quanto de usabilidade, a disciplina Laboratório de Engenharia de Software, da Universidade Federal de Campina Grande (UFCG), aperfeiçoou o processo de desenvolvimento ágil *easYProcess* para torná-lo centrado com o usuário, agregando a produção de artefatos para projeto de interface. Foram verificadas dificuldades dos alunos com a produção e manejo desses artefatos. Este trabalho investigou os benefícios que a inclusão de uma metodologia de concepção de interfaces, baseada na tarefa e suportada por ferramental de software, podem proporcionar ao laboratório de software, em que o ambiente da graduação da UFCG foi adotado como referencial.

Palavras-chave: Engenharia de Software – Interfaces, Ensino da Engenharia de Software com Interface Homem-Computador, Laboratório de Software, Metodologias e Ferramentas para Projeto de Interface.

ABSTRACT

Traditionally, Software Engineering (SE) has prioritized the development and application of methodological and tooling support with major focus in functional and planning aspects of software development. This behavior has been reflected in SE teaching where aspects associated with software usability are addressed in a secondary plan, in general. Despite efforts to integrate SE with Human-Computer Interaction (HCI) area, aiming to offer procedures for minimal interface usability treatment, is not rare to find students reporting difficulties in production, comprehension and use of these resources properly. It is desirable that the SE laboratory approach be completed addressing both functional and usability issues. The Laboratory of Software Engineering discipline at Federal University of Campina Grande (UFCG) improved the agile development process *easYProcess* to turn it to be user-centered, by adding usability artifacts to its production. Difficulties have been detected with students during the production and use of these artifacts. This work investigated which benefits the inclusion of an interface conception methodology, based on task analysis and supported by a software tool, may provide to software laboratory, in which the UFCG graduation environment was adopted as referential.

Keywords: Software Engineering – Interfaces, Software Engineering Teaching with Human-Computer Interaction, Software Laboratory, Interface Design Methodologies and Tools.

SUMÁRIO

CAPÍTULO 1.....	11
CONSIDERAÇÕES INICIAIS	11
1.1. RELEVÂNCIA DE PESQUISAS NO CONTEXTO DE ES	11
1.2. O CARÁTER INTERDISCIPLINAR DO ENSINO DA ES E SUAS DIFICULDADES.....	12
1.3. A ABORDAGEM SITUACIONAL NO ENSINO DA ES.....	13
1.4. INTERDISCIPLINARIDADE: O CASO ESPECÍFICO DA INTEGRAÇÃO DE IHC NO ENSINO DA ES.....	15
1.5. A PRÁTICA DE ES COM TÉCNICAS DE IHC NA INDÚSTRIA DE SOFTWARE	18
1.6. DEFINIÇÃO DO PROBLEMA.....	19
1.7. OBJETIVOS DO TRABALHO.....	25
1.7.1. OBJETIVO GERAL	25
1.7.2. OBJETIVOS ESPECÍFICOS.....	26
1.8. RELEVÂNCIA.....	26
1.9. METODOLOGIA DE DESENVOLVIMENTO.....	27
1.10. ESTRUTURA DA DISSERTAÇÃO.....	28
CAPÍTULO 2.....	30
APRESENTAÇÃO DE YP E DA PROPOSTA DE SUPORTE METODOLÓGICO E FERRAMENTAL PARA O TRATAMENTO DA USABILIDADE	30
2.1. EASYPROCESS: UM PROCESSO ÁGIL, COMPLETO E ACADEMICAMENTE CORRETO.....	30
2.2. LOCALIZANDO O PROBLEMA DO MANEJO DA USABILIDADE EM YP.....	34
2.3. MEDITE: UMA ALTERNATIVA DE SUPORTE METODOLÓGICO PARA CONCEPÇÃO DE INTERFACES EM YP.....	36
2.3.1. MODELO DA TAREFA	39
2.3.2. MODELO DA INTERAÇÃO.....	40
2.3.3. PROTÓTIPO DE MÉDIA-FIDELIDADE	41
2.3.4. PROTÓTIPO DE ALTA-FIDELIDADE.....	41
2.4. FASTINTERFACE: SUPORTE FERRAMENTAL À MEDITE	42
2.5. SÍNTESE DE FATOS DO CONTEXTO ATUAL DE YP COM USABILIDADE E PERSPECTIVAS DE MELHORIA	44
CAPÍTULO 3.....	47
AVALIAÇÃO DE YP COM MEDITE + FASTINTERFACE: PLANEJAMENTO, CONFIGURAÇÃO E RESTRIÇÕES	47
3.1. AMBIENTE DE AVALIAÇÃO.....	47
3.2. EXPERIMENTO CONTROLADO	50
3.2.1. OBJETIVOS.....	50

3.2.2.	ARTEFATOS FONTES DE DADOS.....	51
3.2.3.	MÉTRICAS.....	52
3.2.4.	DEFINIÇÃO DAS HIPÓTESES	54
3.2.5.	VARIÁVEIS INDEPENDENTES	56
3.2.6.	VARIÁVEIS DEPENDENTES	57
3.2.7.	TRATAMENTO.....	57
3.2.8.	SUJEITOS E AMOSTRAGEM.....	57
3.2.9.	INSTRUMENTAÇÃO	59
3.2.10.	AVALIAÇÃO DA VALIDADE.....	60
3.2.11.	CONSIDERAÇÕES SOBRE FINAIS SOBRE A CONFIGURAÇÃO DO EXPERIMENTO.....	62
CAPÍTULO 4.....		64
APRESENTAÇÃO E ANÁLISE DE RESULTADOS		64
4.1.	VARIÁVEIS QUANTITATIVAS OU INDICADORES OBJETIVOS	64
4.2.	VARIÁVEIS QUALITATIVAS OU INDICADORES SUBJETIVOS.....	69
CAPÍTULO 5.....		78
CONSIDERAÇÕES FINAIS		78
5.1.	REVISÃO DA PROBLEMÁTICA DA FALTA DE APOIO METODOLÓGICO E FERRAMENTAL PARA USABILIDADE NO LABORATÓRIO DE SOFTWARE	78
5.2.	CONCLUSÕES QUANTO AO IMPACTO METODOLÓGICO E FERRAMENTAL SUGERIDO A PARTIR DO EXPERIMENTO	79
5.3.	CONSIDERAÇÕES ESPECÍFICAS À RESPEITO DA DISCIPLINA LABORATÓRIO DE ENGENHARIA DE SOFTWARE DA UFCG	81
5.4.	TRABALHOS FUTUROS	84
APÊNDICE A – TESTE T DE DUAS AMOSTRAS PRESUMINDO VARIÂNCIAS DIFERENTES PARA VARIÁVEIS QUANTITATIVAS		93
ANEXO A – PESQUISA SOBRE USO DE TÉCNICAS DE IHC NO ENSINO DA ENGENHARIA DE SOFTWARE.....		98
ANEXO B – QUESTIONÁRIOS ALUNOS.....		102
ANEXO C – QUESTIONÁRIOS CLIENTES/USUÁRIOS.....		110
ANEXO D – TUTORIAL FASTINTERFACE.....		115
ANEXO E – GUIA PARA ALUNOS: MANIPULAÇÃO DE ARTEFATOS DE USABILIDADE EM YP		126
ANEXO F – GUIA PARA PROFESSORES: APLICAÇÃO DO SUPORTE METODOLÓGICO E FERRAMENTAL DE USABILIDADE NO CONTEXTO DE LABORATÓRIO DE SOFTWARE		142

LISTA DE FIGURAS

FIGURA 1 - A VISÃO DE IHC E DE ES SOBRE DESENVOLVIMENTO DE SOFTWARE	17
FIGURA 2 – QUESTIONAMENTOS E POSICIONAMENTOS COMUNS DOS ALUNOS RELATIVOS AO MANEJO DA USABILIDADE	24
FIGURA 3 – FLUXO DO PROCESSO DE DESENVOLVIMENTO EASYPROCESS	33
FIGURA 4 – ADEQUAÇÃO DE MEDITE EM YP.....	35
FIGURA 5 – FLUXO DE MEDITE	37
FIGURA 6 – COBERTURA DOS MÓDULOS DE <i>FASTINTERFACE</i> EM MEDITE, SOB O PRISMA DE <i>CAMELEON</i>	44
FIGURA 7 – CONJUNTOS DE ARTEFATOS PARA COLETA DE DADOS	52

LISTA DE QUADROS

QUADRO 1 – SÍNTESE DOS PROBLEMAS E RELAÇÕES NO EMPREGO DE ARTEFATOS DE USABILIDADE EM YP	23
QUADRO 2 – COMPARATIVO DE EXPECTATIVAS ENTRE AS ABORDAGENS COM E SEM <i>MEDITE+FASTINTERFACE</i>	45
QUADRO 3 – NÍVEIS DA INFORMAÇÃO COM E SEM <i>MEDITE+FASTINTERFACE</i>	48
QUADRO 4 – HIPÓTESES NULAS	55
QUADRO 5 – AMOSTRAGEM DOS GRUPOS DE TRATAMENTO E DE CONTROLE	59
QUADRO 6 – PARÂMETROS PARA NÃO REJEIÇÃO DE HIPÓTESES NULA E ALTERNATIVA	60

LISTA DE GRÁFICOS

GRÁFICO 1 – FACILIDADE DE REPRODUÇÃO DE REQUISITOS COM MODELO DE TAREFAS	69
GRÁFICO 2 – COMPREENSÃO DO MODELO DE TAREFAS PELAS EQUIPES.....	70
GRÁFICO 3 – NÍVEL DE ESFORÇO NA CONSTRUÇÃO DOS ARTEFATOS DE USABILIDADE.....	70
GRÁFICO 4 – SEGURANÇA EM REPRODUZIR O MODELO DE TAREFAS AO TÉRMINO DO PROJETO DE SOFTWARE	71
GRÁFICO 5 – COMPREENSÃO DO MODELO DE TAREFAS DA INTERFACE PELOS CLIENTES	72
GRÁFICO 6 – CONTRIBUIÇÃO DO MODELO DE TAREFAS NA COMPREENSÃO FUNCIONAL DO PROJETO PELOS CLIENTES	72
GRÁFICO 7 – ATENDIMENTO DO PROTÓTIPO DE INTERFACE ÀS EXPECTATIVAS DO CLIENTE	73
GRÁFICO 8 – UTILIDADE DO MODELO DE TAREFAS E PROTÓTIPO DE INTERFACE NO PROJETO PARA O CLIENTE	73
GRÁFICO 9 – IMPACTO DO MODELO DE TAREFAS E DO PROTÓTIPO NA COMUNICAÇÃO DAS EQUIPES.....	74
GRÁFICO 10 – IMPACTO DO MODELO DE TAREFAS NA DECOMPOSIÇÃO FUNCIONAL DO SOFTWARE PELAS EQUIPES.....	75
GRÁFICO 11 – IMPACTO DO MODELO DE TAREFAS NA COMPREENSÃO DE US E TAs.....	75
GRÁFICO 12 – IMPACTO DO MODELO DE TAREFAS NO PLANEJAMENTO DE ITERAÇÕES E RELEASES	76
GRÁFICO 13 – INFLUÊNCIA DO MODELO DE TAREFAS NA CONCEPÇÃO DO PROTÓTIPO DE INTERFACE.....	76

LISTA DE SIGLAS E ABREVIATURAS

AM	Agile Modeling
ES	Engenharia de Software
IHC	Interface Homem-Computador
LES	Laboratório de Engenharia de Software
MCI	Método de Concepção de Interface
PDS	Processo de Desenvolvimento de Software
RUP	Rational Unified Process
TA	Teste de Aceitação
UCD	User-Centered Design
UFCG	Universidade Federal de Campina Grande
US	User Storie
XP	eXtreme Programming
XPU	eXtreme Programming with Usability
YP	easYProcess

CAPÍTULO 1

CONSIDERAÇÕES INICIAIS

1.1. RELEVÂNCIA DE PESQUISAS NO CONTEXTO DE ES

A Engenharia de Software (ES) tem procurado se adequar a exigências atuais de desenvolvimento de software, tais como: (i) alto número de projetos; (ii) escassez de tempo; (iii) foco na qualidade; e (iv) gestão dos fatores humanos de software. Essa adequação exige o esforço contínuo na busca de melhores práticas de ensino-aprendizagem, seja na experimentação de novas abordagens de ensino ou na atualização dos currículos, a partir da inserção de tendências oriundas de pesquisas acadêmicas ou da indústria de software, nos cursos de ES.

Hislop (2009) afirma que, para suprir a demanda crescente pela formação de novos desenvolvedores alinhados a essas tendências, as quais muitas vezes transcendem o currículo, os cursos de ES enfrentam novos desafios para a obtenção de resultados mais efetivos no processo de ensino-aprendizagem da área.

Na tentativa de apontar os principais desafios do ensino da ES, Lethbridge et. al. (2007) destacam que a comunidade deve ter a responsabilidade de concentrar esforços no sentido de melhorar a educação dos futuros engenheiros de software. O autor enumera as seguintes ações para a superação dos desafios: (i) tornar os programas mais atrativos para os estudantes; (ii) focar na educação da ES e melhoria de suas práticas; (iii) aplicar tendências na indústria de forma mais efetiva; e (iv) definir currículos que cubram ferramentas e

métodos inovadores para uma boa prática da ES. O autor ainda afirma que a superação destes desafios pode ser alavancada a partir de maiores incentivos às pesquisas na área de educação da ES, inclusive assegurando que os estudos relacionados sejam mais empíricos¹ para permear a formulação de novas teorias em suas práticas de ensino.

1.2. O CARÁTER INTERDISCIPLINAR DO ENSINO DA ES E SUAS DIFICULDADES

No Brasil, a Sociedade Brasileira de Computação mediante o documento "*Grandes Desafios da Pesquisa em Computação no Brasil*" (SBC, 2006) indica que o incentivo à interdisciplinaridade, exemplificada pela cooperação entre especialistas das diversas subáreas da computação, atua como um dos fatores de grande relevância na superação de problemas da área.

Dentre os fatores que dificultam a superação deste desafio se pode tomar como exemplo o caráter multidisciplinar do ensino da ES. Por ser uma disciplina pertencente às engenharias, esta deve buscar, constantemente, inclusive com colaborações de outras áreas do conhecimento, os subsídios necessários à produção de aplicações de software de alta qualidade com uma melhor relação custo-benefício. Isto abrange a seleção e modo de aplicação de teorias, técnicas, métodos, processos, métricas e ferramentas de apoio durante o ensino da ES.

Com o objetivo de discutir diretrizes curriculares para aplicação da ES, contextualizada com outras áreas, Braga (2009) realizou um estudo no qual evidencia vários trabalhos envolvendo o uso de novas metodologias de ensino, que utilizam estudos de caso, emprego de

¹ Estudos não somente baseados em heurísticas, mas principalmente baseados em evidências, ou seja, a partir de observações práticas.

ferramentas de suporte computacional, jogos e aprendizado colaborativo². Ao mesmo tempo, a autora enumera que, dentre as dificuldades presentes no campo do ensino, provavelmente inerentes ao caráter multidisciplinar da ES, têm-se como principais os seguintes problemas: (i) muito conteúdo sendo dado em pouco tempo; (ii) pouca motivação que os alunos de Ciência da Computação possuem para estudar os conceitos teóricos de ES; (iii) dificuldades em preparar os estudantes para a prática profissional dentro de ambientes acadêmicos.

Em síntese, podemos tirar destas constatações que a vasta abrangência da ES geralmente contribui para a geração de dúvidas na consideração do que deve ser abordado na academia, tanto da parte de quem ensina, ou seja, o que deve ser abordado e de qual forma; quanto de quem aprende, ou seja, quais aspectos de quais áreas são considerados relevantes após o aprendizado (Wangenheim & Silva, 2009).

1.3. A ABORDAGEM SITUACIONAL NO ENSINO DA ES

A educação em Engenharia de Software ao longo dos anos tem incorporado novas abordagens e técnicas visando à melhoria na qualidade de produção de software com a integração entre disciplinas, uso jogos para abordagem teórica, aplicação de projetos reais de software, dentre outras opções (Silva *et. al.*, 2008).

O ensino da ES utilizando trabalhos práticos é um fator de motivação para os alunos (Soares, 2008). Navarro e Hoek (2009) sugerem que para que haja uma efetiva aprendizagem de ES, seu ensino deve se utilizar de abordagens situacionais, em que a idéia seja a de aprender executando atividades ou desempenhando papéis de desenvolvimento, em ambientes realísticos ou que ao menos simulem o desenvolvimento

² Aprendizado de conceitos de forma prática a partir da troca de experiências com outros alunos, exemplificada pela experiência e aprendizado adquiridos no desenvolvimento de atividades em equipe ou em grupo.

de uma aplicação de software, promovendo a: (i) **atenção**, no sentido de permear o engajamento do aprendiz; (ii) **relevância**, ou seja, a compreensão dos alunos sobre a importância e aplicabilidade do que foi aprendido em situações reais; (iii) **confiança**, obtida a partir do momento em que o aprendiz percebe que há oportunidade de obter sucesso no tratamento das situações, muitas vezes não triviais, que o desenvolvimento de software exige; (iv) **satisfação**, promovida na medida em que os alunos possam notar as conseqüências positivas da aplicação do conhecimento/habilidade adquiridos em uma configuração realística ou simulada de desenvolvimento.

A premissa para o uso de abordagens mais situacionais de ensino se baseia em fatos como os de que os alunos egressos de cursos e disciplinas de ES devem demonstrar habilidades para gerenciar, analisar, projetar, verificar, validar, implementar, manter sistemas e projetos de software (Castro, Gimenes & Maldonado, 2000).

Considerando a abordagem situacional inserida numa configuração realística de desenvolvimento de software e o conhecimento multidisciplinar envolvido, é inevitável que problemas de interesse e aprendizado ocorram, especialmente quando determinadas práticas são realizadas pela primeira vez pelo aluno.

Portanto, surge a seguinte pergunta: *quais critérios adotar para se eleger uma conduta dentro de uma abordagem situacional de ensino de ES?* Estudos de caso e jogos podem ser bastante utilizados, todavia não propiciam uma vivência integrada entre as diversas etapas de desenvolvimento de uma aplicação de software. Em um laboratório, a partir do desenvolvimento de projetos de software completos, com demandas e preocupações reais, cria-se um ambiente mais propício à aplicação da abordagem situacional. A implantação de um laboratório de software pode ser idealizada a partir da seleção de um processo de desenvolvimento de software (PDS) que propicie uma experiência de

desenvolvimento da maneira mais completa possível (Lima, Aguiar & Lula, 2009).

1.4. INTERDISCIPLINARIDADE: O CASO ESPECÍFICO DA INTEGRAÇÃO DE IHC NO ENSINO DA ES

Considerando o caso específico de integração da ES com a área de IHC, no sentido de extrair melhores alternativas e técnicas de projeto de interface, o alvo de questionamentos seria o que considerar como praticável dentro do ensino da ES. São muitas as propostas no sentido de fornecer um relacionamento sinérgico entre práticas de projeto funcional de ES e práticas de projeto de interface e de usabilidade, oriundos de IHC. Podemos tomar como exemplo principal o surgimento de processos de desenvolvimento de software que dispõem de métodos para realização de um *design* mais centrado no usuário (em inglês *User Centered Design*, UCD) (GOMES *et al.*, 2007).

O design centrado no usuário traz uma nova visão sobre a concepção de produtos em geral tratando-os sob o ponto de vista do usuário e de como atender às suas necessidades de maneira satisfatória. Segundo Nielsen (Nielsen *et al.*, 1992), a usabilidade pode ser compreendida e mensurada a partir de cinco atributos: (i) facilidade de aprendizado, (ii) eficiência de uso, (iii) facilidade de memorização, (iv) baixa taxa de erros; e (v) satisfação com o produto de software.

Gomes *et al.* (2007) afirmam que a formação de engenheiros de software é prioritariamente voltada ao desenvolvimento de habilidades relacionadas à implementação e uso de tecnologias, sendo categórico: “*Essa estratégia torna o senso de autocrítica limitado e compromete a busca por uma solução mais inovadora, competitiva, e com foco no mercado*”.

Ainda, se aponta que o desenvolvimento de competências para projeto de produtos inovadores³ e centrados no usuário em cursos de graduação no Brasil ainda é muito limitado e afirma

[...] as experiências em importantes universidades estrangeiras refletem uma maior preocupação com a abordagem interdisciplinar na formação de competências para atuar na área [a exemplo da ES com IHC] (GOMES et al., 2007).

Considerando as demandas da indústria de software por produtos inovadores e que satisfaçam a seus clientes, *por que não inserir práticas de desenvolvimento centradas no usuário no ensino da engenharia de software?* Considerando os benefícios da abordagem situacional de ensino de ES, então, reformula-se a questão: *porque não inserir um processo de desenvolvimento de software centrado no usuário no contexto do laboratório de software?*

Na tentativa de buscar respostas para a questão, uma breve pesquisa do tipo *survey* (Anexo A) foi realizada entre junho e agosto de 2010, com 30 professores de cursos de computação, em instituições representativas na região Nordeste (destaque para UFPE, UFCG, UFRN, UFPB, Unifor, IFPB).

Quando perguntados sobre a importância de se abordar conceitos de projeto de usabilidade e interface de forma contextualizada em disciplinas de ES, apenas 4% discordaram sobre a relevância de se abordar tais conceitos. No entanto, quando indagados sobre a crença na boa compreensão de seus alunos sobre técnicas de projeto de

³ Gomes et al. (2007) se referem à inovação como produto da criatividade, em que o uso de métodos para o projeto de usabilidade de interfaces pode contribuir na ampliação da visão criativa do engenheiro de software, auxiliando na criação de melhores formas de interação.

interface/usabilidade, 26% se posicionaram neutros, 41% discordaram e 7% discordaram totalmente.

Os professores também foram perguntados sobre os principais motivos para a dificuldade em se realizar um ensino de ES com uma ênfase básica em técnicas de projeto de interface/usabilidade. Neste caso, 70% apontaram questões culturais no ensino da ES, pois muitos associam IHC como uma área opcional/especializada; 19% indicaram a complexidade das técnicas de IHC; e 44% escassez de tempo, pois outros assuntos relativos ao funcional do software como gerência, planejamento, testes e arquitetura inevitavelmente acabam sendo mais priorizados. Alguns, inclusive, apontaram que muitos colegas desconhecem a importância das soluções propostas pela área de IHC para desenvolvimento de software com usabilidade. Sobre a questão cultural na integração de ES com IHC é importante contrastar a visão das duas áreas sobre o desenvolvimento de software (Figura 1). Indiretamente, há um acirramento entre o projeto funcional e o projeto da usabilidade (Silva *et al.*, 2005).

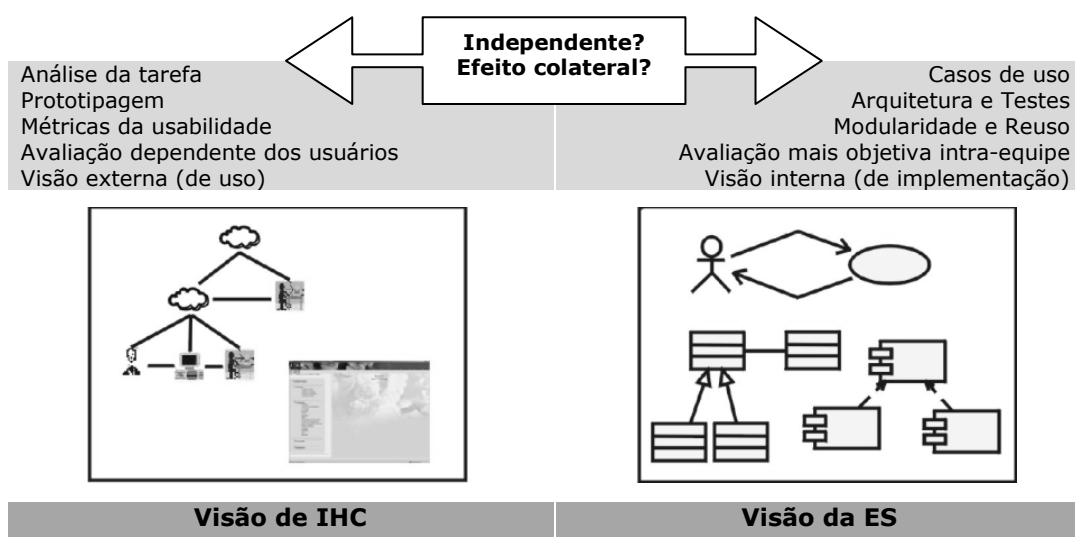


Figura 1 - A visão de IHC e de ES sobre desenvolvimento de software

Tal acirramento, de certa forma reproduzido na pesquisa anteriormente mencionada, pode ser justificado pelas afirmações de Constantine e Lockwood (1999) sobre o tema: a ES tradicionalmente tem encorajado a construção de sistemas com foco no desenvolvimento de funcionalidades, ou seja, com ênfase no processamento lógico correto e desempenho adequado. Neste caso, as métricas de qualidade se restringem a fatores como reusabilidade, portabilidade, confiabilidade, modularidade, acoplamento e robustez. Portanto, os estudantes têm a tendência a desenvolver melhor unicamente a visão de ES supracitada e a relativamente depreciar a visão de IHC.

1.5. A PRÁTICA DE ES COM TÉCNICAS DE IHC NA INDÚSTRIA DE SOFTWARE

O problema de se encontrar uma harmonia entre a prática de ES com técnicas consistentes de IHC é refletido também na indústria de software, onde persiste a crença de que o emprego de técnicas de usabilidade na confecção de produtos implica em um significativo atraso no cronograma de projetos (Ferré, 2001). A usabilidade é vista como algo subjetivo e especializado demais e, talvez, por isso os profissionais tendem a acreditar que muito esforço deva ser desenvolvido para tratá-la.

Em breve pesquisa realizada com as empresas mais representativas do reconhecido pólo de desenvolvimento de tecnologia da informação e comunicação do estado de Pernambuco, o Porto Digital, Bernardino (2005) verificou que das principais empresas de pequeno e médio porte incubadas poucas são as que trabalham junto a seus usuários finais. A autora acredita que esta cultura poderia ser diferente se demonstrações e experimentações mais claras sobre a adequação de metodologias centradas no usuário, ou no *design* interativo, pudessem ser realizadas em projetos reais. Ainda, ela aponta que roteiros e guias poderiam ser

desenvolvidos para contribuir na adequação de metodologias centradas no usuário em projetos de software executados por empresas.

Porém, *qual a empresa se disponibilizará no experimento de adequação de metodologias, principalmente considerando a dificuldade em se trabalhar aspectos humanos no projeto de software?* Certamente, considerando os riscos envolvidos no cumprimento de prazos e custos de projetos a resposta será quase nenhuma empresa.

Adequar uma metodologia de qualquer natureza implica em muitas intervenções e prováveis transtornos. Desta forma, acredita-se que a academia deva ser o meio responsável por apontar sugestões em favorecimento de uma cultura de usabilidade objetiva, mínima e praticável, que possa contribuir na formação de profissionais mais responsáveis no âmbito de projeto de uso e interação com sistemas.

1.6. DEFINIÇÃO DO PROBLEMA

Ao considerar o software como sendo composto por duas partes básicas: (i) **a parte funcional**, que consiste em todo o aparato que permite a implementação das funcionalidades do sistema, e (ii) **a parte interativa**, que possibilita que os usuários do sistema utilizem as funcionalidades oferecidas, é preciso que processos de desenvolvimento de software compreendam tanto etapas relativas às funcionalidades (perspectiva interna) quanto às da usabilidade (perspectiva externa) do sistema com a mesma importância. O ideal é que ambos os conceitos sejam considerados, havendo uma relação de completude, inclusiva e não excludente (Cybis *et. al.*, 1999).

Considerando a efetividade que o ensino a partir de uma abordagem situacional oferece, a necessidade de inserir boas práticas de projeto de interface e a complexidade de se conciliar tudo isso, formula-se a seguinte pergunta: *como realizar uma prática em laboratório de*

software simulando um desenvolvimento completo a partir de demandas de clientes reais e abordando questões relativas à usabilidade, numa disciplina prática de ES?

Avaliando a evolução que a disciplina Laboratório de Engenharia de Software (LES) da UFCG adquiriu nos últimos anos, sugere-se o uso de um processo de desenvolvimento de software academicamente adequado. Entende-se por academicamente adequado um processo que considere (Lima, Aguiar & Lula, 2009): (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 consideradas *estado da arte*; (iv) o aprendizado colaborativo, com promoção da comunicação intra-equipe e com os clientes e usuários; (v) a agilidade, como consequência direta do tempo limitado, do escopo restrito e do aprendizado colaborativo; (vi) a avaliação contínua e iterativa do progresso do projeto, para que o aprendizado dos alunos possa ser avaliado ao logo da disciplina. Além disto, com o advento, progresso e a recomendação (do uso) de novas técnicas de interação homem-computador (IHC), um PDS academicamente adequado também deve (vii) adequar práticas apoio ao projeto de interface e de usabilidade em seu ciclo de desenvolvimento.

Os esforços desenvolvidos em LES contribuíram para a constante melhoria do processo de desenvolvimento *easYProcess* (YP), que busca conciliar as melhores tendências em técnicas e práticas para um desenvolvimento de software ágil. YP é um PDS inicialmente concebido para ambientes acadêmicos e vem sendo empregado em LES desde 2003. O YP já esteve presente em disciplinas de cursos de informática e computação de outras instituições de ensino ligadas à UFCG por meio de seus ex-alunos. Instituições como a UEPB, IFPB, FIP e FATEC, que já

experimentaram YP em disciplinas relacionadas ao projeto e desenvolvimento de software.

A proposta de YP (easYProcess, 2006) é ser simplificado, iterativo e incremental, apoiando-se em boas práticas de XP (Beck, 2000), RUP (Jacobson, Booch & Rumbaugh, 1999) e *Agile Modeling* (Ambler, 2002).

O YP inicialmente não sugeria a construção de artefatos que apoiassem a concepção e validação de interfaces com o usuário. Os artefatos gerados com a aplicação do YP tinham como foco apenas a parte funcional do sistema, com levantamento de requisitos funcionais, projeto arquitetural, projeto lógico, testes de unidade e de aceitação, dentre outros. Para se aprimorar como processo centrado no usuário, o YP teve que se adaptar, passando a incluir os artefatos de usabilidade (Vasconcelos, Garcia, & Turnell, 2003).

Estes artefatos compreendem o conjunto de técnicas e ferramentas para a coleta, a especificação e a avaliação de alguns requisitos de usabilidade, tais como: caracterização dos usuários, estudo de tarefas efetuadas pelos mesmos, definição de ambientes de trabalho e de metas de usabilidade (eficiência, eficácia, satisfação, aprendizado, dentre outras).

No entanto, embora projetos derivados do YP tenham sido bem avaliados, tanto do ponto de vista funcional quanto de usabilidade e o aprendizado dos alunos tenha atingido um nível satisfatório, a quantidade de artefatos a serem gerados, devido à inserção de práticas de projeto de interface, tornou a aplicação do YP mais “pesada” do que o desejado.

Ademais, acompanhamentos e relatos obtidos na aplicação de disciplinas de ES, inclusive em outras instituições de ensino, apontaram que, em se tratando da geração e manipulação dos artefatos de usabilidade, os alunos contestam a relação custo-benefício no produto

final do software, evidenciando certa depreciação e falta de confiança nas práticas de usabilidade recomendadas pelo processo.

A partir disto, se faz necessário levantar as reais causas para a desconfiança nos ganhos com uso de artefatos de usabilidade e ainda apontar soluções que possibilitem o manejo desses artefatos sem comprometer a agilidade do processo.

Em outras palavras, é necessário encontrar uma maneira de motivar os usuários de YP em sua aplicação integral, com a construção de todos os seus artefatos, incluindo-se aqueles de usabilidade.

Em observações preliminares, foram levantadas algumas causas prováveis para o problema do esforço e desmotivação com o uso dos artefatos de usabilidade: (i) oferta de diversos métodos para a obtenção do modelo de tarefas e do protótipo de interface, muitas vezes complexos, ou com excesso de formalismos para um iniciante; (ii) falta de suporte ferramental adequado e integrado para desenvolvimento dos artefatos como o protótipo de interface e principalmente o modelo de tarefas, esse último muitas vezes acaba sendo abandonado; (iii) realização *ad-hoc* da validação e emprego dos artefatos de usabilidade, dada a falta de indicações claras sobre como gerar e utilizar estes artefatos com clientes, além da impossibilidade de se assistir estas etapas.

O Quadro 1 sintetiza os principais problemas e principais causas. Dentre as causas citadas, principalmente para a esporádica desistência em se elaborar o modelo de tarefas de qualidade, se destaca a impossibilidade de percepção de continuidade por parte dos alunos.

Considerando que os artefatos de usabilidade recomendados por YP e até em outros processos centrados no usuário, sugerem sua geração e manipulação logo após a fase de análise de requisitos, não fica evidente quais dos artefatos desenvolver primeiro. Este problema é agravado

quando se sabe que, após a análise de requisitos outros artefatos funcionais, também devem ser desenvolvidos, a exemplo das especificações de casos de uso, de testes de aceitação e de unidade, o projeto arquitetural e lógico, além do planejamento e escalonamento de atividades.

Quadro 1 – Síntese dos problemas e relações no emprego de artefatos de usabilidade em YP

		Principais Causas			
		Oferta de múltiplas metodologias de interface	Falta de suporte ferramental adequado	Realização <i>ad-hoc</i> de validação e uso	Impossibilidade de compreensão de formalismos de interface em tempo hábil
Problemas	Alto esforço na obtenção do modelo de tarefas e protótipo de interface	X	X		X
	Falta de compreensão do uso adequado dos artefatos de usabilidade	X	X		X
	Ausência de uso dos artefatos de usabilidade		X	X	X
	Insatisfação/descrência na aplicabilidade de procedimentos de usabilidade		X	X	X

A impossibilidade de se obter direcionamentos mais objetivos na construção de artefatos de usabilidade como o modelo de tarefas e o protótipo contribuem para afirmações do tipo: “tanto faz fazer um artefato antes ou depois do outro, ao mesmo tempo ou até mesmo abortar artefatos inconvenientes (obviamente os de usabilidade)”.

De qualquer forma, os artefatos de usabilidade devem ser bastante explorados na concepção de projetos de software com usabilidade nas fases antecessoras à implementação. Na ausência disto, acabam sendo mal elaborados e conseqüentemente inutilizados.

A Figura 2 demonstra os questionamentos e posicionamentos comuns dos alunos, a partir da observação de YP com usabilidade em LES, relativos ao manejo dos artefatos de usabilidade.

ALUNO x ARTEFATOS DE USABILIDADE

Modelo da tarefa? Protótipo? Serve para quê? Como fazer? Está certo? Qual dessas ferramentas e metodologias usar (não aprendi nada sobre elas)? Isso não é desnecessário ou redundante? Tem que atualizar?...

...Não tenho a menor noção e isso está dando muito trabalho! Farei de qualquer jeito é melhor me concentrar nos casos de uso, testes, projeto arquitetural e no planejamento! Se der tempo eu projeto a interface!




Figura 2 – Questionamentos e posicionamentos comuns dos alunos relativos ao manejo da usabilidade

O modelo de tarefa é geralmente bastante presente nos processos centrados no usuário, por ser um artefato que sintetiza a ordem lógica do uso das funcionalidades do sistema e auxilia na etapa de prototipagem da interface. Pode-se, com isto, pressupor que se o modelo de tarefas auxilia na compreensão do uso de funcionalidades do sistema e na obtenção visual desse uso, com o protótipo de interface. Conseqüentemente, tanto o modelo de tarefas quanto o protótipo podem auxiliar na compreensão do escopo, complexidade e implementação das partes do software. Sendo assim, esses artefatos poderiam supostamente contribuir para uma melhor gestão do conhecimento sobre as funcionalidades do sistema e conseqüentemente para uma maturação da análise de requisitos funcionais.

As experiências em LES com YP reproduzem, de certa forma, a problemática de se integrar procedimentos de projeto de interface e de usabilidade em disciplinas de ES e, porque não dizer também, em

processos de software. Em nível mais abrangente, tomando o mercado de software como exemplo, não é inoportuno presumir que a problemática em volta do manejo de artefatos de usabilidade em contextos ágeis de desenvolvimento possa se reproduzir.

Diante do exposto, algumas questões podem ser levantadas:

- *Por que o emprego de artefatos de usabilidade, sobretudo o modelo de tarefas e o protótipo de interface, tendem a não serem bem aceitos e usados adequadamente?*
- *Será que o emprego de uma metodologia de desenvolvimento de interfaces, com modelos simples, ágeis e implementáveis com suporte ferramental adequado às restrições acadêmicas e as expectativas dos alunos, não melhorariam a compreensão e aceitação de artefatos de usabilidade?*
- *Qual metodologia de concepção de interfaces poderia se adequar a este contexto?*
- *Como verificar o impacto e resultados da aplicação de uma metodologia adequada na minimização do problema? Há vantagens representativas?*

1.7. OBJETIVOS DO TRABALHO

1.7.1. OBJETIVO GERAL

O objetivo geral deste trabalho consiste em sugerir uma adequação das técnicas de IHC no laboratório de ES, presumindo que o aluno desse laboratório possa não ter conhecimento algum da visão de IHC. Desta forma, objetiva-se avaliar o impacto que essa adequação proporcionaria aos alunos a partir da condução de um experimento que toma como partida o processo YP, por ser centrado no usuário e por já se encontrar implantado no ambiente da disciplina LES da UFCG.

1.7.2. OBJETIVOS ESPECÍFICOS

- Verificar as causas para o problema de integração das visões de ES e IHC, considerando uma grande restrição: um processo de desenvolvimento ágil dentro de um contexto de uso para ensino em laboratório;
- Definir uma estratégia de adequação de técnicas de projeto de interface e usabilidade ao laboratório de software, a partir da inserção de uma metodologia de concepção de interfaces provida de suporte ferramental de software;
- Avaliar o impacto do suporte metodológico e ferramental proposto, tanto em nível de usabilidade quanto funcional, além da verificação da compreensão e satisfação dos alunos e clientes de projeto com o novo tratamento sugerido.

1.8. RELEVÂNCIA

As estratégias de adequação sugeridas neste trabalho para o tratamento da usabilidade buscam integrar melhor a visão de ES e a visão de IHC dentro do contexto de ensino, considerando um processo de desenvolvimento de software aplicado em âmbito acadêmico. Conseqüentemente, as principais contribuições deste trabalho são reproduzíveis ou impactadas a partir da:

- Disponibilização de guias com recomendações para o manejo de artefatos de usabilidade, inserida dentro de um processo ágil de desenvolvimento, utilizando técnicas e métodos consolidados de IHC, considerando a oferta de um suporte metodológico assistido por uma ferramenta de software adequada para tal.
- Aquisição de evidências práticas para demonstração dos possíveis ganhos em se tratar a usabilidade no desenvolvimento de software e na qualidade do produto final. Nisto, inclui-se a busca de

- evidências sobre as relações entre técnicas de projeto de interface e de usabilidade e o manejo de aspectos funcionais do software;
- Auxílio na fomentação da cultura de desenvolvimento centrada no usuário, ou de *design* interativo, demonstrando que a mesma pode ser descomplicada e praticável, inclusive em contextos ágeis de desenvolvimento. Aplicada no ensino prático da ES, a exemplo das sugestões a serem realizadas no laboratório de software da UFCG, essa cultura tem maiores chances de ser reproduzível inclusive transpassando as barreiras da academia;
 - Realização de pesquisas experimentais em educação de ES. Os estudos para avaliação da abordagem metodológica e ferramental para usabilidade proposta devem favorecer o compartilhamento dos procedimentos experimentais, que poderão ser observados, adaptados ou até replicados em outros momentos ou contextos de pesquisa;
 - Dar subsídios para o melhor desenvolvimento da disciplina LES da UFCG e de disciplinas de laboratório de software similares, a partir das observações realizadas durante a pesquisa;

1.9. METODOLOGIA DE DESENVOLVIMENTO

A metodologia adotada na realização deste trabalho de dissertação pode ser decomposta nas seguintes etapas, a saber:

- i. Revisão bibliográfica, a fim de identificar a existência de problemas na integração da visão IHC no contexto de ensino da ES, principalmente, considerando a abordagem prática/situacional do laboratório de software. A fim de verificar tendências para confirmação da existência desses problemas, foi realizada uma pesquisa do tipo *survey*, em âmbito regional;

- ii. Levantamento dos problemas de compreensão, satisfação e qualidade relativos ao manejo atual da usabilidade, usando o ambiente da disciplina LES da UFCG;
- iii. Elaboração de guia de aplicação para melhor manejo da usabilidade, considerando melhor facilidade em obter artefatos e uma sistemática de obtenção, edição e manutenção dos mesmos;
- iv. Elaboração de guia destinado ao professor, com recomendações e sugestões para uma melhor aplicação do suporte metodológico e ferramental de apoio a usabilidade. Ainda, esse guia sugeriu formas de como acompanhar e avaliar melhor os artefatos de usabilidade junto aos alunos e clientes de projeto de software;
- v. Elaboração e execução de um experimento controlado utilizando o ambiente da disciplina LES da UFCG. Tal estudo é de natureza comparativa, em que foram verificadas as diferenças entre a abordagem atual de manejo de artefatos de usabilidade (grupo de controle) e a nova abordagem, com suporte metodológico e ferramental a concepção de interfaces (grupo de tratamento). Esse experimento utilizou métricas quantitativas, relativas à aspectos de interface e funcionais; e qualitativas, relativas à questões subjetivas, como níveis de compreensão e satisfação dos alunos e de clientes de projeto com os procedimentos indicados pela nova abordagem.

1.10. ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está dividida em 5 capítulos. O Capítulo 1 contém informações necessárias ao entendimento do que representa a pesquisa, tais como a contextualização do problema da integração da

visão de IHC no ensino da ES, a definição do problema, os objetivos do trabalho, sua relevância e metodologia de realização.

O Capítulo 2 se destina a apresentar o processo desenvolvimento YP e seu contexto da disciplina LES da UFCG, que servirá para o estudo controlado. Neste capítulo, ainda, serão apresentados os problemas e suas respectivas localizações dentro do processo, além da apresentação do suporte metodológico e ferramental para manejo da usabilidade dentro do YP.

O Capítulo 3 trata do experimento controlado, apresentando suas características, restrições de validade e de configuração. Detalhes sobre hipóteses, as variáveis, suas medições e relações também serão abordadas neste capítulo.

O Capítulo 4 apresenta a análise dos resultados obtidos com a utilização da abordagem proposta para tratamento da usabilidade. Algumas inferências sobre o impacto da nova abordagem serão comentadas, tanto relativas às questões da qualidade do projeto de interface, quanto no desenvolvimento de outros artefatos funcionais, à da compreensão e satisfação dos alunos, comparativamente com a abordagem anterior.

No Capítulo 5 apresentam-se as conclusões e argumentos referentes da realização desta pesquisa, além de proposições para trabalhos futuros.

CAPÍTULO 2

APRESENTAÇÃO DE YP E DA PROPOSTA DE SUPORTE METODOLÓGICO E FERRAMENTAL PARA O TRATAMENTO DA USABILIDADE

Neste capítulo, apresentam-se os detalhes de YP antes e depois da inclusão de etapas para o tratamento da usabilidade. Alguns comentários sobre a origem de problemas nesta última configuração serão formulados, culminando com a necessidade de se incluir um suporte metodológico e ferramental. As devidas justificativas e ressalvas serão abordadas e como solução será apresentada a alternativa de adequação da metodologia de concepção de interfaces MEDITE dentro do processo.

2.1. EASYPROCESS: UM PROCESSO ÁGIL, COMPLETO E ACADEMICAMENTE CORRETO

Um processo de desenvolvimento de software ágil é aquele capaz de reagir rapidamente a mudanças e de agregar valor rapidamente ao cliente. Segundo Boehm (2002), para definir um processo como ágil, é necessário que: (i) os desenvolvedores estejam mais direcionados ao conhecimento adquirido de forma colaborativa do que orientados a registrar os conhecimentos existentes em forma de planejamento; (ii) os clientes colaborem diretamente com a equipe de desenvolvimento, trocando experiências e informações sem a necessidade explícita de uma documentação formal sujeita a má-compreensão; (iii) os requisitos possam sofrer mudanças, e não necessariamente sejam bem definidos e já consolidados (estáveis); (iv) a arquitetura do sistema seja

projetada para as necessidades atuais e não para prever modificações futuras que podem vir a não acontecer; (v) as equipes de desenvolvimento sejam pequenas e os produtos gerados também tenham escopo reduzido; e (vi) o custo com a refatoração seja baixo e atrativo.

O easYProcess (YP) consiste em um processo de desenvolvimento de software simplificado, apoiado em práticas de XP, AM e RUP, com o objetivo inicial de uso acadêmico que propicia a construção de aplicações de software de forma ágil. YP propõe a construção de artefatos necessários ao contexto acadêmico, no qual é preciso realizar avaliações constantes da evolução dos alunos na aplicação do processo, “simulando” a realidade fora da universidade.

Para a elaboração do YP, as práticas e os artefatos presentes nas metodologias XP, RUP e AM foram estudados, a fim de identificar quais práticas, regras e artefatos de cada um dos processos poderiam ser unidos, considerando as especificidades de uma aplicação acadêmica ou de ensino. 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. Com relação às etapas de implementação, são aproveitados os princípios de refatoração e integração contínuas, a propriedade coletiva do código, o uso de padrões de projeto, a programação em pares e o 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, lista de riscos e a coleta de métricas (*big chart* YP). AM forneceu a YP princípios de mudanças incrementais e com a idéia de aprendizado colaborativo, no qual todos podem aprender uns com os outros.

O fluxo de YP compreende as seguintes fases: (i) **Definição de papéis**, em que as habilidades e competências dos *stakeholders* do projeto são

descritas; (ii) **Conversa com o cliente**, que consiste na definição de uma visão sobre o processo de negócio, levantamento dos requisitos funcionais e não funcionais, do perfil do usuário e dos objetivos de usabilidade; (iii) **Inicialização**, que é composta das atividades de modelagem da tarefa, levantamento das *user stories*, testes de aceitação, geração do protótipo de interface, elaboração do projeto arquitetural e geração do modelo lógico de dados (quando necessário); (iv) **Planejamento**, na qual se define os planos de release e respectivas iterações; (v) **Implementação**, que consiste na construção do código da aplicação orientada pelo planejamento e respectiva especificação de testes. As Reuniões de Acompanhamento são determinantes na avaliação da implementação realizada durante as iterações e fundamentais para o planejamento das iterações e *releases* posteriores. E, por fim, (vi) disponibilização da Versão do Sistema/Produto. Na figura 3, vê-se o fluxo de YP (Figura 3), extraído da forma que é apresentado em seu site (easYProcess, 2006).

Com a inserção em seu fluxo de etapas, da produção de artefatos de usabilidade como o perfil do usuário, objetivos de usabilidade, análise da tarefa e prototipagem da interface, além de incorporar a avaliação contínua do cliente, inerentes ao projeto iterativo e incremental, o YP pode ser considerado um processo centrado no usuário. Logo, o YP é academicamente adequado, conforme as restrições descritas por Lima, Aguiar e Lula (2009), comentadas na seção 1.6, para as devidas considerações.

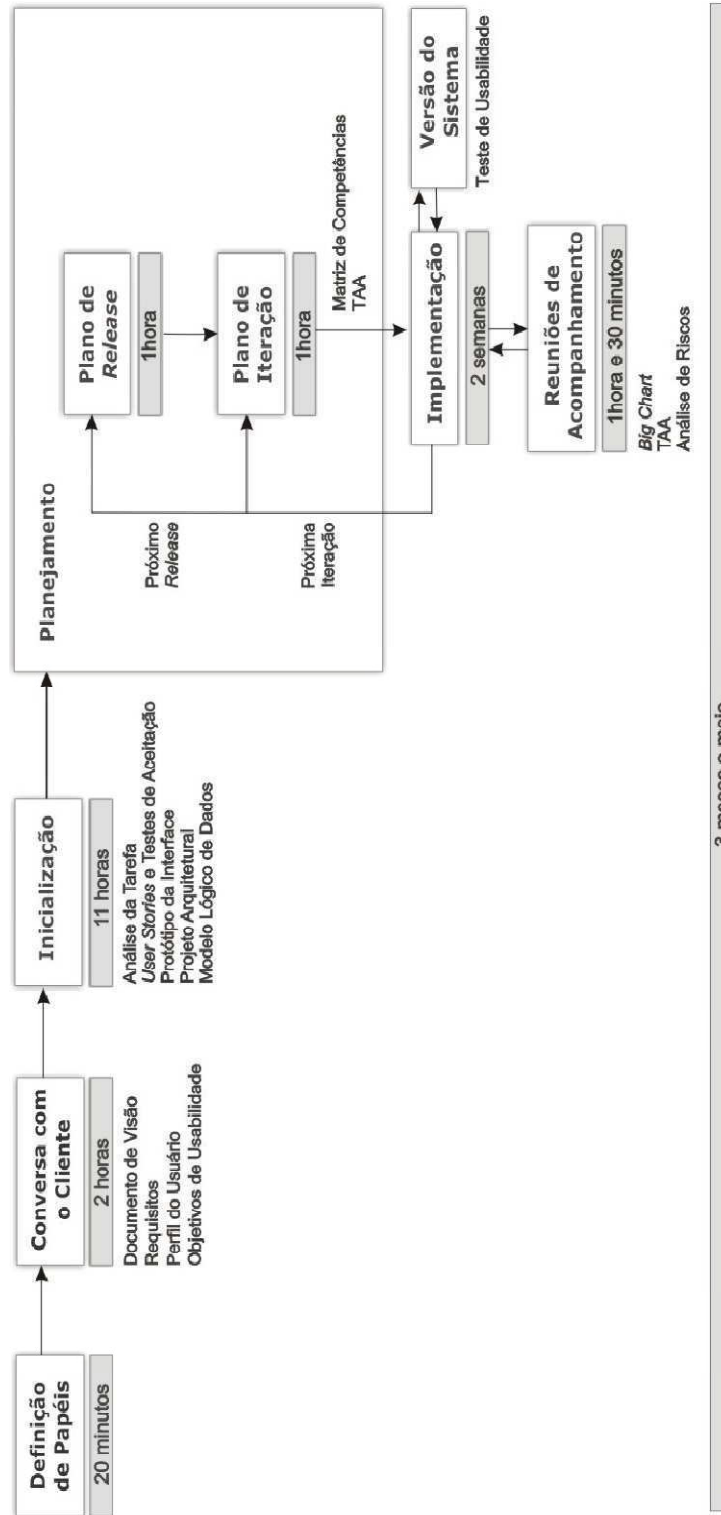


Figura 3 – Fluxo do processo de desenvolvimento easYProcess
FONTE: easYProcess (2009)

2.2. LOCALIZANDO O PROBLEMA DO MANEJO DA USABILIDADE EM YP

Apesar de YP sugerir a geração e manejo dos artefatos de usabilidade supracitados, não existe nenhum contexto metodológico específico que oriente e auxilie na elaboração desses artefatos (Aguiar, 2007).

O acompanhamento da aplicação de YP pelos alunos de LES, na UFCG, permitiu perceber que as equipes de desenvolvimento se queixavam do esforço despendido na construção dos artefatos de usabilidade, especialmente aqueles referentes à análise da tarefa e à obtenção do protótipo de interface. Ao mesmo tempo, as equipes, avaliavam como pequeno o impacto que tais artefatos traziam para o produto final.

Considerando que diversos clientes acompanharam o desenvolvimento de software a partir da aplicação do YP, com e sem os artefatos de usabilidade e que os relatos feitos por tais clientes expressaram relativamente um índice de satisfação superior com os produtos oriundos de YP com usabilidade, pode-se concluir que tais artefatos agregam valor ao produto final e que a queixa dos alunos se deve à dificuldade em produzir, manejar e validar os recursos de usabilidade sugeridos pelo processo. A exposição destes problemas já foi apresentada de forma sintética na seção 1.6 (Quadro 1).

O problema de utilização do processo e de confiança no tratamento da usabilidade ocorre já na fase de Inicialização de YP, pela necessidade de gerar o modelo da tarefa e o protótipo de interface. Até então este manejo era realizado sem um suporte metodológico e ferramental específico.

Aguiar (2007), em seu trabalho sobre prototipagem de média-fidelidade sugere que o processo de desenvolvimento YP poderia ser assessorado na questão do projeto de interface e de usabilidade pelos conceitos fornecidos pela metodologia de concepção de interfaces MEDITE. Analisando de forma mais abstrata e independente este

posicionamento, uma metodologia para a concepção de interfaces adaptável ao contexto de YP deve contemplar as seguintes critérios: (i) **uso de modelos conceituais** para a gestão do conhecimento produzido sobre o usuário e a interface compatíveis com os artefatos definidos por YP, especialmente o modelo da tarefa e o protótipo da interface; (ii) **integração entre modelos**, a partir da oferta de uma sistemática clara de sua geração e uso; (iii) **suporte ferramental**, para tornar a geração e manejo desses modelos mais ágil, automatizando algumas etapas que possam requerer um mapeamento entre modelos, a fim de propiciar uma visualização clara de sua integração; (iv) **múltipla prototipagem**, para permitir a avaliação antecipada de problemas no projeto de interface. Na Figura 4, resume-se a adequação de MEDITE a YP.

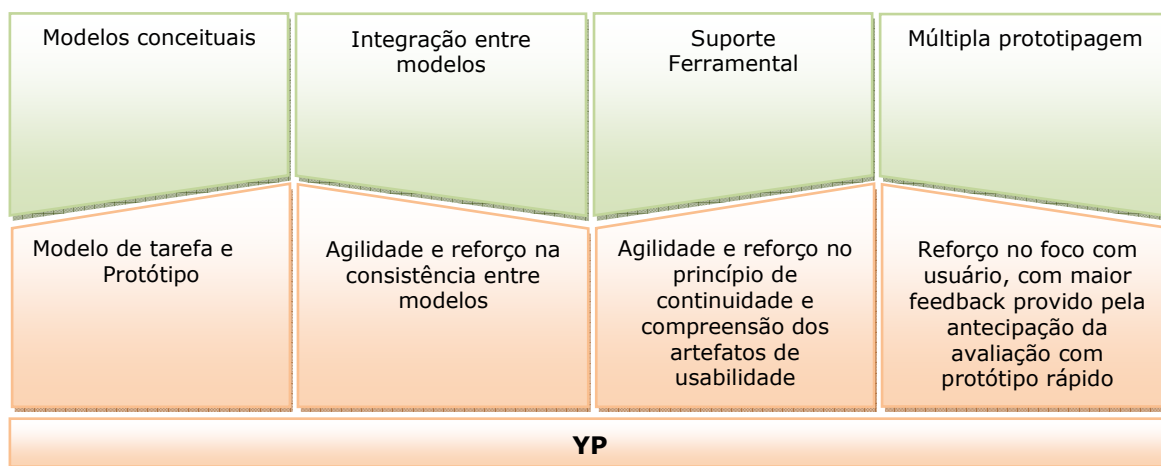


Figura 4 – Adequação de MEDITE em YP.

A seleção da metodologia MEDITE (Guerrero, 2002; Oliveira, Aguiar, & Lula, 2007; Suárez & Lula, 2004) para assessorar a usabilidade poderia ser replicável utilizando-se outras adaptações de processos de software ágeis e centrados no usuário, a exemplo da adaptação do PDS XP para tratamento da usabilidade proposta por (Bankston, 2002); o *Agile Usage-Centered Design* (Constantine, 1999); ou XPU (Vasconcelos,

Garcia, & Turnell, 2003) que é assessorada por outra metodologia de interfaces, similar a MEDITE, a MCI. Esta predisposição é possível desde que o PDS considerado incorpore os três princípios básicos de Gould&Lewis para projetos de software centrados no usuário (*apud* RUBIN, 1994): (i) ênfase no usuário e na tarefa deve ocorrer desde o início; (ii) definição de medidas empíricas da usabilidade do produto em desenvolvimento; (iii) design iterativo do produto, ou seja, modificação e teste do mesmo repetidas vezes.

A seleção de YP, especificamente, levou em consideração a existência de um ambiente já amadurecido de uso, no caso o da disciplina LES da UFCG. A seleção da metodologia MEDITE se deu, principalmente, pela simplicidade de seus modelos, com a presença de suporte ferramental específico e integrado para a concepção de interfaces. Na seção a seguir, descrevem-se os detalhes de MEDITE.

2.3. MEDITE: UMA ALTERNATIVA DE SUPORTE METODOLÓGICO PARA CONCEPÇÃO DE INTERFACES EM YP

MEDITE é uma metodologia para a concepção de interface com o usuário orientada a modelos, baseada no modelo da tarefa e em múltipla prototipagem (Figura 5). MEDITE propõe o desenvolvimento da interface de forma iterativa e incremental, sendo compatível com o framework *Cameleon* (Calvary, Coutaz, & Thevenin, 2003), considerado de referência na literatura e concebido para descrever o ciclo de vida de desenvolvimento de interfaces com o usuário.

O uso de MEDITE é suportado atualmente por um ambiente de desenvolvimento de interface com o usuário, o *FastInterface* (Oliveira, Aguiar, & Lula, 2007), que consiste em uma suíte de ferramentas integradas para agilizar a construção do modelo da tarefa e de protótipos em vários níveis de abstração. De acordo com estas

características, observa-se a compatibilidade de MEDITE com os critérios de elegibilidade de uma metodologia de concepção de interfaces para YP, supracitados.

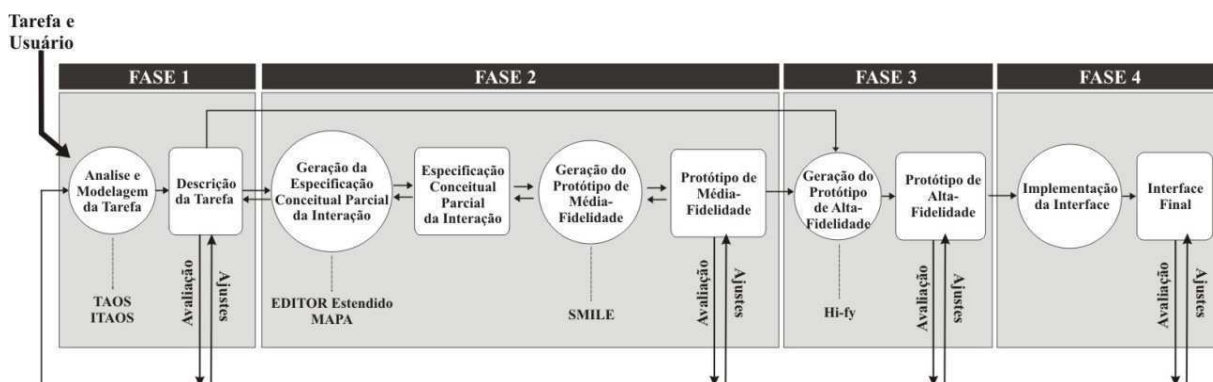


Figura 5 – Fluxo de MEDITE

Fazendo-se uma relação entre os artefatos definidos por YP e os processos fornecidos por MEDITE para a elaboração dos artefatos de usabilidade, tem-se que:

Na **fase 1** de MEDITE (*Task&Concept*) são apresentados os processos relacionados a análise e modelagem da tarefa do usuário. Para que estes processos sejam executados com sucesso, se faz necessário realizar uma atividade de levantamento do perfil do usuário (implícita). Desta forma, pode-se dizer que a execução da Fase 1 de MEDITE resulta no artefato Modelo da Tarefa que é, por sua vez, definido em YP.

Na **fase 2** de MEDITE (*AUI – Abstract User Interface*) são introduzidos os processos relativos a construção de um protótipo da interface do usuário. Esta fase acrescenta informações ao Modelo da Tarefa (obtido na fase anterior) para que seja possível construir um protótipo de média-fidelidade da interface do usuário, ou seja, um protótipo de aparência ainda rudimentar, mas que permite apresentar a disposição

dos componentes de interação da interface, sendo possível simular a interação com os mesmos. Sendo assim, a fase 2 de MEDITE resulta em protótipo da interface do usuário, artefato também definido por YP.

A **fase 3** (*CUI – Concrete User Interface*) de MEDITE é responsável por refinar o protótipo obtido na fase anterior, a fim de tornar a sua aparência e funcionamento (ainda simulado) os mais próximos da interface final do sistema em desenvolvimento. Ao final da execução desta fase, se obtém um segundo protótipo da interface, o protótipo de alta-fidelidade.

A **fase 4** (*FUI – Final User Interface*), consiste na implementação real da interface do usuário, a partir do protótipo de alta-fidelidade. Nesta fase se concebe a interface final já implementada, ou seja, com o código subjacente a uma plataforma ou ambiente.

Embora YP não sugira explicitamente a necessidade de construção de múltiplos protótipos, de alta e de média fidelidade, é prática, em um processo iterativo e incremental de desenvolvimento da interface, evoluir de protótipos mais simples para protótipos mais completos à medida que o entendimento do contexto de uso do software, ou do domínio do problema, se consolida (Oliveira, Aguiar, & Lula, 2007; Coyette, Kieffer, & Vanderdonckt, 2007).

Em YP, os testes de usabilidade são conduzidos após a geração da versão final do sistema em desenvolvimento. Com a inserção de conceitos de MEDITE no processo, a realização dos testes de usabilidade pode acontecer ao longo da construção do software, pois, uma vez que a equipe de desenvolvimento possui os protótipos da interface do usuário, pode executar testes de aprovação em cima desses. Desta forma, não se faz necessário possuir uma versão funcional do sistema para que o mesmo seja testado do ponto de vista da usabilidade. Vários testes de usabilidade podem ser executados ao longo do

desenvolvimento do software, de acordo com a necessidade, antecipando insatisfações dos usuários potenciais do sistema (e clientes), permitindo a realização de ajustes ao longo da especificação da interface, ao invés de terem que ser detectados somente ao término do desenvolvimento.

Por se tratar de uma metodologia baseada em modelos, MEDITE utiliza o modelo da tarefa e o modelo da interação⁴ como base para a construção dos artefatos propostos pela metodologia. Os múltiplos protótipos suportados por MEDITE (média e alta fidelidade) são derivados desses modelos, assim como o código final da interface com o usuário. Os artefatos preconizados por MEDITE estão associados a formalismos e modelos que propiciam o mapeamento de seus elementos a fim de favorecer a construção de um artefato com base no artefato que o antecedeu. Nas subseções seguintes, apresentam-se cada um dos artefatos, propostos por MEDITE.

2.3.1. MODELO DA TAREFA

O modelo da tarefa consiste em uma representação das tarefas a serem realizadas pelo usuário com o auxílio do sistema em desenvolvimento (Lu, Paris, & Linden, 1998; Johnson *et al.*, 1988). A modelagem da tarefa acontece em sincronia com o processo de análise da tarefa, em que ambos objetivam melhorar o entendimento dos desenvolvedores acerca das tarefas a serem realizadas e das características e necessidades dos usuários ao executá-las (aspectos relacionados às características da tarefa, ao perfil do usuário e aos objetivos de usabilidade devem ser levantados em uma etapa anterior).

⁴ Especificação conceitual parcial e/ou total da interação do usuário com os elementos da interface.

A maturidade com relação à tarefa e ao usuário é atingida quando os desenvolvedores compreendem a natureza da tarefa, as partes que a compõe e a ordem na qual devem ser realizadas. A partir do modelo da tarefa, apresenta-se a decomposição sucessiva de tarefas em sub-tarefas, até atingir as ações elementares a serem ativadas diretamente pelos usuários. O processo de análise e modelagem da tarefa do usuário em MEDITE utiliza o formalismo TAOS (*Task and Action Oriented System*) (Medeiros, Kafure, & Lula, 2000) que possui como conceitos fundamentais tarefa, sub-tarefas e ações.

2.3.2. MODELO DA INTERAÇÃO

No modelo da interação, define-se como os usuários podem executar as tarefas a partir dos dispositivos de interação definidos pelo sistema. Em MEDITE, o modelo da interação é definido pelo modelo EDITOR Estendido (Rodrigues, Lula, & Suárez, 2005) que compreende dois componentes: a apresentação e o diálogo. O componente de apresentação define a representação da interface através dos elementos espaço, visão e objeto de interação. O componente de diálogo descreve o fluxo da interação (seqüências de trocas de mensagens) entre o usuário e os componentes de apresentação e é comumente descrito na forma de um diagrama de transição de estados. Assim, o componente de diálogo pode ser descrito de maneira análoga a uma máquina de estados finitos, onde os estados representam os elementos de apresentação (espaços, visões e objetos de interação) e os arcos representam as ações dos usuários sobre estes elementos (ativação de funções do sistema).

Metodologias para a concepção de interface com o usuário baseadas na tarefa sugerem que o modelo da interação seja gerado a partir das informações presentes no modelo da tarefa (Johnson, Johnson, &

Wilson, 1995). Desta forma, com o uso de MEDITE, é possível partir da descrição da tarefa (modelo da tarefa) e das características e necessidades dos usuários (perfil do usuário) para obter o modelo da interação.

2.3.3. PROTÓTIPO DE MÉDIA-FIDELIDADE

Protótipos de média-fidelidade são obtidos a partir da prototipagem de baixa-fidelidade (esboços, *storyboards*) com suporte computacional, de modo a possibilitar a simulação de seu comportamento tal como um protótipo de alta-fidelidade. Essa simulação favorece a prática da prototipagem exploratória e evolutiva (Aguiar, 2007; MOFFAT *et al.*, 2003; Engelberg & Seffah, 2002; Leone & Gillihan, 2000).

Em MEDITE, os elementos presentes no modelo da interação (espaços, visões e objetos de interação) e aspectos da comunicação estabelecida entre estes elementos, tais como sua ativação e relacionamento espaço-temporal, são suficientes para se produzir uma primeira representação visual da interface na forma de um protótipo de média-fidelidade, sem a necessidade de geração de código executável de interface.

2.3.4. PROTÓTIPO DE ALTA-FIDELIDADE

Protótipos de alta-fidelidade aproximam-se bastante do produto idealizado. São representações executáveis (código) construídas com o uso de uma linguagem de programação (ou ferramentas de apoio) e contêm as principais funcionalidades presentes na interface do futuro sistema (Buskirk & Van Moroney, 2003; Rudd, Stern, & Insensee, 1996; Rettig, 1994). Estes protótipos permitem que os usuários possam simular o comportamento da futura interface, executando operações (entrada de dados, preenchimento de formulários e seleção de ícones)

de maneira muito semelhante às que serão realizadas com a interface final. Desta forma, o usuário tem o sentimento de como o produto irá operar ainda na fase de projeto, podendo fornecer, prematuramente, *feedbacks* mais significativos dos usuários e clientes, principalmente relativos à estética da interface.

Em MEDITE, o protótipo de alta-fidelidade consiste em um refinamento do protótipo de média-fidelidade. Os elementos presentes no protótipo de média-fidelidade (espaços, visões e objetos de interação), ainda com representação abstrata, são mapeados para elementos concretos (janelas, botões, menus, *checklist*, etc.). O protótipo de alta-fidelidade é o ponto de partida para a geração de código da interface final do sistema em desenvolvimento para execução em uma plataforma particular.

2.4. FASTINTERFACE: SUPORTE FERRAMENTAL À MEDITE

Nas seções anteriores, pôde-se perceber que é inevitável o investimento em tempo e esforço para tratar de maneira consistente os modelos sugeridos por uma metodologia de concepção de interfaces, inclusive MEDITE. Deve-se, ainda, destacar a complexidade de mapeamento entre os modelos de tarefa e a obtenção do protótipo de média-fidelidade: desenvolver o modelo de interação, manualmente é inviável, sobretudo considerando a manutenção da agilidade e do estímulo a práticas de IHC.

Para os procedimentos para concepção de interfaces de MEDITE se adéquem ao YP de maneira integrada e mantendo sua agilidade é necessário disponibilizar ferramentas capazes de auxiliar na construção das etapas de modelagem de tarefa e de prototipagem, se possível, assistidos por computador. Assim, mapeamentos podem ser realizados de forma automática entre os elementos dos modelos envolvidos: o uso

de ferramentas computacionais de apoio torna-se, portanto, um ponto fundamental para tornar a construção dos artefatos de usabilidade de YP mais ágil e, na medida do possível, estimulante.

O suporte ferramental de MEDITE atualmente é possível através da ferramenta integrada *FastInterface* (Oliveira, Aguiar, & Lula, 2007), que foi desenvolvida especialmente para permitir um processo de obtenção da interface em 4 níveis de abstração, de acordo com o framework *Cameleon*. Sua aplicação no manejo da usabilidade em YP, implicaria em: (i) melhoria da agilidade à fase de Inicialização de YP, (ii) modelagem assistida por computador da interface; e (iii) obtenção de artefatos consistentes.

Os níveis de abstração mencionados foram concretizados a partir da integração de 4 módulos de ferramentas para apoio às fases de MEDITE:

- i. **ITAOS** (Medeiros, Lula, & Cordeiro, 2002) para a construção e análise do modelo da tarefa (Tarefa e Conceitos/Fase 1 de MEDITE);
- ii. **MAPA** (Rodrigues, Lula, & Suárez, 2005) algoritmos de transformação e manutenção da consistência (e rastreabilidade) entre o modelo da tarefa e modelo de interface abstrata do usuário, baseado em regras ergonômicas (AUI/Fase 2 de MEDITE);
- iii. **SMILE** (Aguiar, 2007) para a edição e simulação do modelo de interface abstrata do usuário em forma de protótipos de média fidelidade (AUI/Fase 2 de MEDITE);
- iv. **Hi-fy** (Oliveira, Aguiar, & Lula, 2007) para a geração e edição do modelo de interface concreta do usuário sob a forma de um protótipo de alta fidelidade (CUI/Fase 3 de MEDITE) e para geração de interface final do usuário (FUI/Fase 4 de MEDITE).

O uso dos módulos que compõe *FastInterface* são detalhados no Anexo D. Na figura 6, apresenta-se a correspondência entre os níveis de abstração do processo de desenvolvimento proposto em *Cameleon*, os módulos que compõem *FastInterface* e as fases correspondentes de MEDITE (Figura 6).

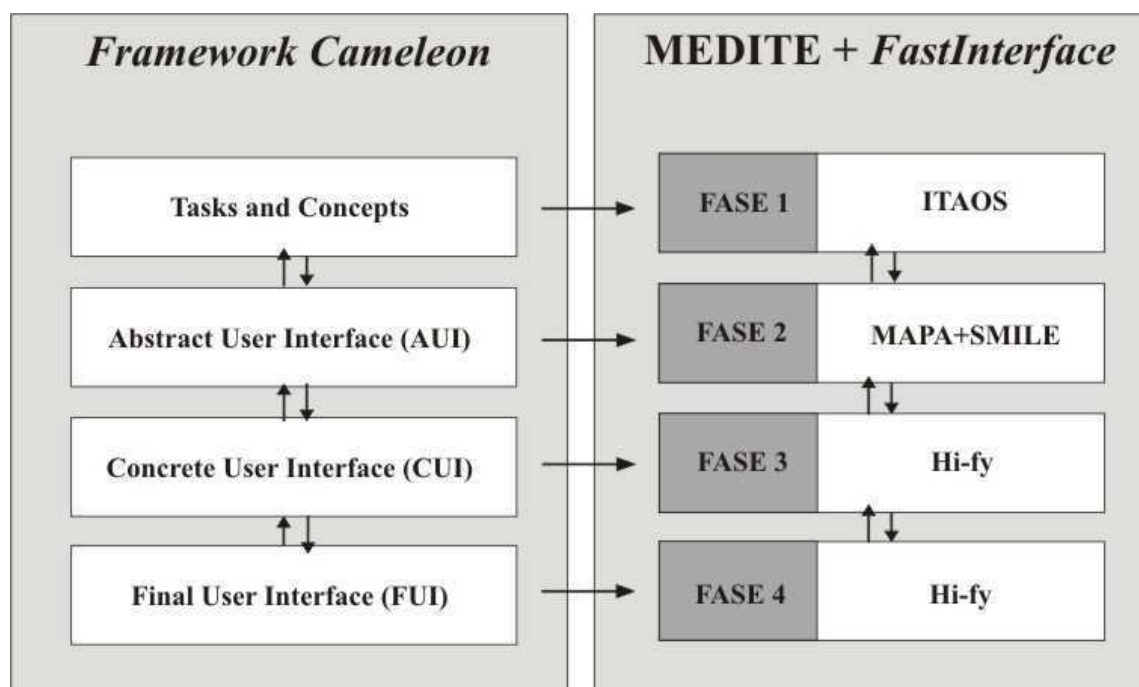


Figura 6 – Cobertura dos módulos de *FastInterface* em MEDITE, sob o prisma de *Cameleon*.

2.5. SÍNTESE DE FATOS DO CONTEXTO ATUAL DE YP COM USABILIDADE E PERSPECTIVAS DE MELHORIA

Considerando os problemas já explicitados com o manejo da usabilidade em YP e a proposta de tratamento com a aplicação da metodologia MEDITE assistida pela ferramenta *FastInterface*, apresenta-se o quadro a seguir (Quadro 2), no sentido de sintetizar as relações entre os fatos negativos no atual contexto de aplicação de YP e as expectativas de melhoria.

Quadro 2 – Comparativo de expectativas entre as abordagens com e sem MEDITE+FastInterface

	Problemas atuais Sem MEDITE+FastInterface	Expectativas de Melhoria Com MEDITE+FastInterface
Análise da tarefa	<ul style="list-style-type: none"> • Seleção do modelo de tarefa é indicada por vários formalismos e ferramentas. Sua obtenção fica a critério da equipe de desenvolvimento • Geralmente os alunos não têm subsídios ou antecedentes para decidir com segurança qual seria a melhor opção • Há uma tendência em se modelar a tarefa sem critérios de avaliação claros, o que pode levar a incredibilidade, ou até o descarte, desse artefato 	<ul style="list-style-type: none"> • Modelo de tarefas utiliza o formalismo TAOS suportado pelo módulo iTAOS na ferramenta FastInterface • O Formalismo TAOS possui diversos níveis de especificação para tarefas. No entanto, se pode incentivar apenas a disposição hierárquica e a lógica de execução das mesmas, a partir do emprego de operadores. • Modelagem da tarefa com critérios mínimos de validação. A obtenção do protótipo de interface é uma reflexão clara do modelo de tarefas, a partir da aplicação de regras ergonômicas de transformação, até, porque, o protótipo será gerado somente se o modelo de tarefas estiver em um nível mínimo de especificação.
Prototipagem	<ul style="list-style-type: none"> • Não há indicação explícita de avaliação de protótipos junto aos clientes/usuários. • Há uma tendência de obtenção do protótipo apenas para auxiliar no desenvolvimento da interface intra-equipe. • Não é intuitivo o uso do modelo de tarefa como guia na obtenção do protótipo da interface, pois, geralmente esses são gerados com editores gráficos 2D, a mão ou em HTML em separado 	<ul style="list-style-type: none"> • Protótipos de média e de alta fidelidade devem ser avaliados pelo cliente. • Ainda que o de alta não seja utilizado o de média auxilia no fechamento dos elementos de interface, sua disposição e estilo de interação junto aos clientes/usuários • Protótipo é obtido automaticamente em caráter preliminar, podendo ser aperfeiçoado, exigindo menos esforço em sua obtenção • O aluno deve melhorar o protótipo para atender as expectativas dos clientes e usuários, devendo ser aprovado pelos mesmos. • Tendência a rápidas e poucas intervenções devido ao auxílio computacional.
Gestão de artefatos de usabilidade	<ul style="list-style-type: none"> • Gerência dos documentos relativos à especificação do modelo de tarefas e do protótipo ficam em separado em múltiplos arquivos e mídias, dificultando o manejo. 	<ul style="list-style-type: none"> • Gerência integrada do modelo de tarefas e do protótipo, com geração do protótipo dependente do modelo de tarefas.

Vale salientar que para que esta adequação de MEDITE a YP seja realizável, algumas adaptações tiveram que ser sugeridas, principalmente quanto: (i) à minimização da especificação da análise da tarefa, que pode ser explorada com ênfase na obtenção da disposição hierárquica de abstração, da lógica de execução das tarefas e de sua ocorrência; (ii) ao uso opcional dos recursos de geração do protótipo de

alta-fidelidade por *FastInterface*, mas, reforçando o emprego do protótipo de média-fidelidade já que este é gerado automaticamente e sem muito esforço de adaptação.

Essa adaptação foi realizada pelo fato de que o suporte à edição do protótipo de alta-fidelidade por *FastInterface* contempla apenas a plataforma HTML/Web e, ainda, pode ser deficiente em alguns casos por não atender às expectativas do usuário quanto a estilo, além da impossibilidade de aplicação de restrições de estilo consistentes com outras plataforma de implementação para interfaces.

CAPÍTULO 3

AVALIAÇÃO DE YP COM MEDITE + *FASTINTERFACE*: PLANEJAMENTO, CONFIGURAÇÃO E RESTRIÇÕES

Neste capítulo, descreve-se como ocorreu processo de avaliação do impacto em YP via MEDITE+*FastInterface*. As técnicas utilizadas e detalhes sobre configuração, hipóteses, restrições de validade, variáveis, medições e relações serão apresentados aqui.

3.1. AMBIENTE DE AVALIAÇÃO

A disciplina Laboratório de Engenharia de Software (LES), do curso de Bacharelado em Ciência da Computação da UFCG, tem o propósito de abordar procedimentos e técnicas consolidadas de *design* de software, em que o aprendizado é reforçado pela vivência real de concepção, desenvolvimento, produção e validação do software. Em geral, esses produtos de software possuem a característica de serem utilizados no atendimento à demandas autênticas de software, com clientes e usuários reais que interagem diretamente com os alunos.

O processo YP, é capaz de orientar todo o laboratório de software. As demandas de software são distribuídas entre grupos de alunos que devem desenvolver cooperativamente todas as etapas recomendadas pelo processo, de maneira rotativa, em que todos têm a oportunidade de assumir papéis de gerente ou de desenvolvedor, ao longo do semestre letivo. As equipes de desenvolvimento, em geral, são compostas por 3 ou 4 alunos.

A fim de extrair evidências comparativas entre a abordagem de uso de YP com e sem suporte metodológico/ferramental de

MEDITE+*FastInterface*, um experimento foi elaborado a partir da composição de 2 cenários distintos. Em um deles, as equipes de desenvolvimento conduziam o projeto de software da forma habitual (grupo de controle). No outro, as equipes adotavam a nova abordagem (grupo de tratamento). No Quadro 3, sintetizam-se os níveis da informação prestada em relação aos procedimentos de usabilidade, em ambos os cenários, na condução do experimento.

Quadro 3 – Níveis da informação com e sem MEDITE+*FastInterface*

Prestação da informação	Cenário habitual Sem MEDITE+<i>FastInterface</i>	Cenário de tratamento MEDITE+<i>FastInterface</i>
Obtenção do perfil do usuário e dos objetivos de usabilidade	<ul style="list-style-type: none"> • Construção após a conversa com o cliente 	<ul style="list-style-type: none"> • Construção após a conversa com o cliente
Análise da tarefa	<ul style="list-style-type: none"> • Realizar a análise da tarefa após o perfil do usuário e dos objetivos de usabilidade, dentro da etapa de planejamento (que contempla outros artefatos funcionais) • Oferta de múltiplos métodos de análise, formalismos e ferramentas. Apenas TAOS é abordado com exemplos. • Seleção do método e da ferramenta é uma responsabilidade da equipe 	<ul style="list-style-type: none"> • Realizar a análise da tarefa logo após a conversa com o cliente. • Ênfase para realizá-la estritamente nesse momento e, principalmente, antes de qualquer artefato funcional • Prestação de informações sobre hierarquia de tarefas, nível de realização e uso de operadores para uso do formalismo TAOS • Reforço de ser compreendido por toda a equipe • Reforço de ser demonstrado ao cliente para ser válido • Reforço em seu emprego para elaboração de manuais de ajuda e roteiro de testes de usabilidade
Prototipagem	<ul style="list-style-type: none"> • Logo após o modelo de tarefas • Tendência a ser realizado concorrentemente com o modelo de tarefas e outros artefatos funcionais da etapa de planejamento de YP • Oferta de múltiplas ferramentas para prototipagem. • Tendência para as ferramentas de prototipagem com formalismos e regras de IHC de não serem escolhidas • Reforço de ser demonstrado ao cliente 	<ul style="list-style-type: none"> • Obtenção do protótipo de média-fidelidade estritamente após o modelo de tarefas: restrição imposta por <i>FastInterface</i> • Reforço de ser validado por toda a equipe • Reforço de ser demonstrado ao cliente para ser válido • Obtenção do protótipo de alta-fidelidade é optativa. A equipe fica livre para escolher a ferramenta de autoria de gráficos para obtenção do protótipo de alta-fidelidade ou já pode implementar sob uma plataforma de interface.
Preparação do material para teste de usabilidade	<ul style="list-style-type: none"> • Informações superficiais relativas à elaboração do material de teste 	<ul style="list-style-type: none"> • Emprego de guia rápido para uso do modelo de tarefas para eleição das tarefas mais representativas a serem avaliadas e para produção do roteiro de teste

A compreensão do cenário habitual, junto ao grupo de controle, também se baseou em informações coletadas com ex-monitores da disciplina e principalmente na especificação do processo YP disponibilizada eletronicamente (easYProcess, 2006), que é usada como referência pelos alunos.

Foi possível constatar que em geral os alunos não possuem noções sobre métodos e técnicas de projeto de interface. Em 2008.2 o experimento pôde ser realizado com metade das equipes inseridas nos cenários de controle e a outra metade no de tratamento, em que pares de equipes formadas por membros distintos foram observadas realizando projetos com clientes e domínios de problemas iguais, sendo uma utilizando a abordagem habitual e a outra a experimental.

Esta configuração foi considerada a melhor, já que os sujeitos estariam sendo observados de forma comparativa a partir do mesmo assunto. Outra alternativa seria aplicar a uma mesma equipe a abordagem habitual e a experimental, variando entre uma equipe e outra se esta ou aquela abordagem seria realizada primeiro, afim de eliminar o viés de uma equipe ter desenvolvido uma noção do domínio de problema do projeto já na primeira abordagem.

Entretanto, dificuldades foram encontradas para se reproduzir as duas configurações pareadas, descritas acima, posteriormente: (i) insuficiência da quantidade de alunos necessários, considerando aqueles matriculados na disciplina; (ii) impossibilidade dos clientes para absorver 2 equipes simultaneamente para um mesmo projeto.

Desta forma, foi decidido que as equipes de LES que desenvolveram projetos de software nos períodos de 2009.1 e 2009.2, continuariam a usar a abordagem habitual, sem o suporte ferramental *FastInterface*. Como o cenário habitual já era conhecido estas equipes não foram observadas de maneira direta e seus dados foram coletados a partir de

registros e artefatos produzidos pelas equipes em seus repositórios de projeto. As opiniões e aspectos subjetivos ao manejo da usabilidade dessas equipes foram registradas por meio de questionários eletrônicos do *GoogleDocs*.

Como a turma de 2010.1 foi relativamente maior, todas as equipes utilizaram a abordagem experimental, para obter uma quantidade de amostras equivalentes para ambos os cenários, habitual e experimental.

3.2. EXPERIMENTO CONTROLADO

A especificação e a condução do experimento controlado seguem as orientações dos seguintes trabalhos de pesquisa experimental em Engenharia de Software (Kitchenham *et al.*, 2002; Travassos *et al.*, 2002).

O estudo se caracteriza como experimental por sugerir uma nova abordagem para manejo da usabilidade em YP no laboratório de software e por se focalizar nas questões dos efeitos dessa nova abordagem. Tais efeitos ou o impacto destes podem ser verificados na satisfação, compreensão e aceitação dos alunos e clientes de projeto, bem como no desempenho relativo ao desenvolvimento do produto, com o manejo de artefatos de usabilidade proposto.

3.2.1. OBJETIVOS

Por já ser tradicional o uso da abordagem *Goal/Question/Metric* (GQM) (Solingen & Berghout, 1999) para descrição de pesquisas em ES, a mesma foi usada para definir os objetivos deste experimento. Seguindo a abordagem GQM para descrição sintética de pesquisas, tem-se para esta contextualização as seguintes premissas:

Analisar a nova abordagem para tratamento da usabilidade MEDITE+*FastInterface*.

Com o propósito de caracterizar o impacto da nova abordagem sobre o esforço no manejo de artefatos de usabilidade e sobre a compreensão funcional do software.

Com respeito à satisfação, compreensão e esforço dos alunos e respectivos clientes/usuários; nível de especificação do modelo de tarefas e protótipo de interface; nível de especificação de casos de uso e testes associados; nível de casos de uso resolvidos.

Do ponto de vista do pesquisador.

No contexto do laboratório de engenharia de software orientado por processo de desenvolvimento ágil e centrado no usuário com alunos inexperientes em técnicas de IHC.

3.2.2. ARTEFATOS FONTES DE DADOS

A seguir, serão descritos os artefatos que foram utilizados na investigação do impacto da presença do apoio metodológico e ferramental de MEDITE+*FastInterface* em YP, no contexto da disciplina LES. Para facilitar a referência e a explicação dos artefatos, para obtenção de dados na pesquisa, os mesmos foram agrupados em 5 conjuntos: (i) artefatos e procedimentos de usabilidade; (ii) artefatos e procedimentos relativos ao funcional do software; (iii) alunos; (iv) clientes/usuários; (v) produto final (Figura 7). Os elementos dos itens (i) e (ii) são automaticamente realizados pelas equipes de alunos, em YP.



Figura 7 – Conjuntos de artefatos para coleta de dados

3.2.3. MÉTRICAS

A análise dos artefatos fontes de dados, apresentados na seção anterior, será realizada com respeito às seguintes variáveis aplicáveis a cada projeto nas equipes de desenvolvimento.

- **Quantidade de tarefas (QT):** diz respeito ao nível de detalhamento das tarefas introduzidas no modelo de tarefas por uma equipe.

$$QT = \sum \text{tarefas}$$

- **Taxa de realização das tarefas (TRT):** diz respeito ao nível de detalhamento de tarefas, obtida a partir da relação de ações (QA) sobre quantidade total de tarefas (QT).

$$TRT = \frac{QA}{QT}$$

- **Uso de Operadores (UO):** diz respeito à quantidade de emprego de cada um dos operadores (SEQ, AND, OR, XOR, SIM, PAR) (O_p) nos modelos de tarefa.

$$UO = \sum O_p$$

- **Tamanho do protótipo (TP):** corresponde a quantidade de representações (de telas) no artefato de prototipagem.

$$TP = \sum \text{telas}$$

- **Taxa de elementos de interação no protótipo (TEIP):** refere-se à distribuição de objetos de interação (OI) em relação ao tamanho do protótipo (TP), ou seja, a taxa de objetos de interação por tela.

$$TEIP = \frac{\sum OI}{TP}$$

- **Nível de especificação do projeto funcional (NEPF):** trata da relação entre a quantidade de testes de aceitação (USe) e a quantidade de casos de uso (USe) especificados.

$$NEPF = \frac{\sum TA}{\sum USE}$$

- **Nível de realização do projeto (NRP):** trata da proporção entre casos de uso especificados (USE) e casos de uso realizados (USR).

$$NEPF = \frac{\sum USE}{\sum USR}$$

As métricas a seguir são subjetivas e foram obtidas a partir de questionários específicos aplicados com as equipes e seus respectivos clientes e usuários. Tais questionários foram elaborados com a intenção de mensurar: (i) esforço e satisfação no aprendizado e construção dos artefatos de usabilidade; (ii) satisfação dos clientes/usuários com os artefatos de usabilidade; (iii) aprendizado colaborativo fruto da melhor interação intra-equipe e entre clientes/usuários, e; (iv) correlação entre artefatos gerados (sejam estes funcionais ou de usabilidade).

Os questionários foram adaptados a partir do modelo para montagem de questionários USE (Queiroz, 2001) empregado na estimação da satisfação subjetiva em avaliações de interface com o usuário. Esse modelo utiliza uma escala de 5 pontos para mensurar o grau de reatividade a cada questão, pelo respondente. As questões foram elaboradas dentro das intenções do processo de sondagem. Os exemplares de cada um dos questionários se encontram nos anexos B e C, para alunos e clientes/usuários, respectivamente.

3.2.4. DEFINIÇÃO DAS HIPÓTESES

Hipóteses foram estabelecidas, a fim de caracterizar o impacto de com (CMF) e sem (SMF) MEDITE+*FastInterface* em YP. A hipótese nula H_0

(ou de refutação) é definida como a inexistência de efeitos significativos ao adotar o suporte metodológico e ferramental proposto. O desdobramento das hipóteses se dá em 11 sub-hipóteses nulas e alternativas, descritas a seguir.

Hipóteses nulas ($H0_{1...11}$): ($H0_1$) quantidade de tarefas CMF igual a SMF; ($H0_2$) nível de especificação da realização das tarefas CMF igual a SMF; ($H0_3$) taxa de uso de operadores CMF igual a SMF; ($H0_4$) tamanho do protótipo CMF igual a SMF; ($H0_5$) número de elementos de interação no protótipo CMF igual a SMF; ($H0_6$) nível de especificação do projeto funcional CMF igual a SMF; ($H0_7$) nível de realização do projeto CMF igual a SMF; ($H0_8$) grau de facilidade de manejo e aprendizado do aluno com artefatos de usabilidade CMF igual a SMF; ($H0_9$) grau de satisfação do cliente com os artefatos de usabilidade CMF igual a SMF; ($H0_{10}$) aprendizado colaborativo CMF igual a SMF; ($H0_{11}$) nível de integração entre a análise da tarefa e a prototipagem CMF igual a SMF.

Quadro 4 – Hipóteses nulas

$H0_1: QT_{SMF} = QT_{CMF}$	$H0_7: NRP_{SMF} = NRP_{CMF}$
$H0_2: TRT_{SMF} = TRT_{CMF}$	$H0_8: FMAU_{SMF} = FMAU_{CMF}$
$H0_3: UO_{SMF} = UO_{CMF}$	$H0_9: SCAU_{SMF} = SCAU_{CMF}$
$H0_4: TP_{SMF} \cong TP_{CMF}$	$H0_{10}: AC_{SMF} = AC_{CMF}$
$H0_5: TEIP_{SMF} = TEIP_{CMF}$	$H0_{11}: CAG_{SMF} = CAG_{CMF}$
$H0_6: NEPF_{SMF} = NEPF_{CMF}$	

Hipóteses alternativas ($H1_{1...11}$): ($H1_1$) quantidade de tarefas maior CMF; ($H1_2$) nível de especificação da realização das tarefas maior CMF; ($H1_3$) taxa de uso de operadores maior CMF; ($H1_4$) tamanho do protótipo maior CMF; ($H1_5$) número de elementos de interação no protótipo maior CMF; ($H1_6$) nível de especificação do projeto funcional

maior CMF; ($H1_7$) nível de realização do projeto maior CMF; ($H1_8$) grau de facilidade de manejo e aprendizado do aluno com artefatos de usabilidade igual maior CMF; ($H1_9$) grau de satisfação do cliente com os artefatos de usabilidade maior CMF; ($H1_{10}$) aprendizado colaborativo maior CMF; ($H1_{11}$) nível de integração entre a análise da tarefa e a prototipagem maior CMF.

Quadro 5 – Hipóteses Alternativas

$H1_1: QT_{CMF} > QT_{SMF}$	$H1_7: NRP_{CMF} > NRS_{SMF}$
$H1_2: TRT_{CMF} > TRT_{SMF}$	$H1_8: FMAU_{CMF} > FMAU_{SMF}$
$H1_3: UO_{CMF} > UO_{SMF}$	$H1_9: SCAU_{CMF} > SCAU_{SMF}$
$H1_4: TP_{CMF} > TP_{SMF}$	$H1_{10}: AC_{CMF} > AC_{SMF}$
$H1_5: TEIP_{CMF} > TEIP_{SMF}$	$H1_{11}: CAG_{CMF} > CAG_{SMF}$
$H1_6: NEPF_{CMF} > NEPF_{SMF}$	

3.2.5. VARIÁVEIS INDEPENDENTES

As seguintes entradas ou fatores estiveram presentes na experimentação.

- Projetos de software de porte pequeno;
- Equipes de projetos constituídas por alunos a realizar o primeiro contato com laboratório de software;
- Aplicação com e sem suporte de MEDITE+*FastInterface* realizada com a mesma ênfase quanto a interação do cliente, já que as bases de ambas abordagens, a habitual e a nova, usam o processo YP com usabilidade como plataforma.

3.2.6. VARIÁVEIS DEPENDENTES

As variáveis de efeito ou de saída são as seguintes:

- Quantidade de tarefas
- Nível de especificação da realização das tarefas
- Taxa de uso de operadores na tarefa
- Tamanho do protótipo
- Distribuição elementos de interação no protótipo
- Nível de especificação do projeto funcional
- Nível de realização do projeto
- Grau de facilidade de manejo e aprendizado dos artefatos de usabilidade
- Grau de satisfação do cliente com artefatos de usabilidade
- Nível de impacto na especificação de artefatos funcionais
- Nível de integração entre a análise da tarefa e a prototipagem

3.2.7. TRATAMENTO

O estudo contemplou um único fator de tratamento, referente à aplicação de MEDITE+*FastInterface* em YP.

3.2.8. SUJEITOS E AMOSTRAGEM

Os sujeitos utilizados foram alunos iniciantes em práticas de ES do curso Ciência da Computação da UFCG. Participaram do grupo de controle metade dos alunos do semestre 2008.2.

O grupo de tratamento contém os alunos de 2010.1 e parte dos alunos do semestre 2008.2. Nos semestres 2009.1 e 2009.2, os alunos não se utilizaram do apoio metodológico e ferramental proposto⁵.

Em ambos os grupos, de controle e tratamento, as equipes de alunos foram selecionadas de maneira aleatória já que no intervalo de semestres do grupo de controle se obteve uma população de 15 equipes (46 alunos), e no de tratamento de 12 equipes (34 alunos). A presença da aleatoriedade na amostragem minimiza perturbações como a cultura, histórico e comportamento aluno. De qualquer forma, a amostragem buscou ser homogênea quanto à verificação do nível de conhecimento em procedimentos para projeto de interface e de usabilidade pelos alunos. O questionário aplicado para o delineamento do perfil dos alunos encontra-se no Anexo B.

A realização de generalizações para os resultados do experimento deve ser realizada com restrições ao ambiente encontrado na disciplina LES, considerando: (i) um contexto de desenvolvimento ágil e centrado no usuário; (ii) intervalo de tempo curto, em torno de 3 meses e meio (semestre letivo) para a realização de projetos; (iii) projetos de médio e pequeno porte; e, ainda, (iv) o uso de recursos para *design* centrado no usuário.

Não seria inoportuno relacionar o impacto do tratamento em contextos de ensino semelhantes⁶, caso o tratamento venha a ser reproduzido. No entanto, em um contexto de condução de projetos de software de mercado, ou industrial, poder-se-ia pensar numa tendência de impacto semelhante, considerando uma configuração mínima de manejo da

⁵ Nos semestres do ano letivo de 2009 não foi possível obter amostras expressivas, devido à quantidade de alunos matriculados. Nesse período a ferramenta *FastInterface* passou por ajustes, em que se optou por aplicar nesses semestres a abordagem habitual, sem suporte da ferramenta. Dada essa configuração, os alunos não foram acompanhados presencialmente, em que os dados provenientes foram obtidos a partir da documentação de seus projetos ou por comunicação online.

⁶ Ensino de ES numa abordagem situacional, com laboratório de software.

usabilidade. Porém, não é possível se generalizar tanto assim, principalmente, considerando que há casos, no contexto de mercado, que exigem maior grau de detalhamento na elaboração do projeto da interface, sobretudo de seus atributos de estilo, o que, todavia, não é possível tratar com efetividade, com os atuais recursos de *FastInterface*.

A configuração das amostras do grupo de controle e de tratamento é resumida no Quadro 5, considerando já a mortalidade⁷.

Quadro 5 – Amostragem dos grupos de tratamento e de controle

	Grupo de Controle	Grupo de tratamento
Equipes (3 a 4 alunos)	12	12
Alunos envolvidos (total)	43	32

3.2.9. INSTRUMENTAÇÃO

Para facilitar a avaliação das métricas quantitativas estabelecidas, foram executados procedimentos para o levantamento de dados estatísticos descritivos, para a obtenção de médias, variâncias e desvios padrão, de cada variável, para os grupos de controle e experimental.

Considerando que não foi possível realizar medições antes e depois em uma mesma equipe, ou realizar medições com equipes formadas por integrantes diferentes realizando o mesmo projeto, com e sem a abordagem proposta, durante todo o período de coleta de dados, foi considerado que a comparação seria feita a partir de amostras de grupos independentes.

⁷ Perda de amostra ocasionada por insuficiência de informações pela equipe, ou pelo projeto do produto não envolver interação com o usuário.

Portanto, para realização dos testes de hipóteses foi adotado o teste *t* presumindo variâncias diferentes (Levine, 2000), já que os assuntos das equipes, ou seja, os domínios de problemas tratados foram diferentes em alguns casos. Caso fosse possível realizar o experimento de outra forma, o teste *t* pareado seria o mais indicado, pois se presumiria que medições foram realizadas antes e depois sobre os mesmos sujeitos e/ou assuntos.

Para compensar a generalização das comparações, foram adotados níveis de significância (α) mais altos, para rejeição da hipótese nula, em que ficou estabelecido o valor de 0,025. Conseqüentemente, a confiança percentual dos testes ($1 - \alpha$) equivale a 97,5%, ou seja, a probabilidade de rejeitar a hipótese nula quando de fato esta for verdadeira será de 2,5%.

A não rejeição da hipótese nula H_0 indica que não houve variação significativa entre os valores obtidos entre os grupos de controle e os experimentais. No quadro 6, explicitam-se os parâmetros de análise para o teste de hipóteses para as variáveis quantitativas.

Quadro 6 – Parâmetros para não rejeição de hipóteses nula e alternativa

$H_0 : \mu_1 = \mu_2$	Em que, $t_0 < -t_{\alpha, v}$
$H_1 : \mu_1 < \mu_2$	

3.2.10. AVALIAÇÃO DA VALIDADE

Os itens para a avaliação da validade deste experimento se basearam em descrições nos seguintes trabalhos de educação, e de educação em ES (Briggs, 2008; Lethbridge *et al.*, 2007):

- **Validade Interna**

Conforme mencionado na seção 3.2.8, o estudo se propôs a utilizar o contexto de ensino de ES, numa abordagem situacional,

com laboratório de software orientado por um processo de desenvolvimento. Nenhum dos integrantes da equipe deve possuir conhecimento prévio em técnicas, métodos e formalismos de IHC. As equipes do grupo de tratamento foram informadas sobre o manejo da usabilidade da maneira habitual. Quanto às equipes do grupo de tratamento, o nível de informação se concentrou em preceitos de MEDITE e restrições de uso de *FastInterface*. Em ambos os casos, se evitou realizar comentários ou intervenções a respeito da qualidade dos artefatos (de usabilidade ou funcionais) gerados durante o período estabelecido para construção dos mesmos.

- **Validade de Conclusão**

Para a verificação do impacto, as hipóteses serão verificadas a partir do desempenho das variáveis quantitativas e qualitativas entre os grupos de controle e tratamento.

- **Validade de Construção**

Na investigação, consideram-se várias fontes de dados (artefatos funcionais, artefatos de usabilidade, questionários, anotações). Houve um acompanhamento das avaliações, presencial ou documentalmente, especialmente nos releases de YP, para verificação de perturbações para as equipes do grupo experimental. O emprego de amostragem aleatória e a definição de métricas diretamente aplicáveis às hipóteses formuladas, além da diversidade da coleção de dados, permitiram melhorar o grau de confiança do estudo.

- **Validade Externa**

Com restrições. Não se pode desconsiderar as tendências dos resultados serem semelhantes em outros contextos ágeis, com desenvolvimento centrado no usuário e em âmbito de ensino prático de ES, via laboratório. Nesse caso, deve-se considerar a robustez de YP como PDS, agregando práticas de outros

processos; e também de MEDITE, por se constituir de um arcabouço de concepção de interfaces que reúne práticas bastante consolidadas de IHC. Por outro lado, em contextos fora da academia ou de ensino e/ou que venham a exigir maior tratamento a respeito do protótipo de alta-fidelidade (questões de estética/estilo) não se pode generalizar tanto, pois o experimento não cobriu o projeto de usabilidade nesse nível. Ainda, considerando a impossibilidade de obtenção de mais sujeitos para as amostras obtidas em ambos os grupos de controle e de tratamento, as conclusões desse experimento podem indicar apenas tendências.

3.2.11. CONSIDERAÇÕES SOBRE FINAIS SOBRE A CONFIGURAÇÃO DO EXPERIMENTO

Neste capítulo foi realizada a descrição do experimento a fim de verificar o impacto do apoio metodológico e ferramental de MEDITE+*FastInterface*, utilizando-se o ambiente do laboratório de software, do curso de computação da UFCG. A configuração do experimento buscou a obtenção de dados que pudessem dar subsídios a verificação do comportamento tanto de aspectos funcionais, quanto do projeto de interface do software. Esse desenvolvimento será apresentado e discutido no próximo capítulo.

Os dados obtidos ao longo dos semestres letivos foram arranjados em dois grupos, o de controle e o experimental, de forma a possibilitar a realização de inferências de natureza comparativa, facilitando a mensuração do impacto da abordagem proposta. É importante considerar que as métricas quantitativas, de uma forma geral, objetivam verificar a qualidade e o nível de realização dos artefatos de usabilidade e dos projetos de software. De forma a facilitar a realização

de inferências, as métricas qualitativas objetivaram mensurar os níveis de compreensão e satisfação com a abordagem sugerida.

As variáveis qualitativas, por serem baseadas em opiniões, também buscaram identificar indícios sobre os resultados, quer sejam eles favoráveis ou desfavoráveis na aplicação da abordagem proposta.

CAPÍTULO 4

APRESENTAÇÃO E ANÁLISE DE RESULTADOS

Neste capítulo, expõem-se e analisam-se os resultados obtidos nas variáveis qualitativas e quantitativas a partir da realização de inferências causais.

4.1. VARIÁVEIS QUANTITATIVAS OU INDICADORES OBJETIVOS

As variáveis quantitativas contemplam as 7 primeiras métricas, expostas na seção 3.2.3: (i) quantidade de tarefas (QT); (ii) taxa de realização das tarefas (TRT); (iii) uso de operadores (UO); (iv) tamanho do protótipo (TP); (v) taxa de elementos de interação no protótipo (TEIP); (vi) nível de especificação do projeto funcional (NEPF); (vii) nível de realização do projeto (NRP).

O valor de cada uma destas variáveis está relacionado diretamente ao desempenho da equipe sobre o aspecto mensurado. Em seguida, os valores das equipes incluídas no grupo de controle foram confrontados com os valores das equipes incluídas no grupo de tratamento, por meio da administração do teste t , conforme descrito na seção 3.2.9. A seguir, serão apresentados os resultados obtidos na administração do teste t para cada uma das variáveis quantitativas supracitadas.

i. Quantidade de tarefas (QT)

A métrica QT mensura o nível de detalhamento das tarefas, a partir do modelo de tarefas elaborado por cada equipe. Neste sentido, se espera que os modelos de tarefas possuam níveis de refinamento melhores

para as equipes que utilizaram a abordagem experimental MEDITE+*FastInterface*. O nível de detalhamento considera uma melhora significativa na quantidade de tarefas especificadas por parte do tratamento, em que uma relativa melhora na quantidade poderia indicar que a nova abordagem contribuiu para uma melhor especificação de tarefas. Logo, buscou-se verificar se a hipótese alternativa $H_{1_1}: QT_{CMF} > QT_{SMF}$ ocorreu.

No item A.1. (Apêndice A) se verifica que a hipótese nula é rejeitada, e que os valores de t indicam um aumento significativo do número de tarefas no grupo de controle.

ii. Taxa de realização das tarefas (TRT)

A comparação da taxa de realização das tarefas permite que se possa visualizar se as mesmas estão sendo decompostas em mais ações. Neste caso, queremos confirmar a afirmação da hipótese $H_{1_2}: TRT_{CMF} > TRT_{SMF}$. Verificou-se que não há diferença significativa entre os dois grupos, controle e tratamento, quanto a esta métrica, vide item A.2 (Apêndice A).

iii. Uso de Operadores (UO)

Para todos os operadores disponíveis na especificação de tarefas (SEQ, OR, XOR, AND, SIM e PAR) não houve alteração significativa em seu emprego nos modelos de tarefa do grupo de controle e experimental conforme se verifica nos itens A.3 a A.6. (Apêndice A). Nesse caso a hipótese $H_{1_2}: TRT_{CMF} > TRT_{SMF}$ é rejeitada.

Os operadores SIM e PAR que, em geral, são pouco utilizados, não foram empregados em nenhuma tarefa dos modelos das equipes do grupo de controle e experimental. Isto demonstra que não é muito

frequente a presença de execução de tarefas em simultaneidade ou paralelismo.

iv. Tamanho do protótipo (TP)

Esta métrica busca inferir sobre a especificação/detalhamento dos protótipos de interface produzidos pelas equipes a partir da quantidade de telas geradas. Se espera que quanto mais telas o protótipo contiver melhor seja o nível de detalhe no esboço dos itens da futura interface. O item A.7. (Apêndice A) aponta indícios de que a hipótese $H_{14}: TP_{CMF} > TP_{SMF}$ é verdadeira.

v. Taxa de elementos de interação no protótipo (TEIP)

A partir desta taxa, tenta-se verificar a média de objetos de interação (botões, rótulos, campos de texto, etc.) presentes por tela no protótipo. Quanto maior o nível infere-se que houve maior preocupação em explorar os elementos para realização das tarefas no sistema. A partir desta métrica, também é possível verificar se as telas dos protótipos tendem a contemplar somente uma visão geral.

Por exemplo, imagina-se uma equipe que dispôs somente divisões de tela (visões) com apenas um objeto de interação em seu interior, num conjunto de 4 telas do protótipo. Logo, a taxa TEIP será baixa, ou seja, não haverá maior uso de objetos de interação para exploração das funcionalidades em tela. Esta métrica foi criada, pois, em observações do cenário LES, anterior à realização do experimento, a maioria das equipes desenvolvia protótipos com essa característica de superficialidade.

Conforme se visualiza no item A.8. (Apêndice A), a hipótese $H_{15}: TEIP_{CMF} > TEIP_{SMF}$ pôde ser confirmada, sendo as equipes de

tratamento mais eficientes na distribuição de objetos de interação pelo protótipo. As causas para isto podem ter relação com o fato do protótipo abstrato, gerado automaticamente por *FastInterface*, ter disponibilizado uma noção mais abrangente da navegação da interface, por cobrir obrigatoriamente todo o modelo de tarefas realizado em etapa anterior. Isto também é consequência da evidência de presença de maior quantidade de tarefas relativas às equipes do grupo experimental.

Apesar dos indícios de que os protótipos com telas passaram a cobrir todo o modelo de tarefas nas equipes do grupo experimental, não significa que o protótipo sugerido tenha correspondido totalmente às expectativas quanto aos tipos de objetos de interação (e. g, listas de seleção múltipla e única, botões, menus, etc.) atribuídos e as relações entre telas sugeridas pela ferramenta, ao menos logo após a transformação do modelo de tarefa em protótipo de média-fidelidade.

Entretanto, a possibilidade de edição do protótipo, após sua geração automática, encoraja os alunos a realizarem a adequação das partes do protótipo que não corresponderam às expectativas com menos esforço: o objeto sugerido pela ferramenta, ainda que não seja do tipo correto, possui relação direta com o nome da tarefa, auxiliando o aluno na realização dos ajustes para se alcançar objetivo de realizar a tarefa correspondente.

vi. Nível de especificação do projeto funcional (NEPF)

Esta métrica retrata se as projeções feitas pelos alunos, no que se refere a casos de uso, foram bem especificadas e se os testes de aceitação acompanham esta tendência. Conforme o item A.9. (Apêndice A), não houve variação significativa no nível de especificação entre os grupos de controle e de tratamento, pois a hipótese nula não é

rejeitada.

A verificação de métrica NEPF considerou que na suposição de que o modelo de tarefas tenha sido especificado em maior nível de detalhe (o desempenho das variáveis QT, TP e TEIP apontaram essa tendência) o mesmo possa ter contribuído para uma melhor compreensão dos domínios de problema dos projetos do grupo experimental, em que isso poderia ter verificado na relação entre *user stories* e testes de aceitação.

Apesar da rejeição da hipótese nula de NEPF e considerando a proximidade das médias em seus valores entre as equipes do grupo de controle e experimental, deve-se considerar ao menos que o suporte metodológico ferramental não prejudicou a especificação do projeto funcional em relação ao grupo de controle. De certa forma, isto pode ser positivo, considerando que durante a aplicação da abordagem experimental se corria o risco desse desempenho ser prejudicado nas equipes do grupo experimental, no uso da abordagem MEDITE + *FastInterface*.

vii. Nível de realização do projeto (NRP)

A variável NRP mede a relação entre a quantidade de *user stories* especificadas e a realização entre casos de uso. Também, sob aspecto, conforme demonstra o item A.10 (Apêndice A) não houve variação significativa entre as equipes do grupo de controle e experimental, portanto a hipótese nula não é rejeitada para esse aspecto.

Considerando a possibilidade da abordagem MEDITE + *FastInterface* que poderia ter impactado negativamente sob a realização dos projetos, verifica-se, pela relação entre os valores das médias, que não houve efetivamente diferença sob este aspecto.

4.2. VARIÁVEIS QUALITATIVAS OU INDICADORES SUBJETIVOS

As variáveis qualitativas contemplam à análise dos questionários aplicados após a fase de planejamento de YP, no meio da etapa de planejamento (release intermediário) ou após o último release, para cada membro da equipe e respectivos clientes. Os indicadores qualitativos são: (i) facilidade de manejo e aprendizado dos artefatos de usabilidade; (ii) satisfação do cliente com artefatos de usabilidade; (iii) aprendizado colaborativo; e (iv) correlação entre artefatos gerados.

Foram consideradas as questões com respostas mais diretamente relacionadas para avaliação dessas hipóteses. A análise foi obtida a partir da distribuição das frequências para cada reação (opção) da questão considerada.

i. Facilidade de manejo e aprendizado dos artefatos de usabilidade

A respeito desta variável, foram consideradas as respostas dos alunos ao *Questionário Pós Etapa de Planejamento (QPEP)* e *Questionário Pós Release Final (QPRF)*. As questões mais significativas serão demonstradas a seguir.

Questão 2 (QPEP) - *Reprodução dos requisitos obtidos na conversa com o cliente dentro do modelo de tarefas com os recursos que você utilizou foi:*

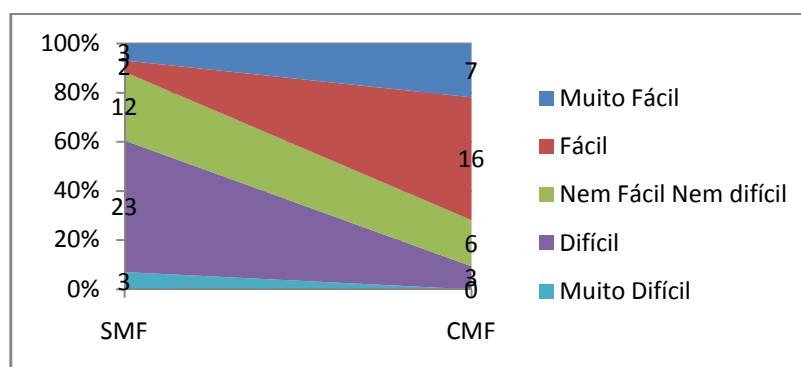


Gráfico 1 – Facilidade de reprodução de requisitos com modelo de tarefas

Questão 3 (QPEP) - *Compreensão do modelo de tarefas obtido pela sua equipe foi:*

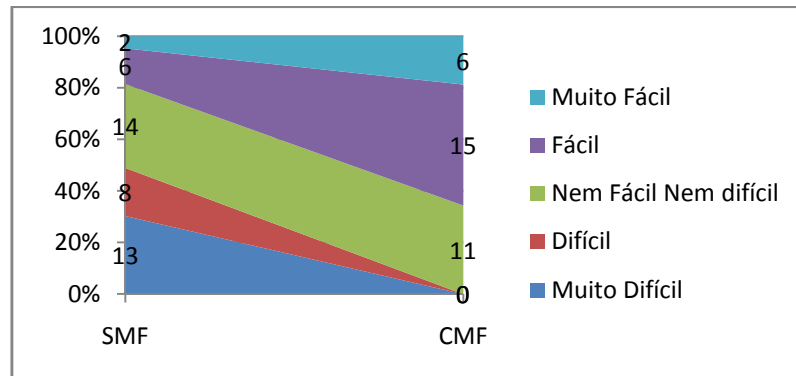


Gráfico 2 – Compreensão do modelo de tarefas pelas equipes

Questão 12 (QPEP) - *Indique o nível de esforço para construção do protótipo de interface com os recursos utilizados.*

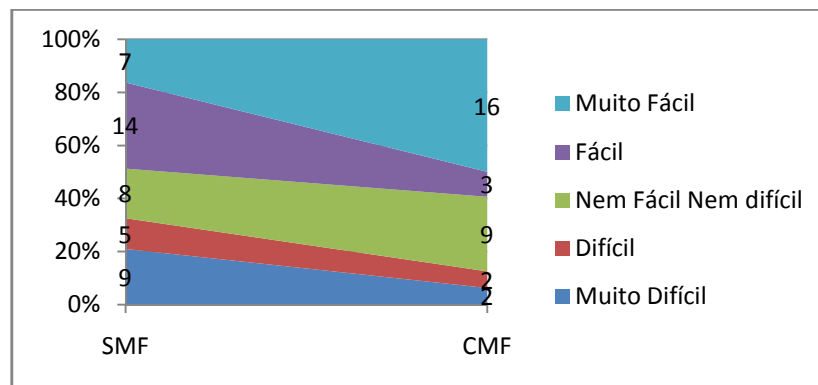


Gráfico 3 – Nível de esforço na construção dos artefatos de usabilidade

Questão 11 (QPRF) - *Se fosse pedido a você desenvolver o modelo de tarefas de um sistema você se sentiria habilitado considerando seu atual grau de compreensão?*

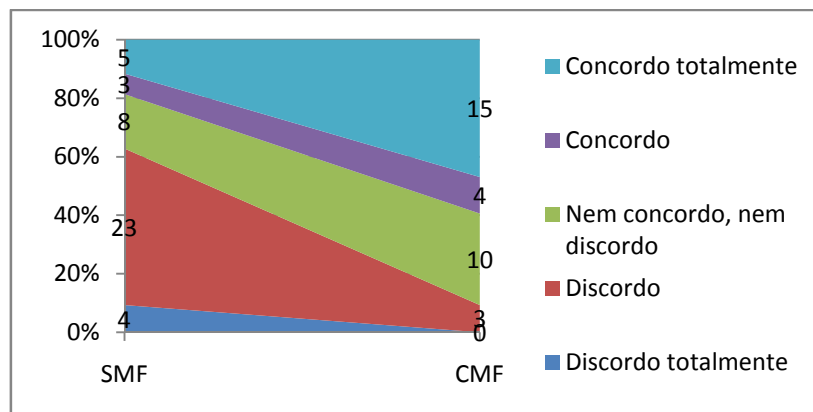


Gráfico 4 – Segurança em reproduzir o modelo de tarefas ao término do projeto de software

Observa-se a tendência relativa à facilidade de trabalho e melhor compreensão do modelo de tarefas no caso da abordagem com MEDITE+*FastInterface* (CMF), conforme verificado na questão 2 (no questionário QPEP). Ainda, se percebe que a maioria dos alunos do grupo sem suporte de MEDITE+*FastInterface* (SMF) não se sentiram tão confiantes em elaborar novamente um modelo de tarefas ao término da disciplina conforme a questão 11 (questionário QPRF).

Isto evidencia o problema na especificação de tarefas anterior à aplicação do tratamento, que foi melhorado devido a exigência de *FastInterface* por modelos de tarefas com o mínimo de especificação para possibilitar a geração do protótipo de média-fidelidade. Desta forma, a hipótese $H_{18}: FMAU_{CMF} > FMAU_{SMF}$ pode ser aceita.

Sobre esforço ao se produzir os protótipo de interface verificou-se que os alunos do grupo experimental acharam muito mais fácil lidar com a elaboração do protótipo, conforme questão 12 (questionário QPEP), embora os alunos do grupo de controle também tenham declarado em um nível relativamente menor (ao invés de muito fácil, apenas fácil) a elaboração deste artefato. Ainda, analisando a reatividade das respostas nos níveis nível difícil e muito difícil houve uma melhora no

grupo experimental em que mais alunos passaram a achar nem fácil, nem difícil (indiferente) o nível de esforço.

ii. Satisfação do cliente com artefatos de usabilidade

Os dados abaixo consideram o *Questionário Release Intermediário (QRI)* e o *Questionário Pós Release Final (QPRI)* para clientes sendo oportuno apresentar os dados em que as equipes utilizaram a abordagem *MEDITE+FastInterface*.

Questão 1 (QRI) - *O que você achou do modelo de tarefas da interface de seu projeto?*

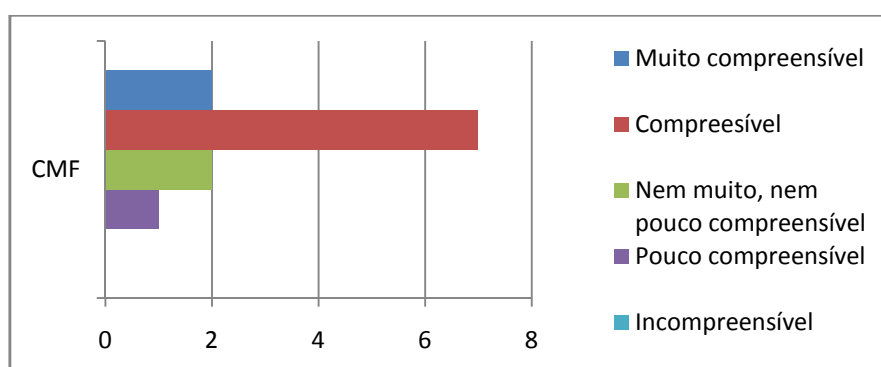


Gráfico 5 – Compreensão do modelo de tarefas da interface pelos clientes

Questão 2 (QRI) - *Você acredita que o modelo de tarefas auxiliou para uma melhor compreensão de funcionalidades de seu projeto?*

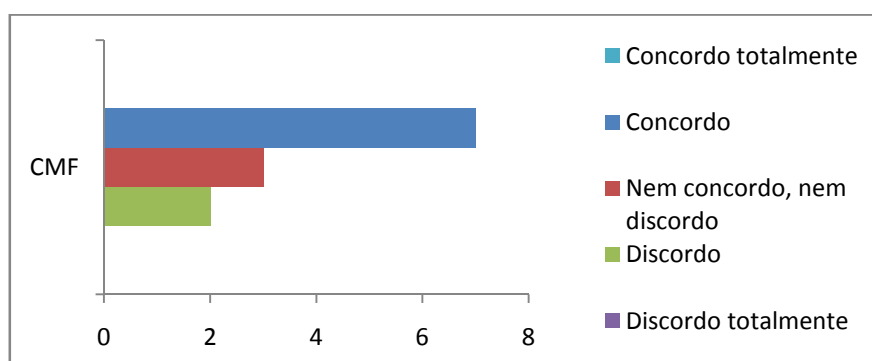


Gráfico 6 – Contribuição do modelo de tarefas na compreensão funcional do projeto pelos clientes

Questão 3 (QRI) - *O protótipo da interface que lhe foi apresentado, por sua equipe, atendeu às suas expectativas?*

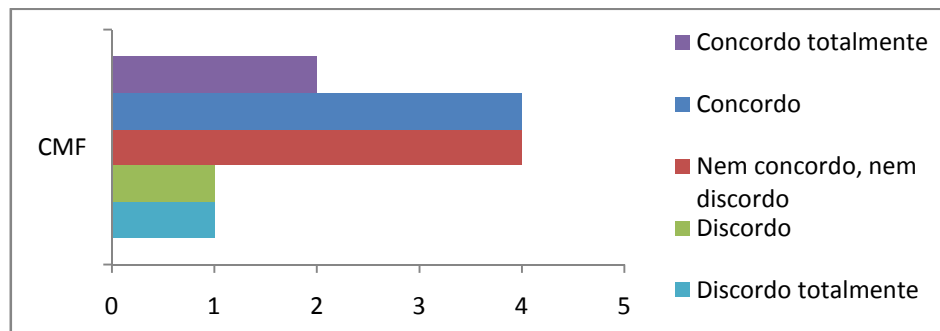


Gráfico 7 – Atendimento do protótipo de interface às expectativas do cliente

Questão 7 (QRI) - *Você achou interessante e útil o uso do modelo de tarefas e do protótipo de interfaces em seu projeto?*

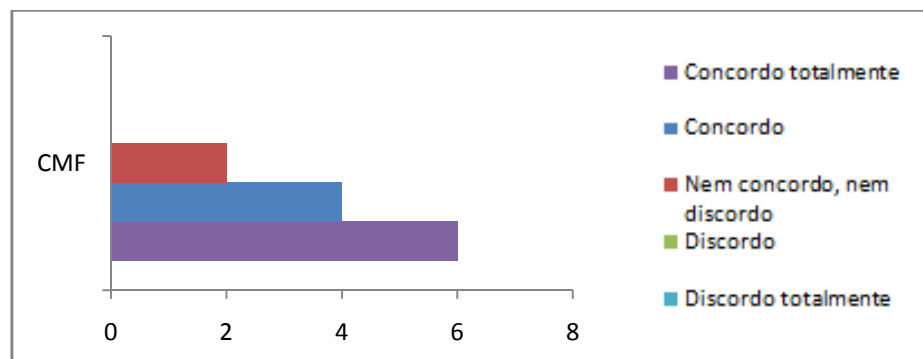


Gráfico 8 – Utilidade do modelo de tarefas e protótipo de interface no projeto para o cliente

A hipótese $H_{10}: SCAU_{CMF} > SCAU_{SMF}$ não pôde ser verificada em totalidade, dada a análise de apenas parte do grupo experimental. No entanto, considerando a possibilidade de perturbações no uso do modelo de tarefas e do protótipo junto ao cliente com os artefatos gerados com *FastInterface*, o que não ocorreu, pode-se considerar que os mesmos foram bem aceitos e bem apreciados pelos clientes.

iii. Aprendizado colaborativo (melhor comunicação intra-equipe)

As questões mais significativas relativas a uma melhor comunicação intra-equipe serão mostradas a seguir. Elas estão inseridas no contexto do *Questionário Pós Release Final (QPRF)*, para alunos.

Questão 6 (QPRF) - *O modelo de tarefas e o protótipo permitiram uma melhor comunicação entre os membros da equipe (internamente) sobre o projeto do software.*

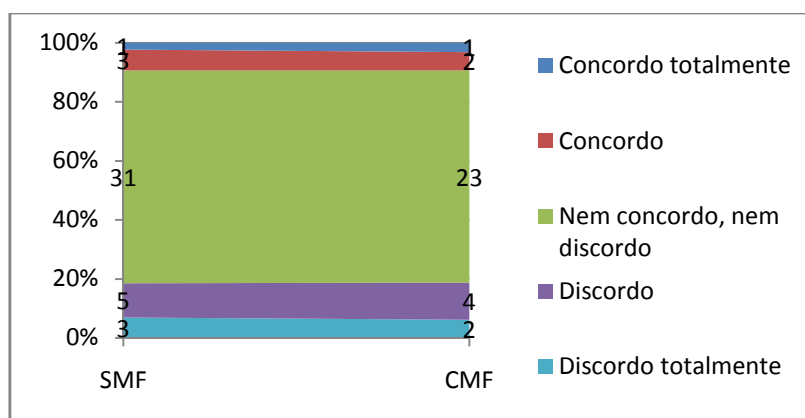


Gráfico 9 – Impacto do modelo de tarefas e do protótipo na comunicação das equipes

Considerando as variações entre o nível de comunicação interno na equipe, foi observada uma semelhança no grau de comunicação entre as equipes, o que indica que não houve diferenças em relação ao grupo experimental. Isto, de certa forma, pode evidenciar uma independência sobre questões de comunicabilidade interna para com os artefatos de usabilidade, em que outras variáveis devem ser consideradas. Portanto, desta forma, a hipótese nula, $H_{0_{10}}: AC_{CMF} > AC_{SMF}$ não é rejeitada.

iv. Correlação entre artefatos gerados

A respeito desta variável foram consideradas as respostas dos alunos ao *Questionário Pós Etapa de Planejamento (QPEP)*. As questões mais significativas serão demonstradas a seguir.

Questão 6 (QPEP) - Acho que o modelo de tarefas auxiliou na compreensão das funcionalidades do sistema (na decomposição de suas funções).

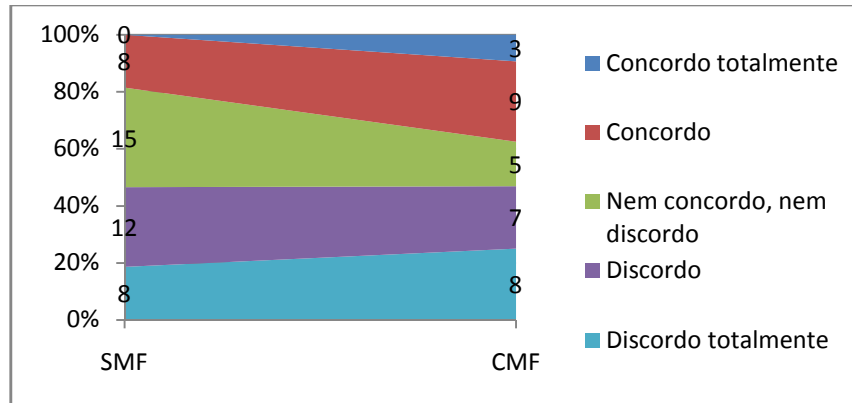


Gráfico 10 – Impacto do modelo de tarefas na decomposição funcional do software pelas equipes

Questão 7 (QPEP) – Acho que o modelo de tarefas obtido contribuiu para uma melhor especificação e/ou compreensão de user stories e testes de aceitação.

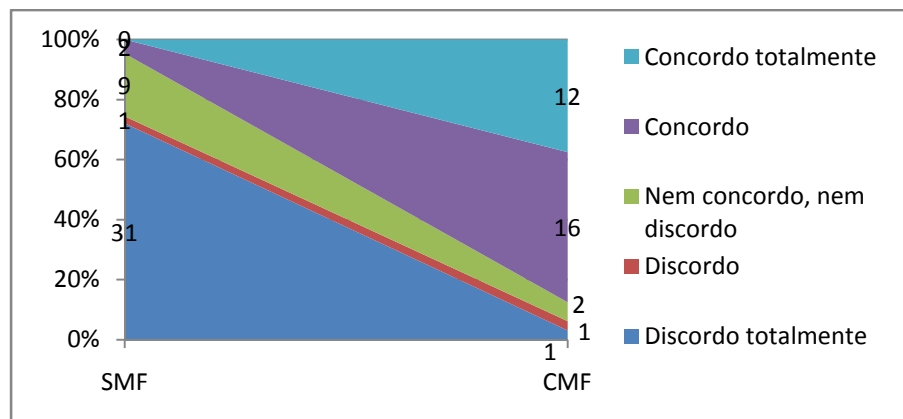


Gráfico 11 – Impacto do modelo de tarefas na compreensão de US e TAs

Questão 8 (QPEP) - Acho que a modelagem da tarefa, por ampliar a compreensão dos requisitos e a forma de utilização do sistema, contribui para uma especificação mais realista do planejamento de iterações e releases.

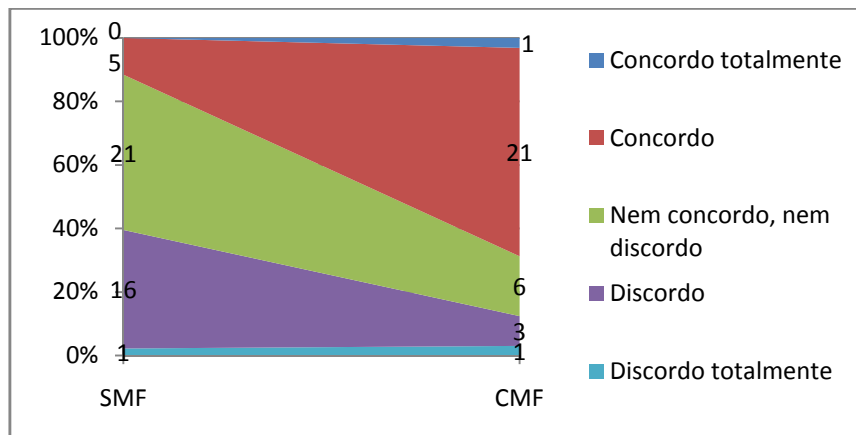


Gráfico 12 – Impacto do modelo de tarefas no planejamento de iterações e releases

Questão 15 (QPEP) - *Acredito que houve influência do modelo de tarefas na concepção do protótipo.*

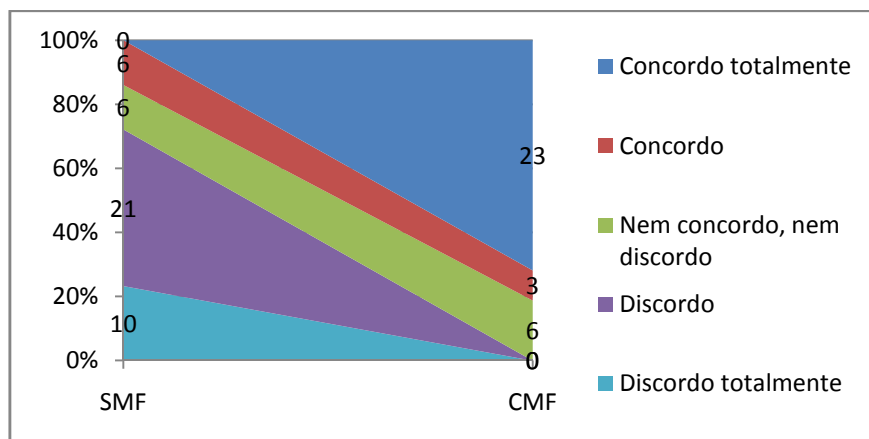


Gráfico 13 – Influência do modelo de tarefas na concepção do protótipo de interface

Observa-se uma tendência de melhora no emprego de MEDITE+FastInterface, especialmente considerando tendências de maior influência do modelo de tarefas no protótipo de interfaces conforme evidencia a questão 15 (questionário QPEP). Estabelecendo uma relação causal, isto pode ser justificado pela promoção da integração entre os artefatos por parte da ferramenta: uma versão

inicial do protótipo pode ser obtida automaticamente a partir do modelo de tarefas elaborado em etapa anterior.

Às declarações dos alunos sobre maior percepção de influência do modelo de tarefas em artefatos funcionais (vide questões 6, 7 e 8 do questionário QPEP) podem ser justificadas pela obtenção de melhor especificação da tarefa, imposta por *FastInterface* para geração automática do protótipo (conforme as métricas QT, TP e TEIP evidenciaram): com o modelo de tarefas melhor especificado os alunos tiveram a oportunidade de apreciar melhor o domínio do problema, o que é mais fácil, pois a visão externa, de maior abstração.

Um bom nível de maturação na análise do domínio do problema auxilia na compreensão de aspectos como escopo e complexidade de funcionalidades, o que impacta num esforço menor no planejamento de estimativas e no escalonamento de atividades, pois se conhecem melhor os detalhes das tarefas que o sistema disponibiliza para o usuário.

Portanto, a partir das declarações dos alunos podemos concluir que a hipótese $H_{11}: CAG_{CMF} > CAG_{SMF}$, que verifica a correlação entre os artefatos de usabilidade elaborados foi confirmada.

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

Neste capítulo, apresentam-se as considerações finais da dissertação, a partir da condução do experimento e dos resultados obtidos. Também são descritos o impacto da abordagem proposta no laboratório de software, bem como suas restrições de reprodução e aplicação. Ainda, são apresentadas algumas proposições para a realização de trabalhos futuros.

5.1. REVISÃO DA PROBLEMÁTICA DA FALTA DE APOIO METODOLÓGICO E FERRAMENTAL PARA USABILIDADE NO LABORATÓRIO DE SOFTWARE

Neste trabalho foi proposta a verificação do problema de adequação de técnicas de projeto de interface/usabilidade com ênfase no ensino situacional da ES: o laboratório de software. Nesse contexto, foi resgatada a relevância na formação do desenvolvedor de software sobre a necessidade de proporcionar uma visão integrada, tanto da parte interna, referente aos componentes funcionais do software, quanto de sua parte externa, referente ao uso e acesso de suas funcionalidades.

Foram apresentadas as possíveis causas da concorrência entre as visões de ES e de IHC no contexto de ensino, em que a visão de IHC acaba por ficar numa posição menos privilegiada devido às restrições de tempo e de esforço na elaboração e compreensão dos procedimentos que a visão IHC sugere. Essas dificuldades foram detectadas junto aos alunos da disciplina LES da UFCG em relação ao tratamento dos artefatos de usabilidade preconizados pelo processo ágil e centrado no usuário *easYProcess*, em especial quanto ao modelo de tarefas e ao protótipo de interface.

Conforme verificado no contexto do laboratório de software da UFCG, ainda que um laboratório de software se utilize de um processo centrado no usuário ou no *design* interativo na tentativa de proporcionar uma integração de aspectos funcionais e de usabilidade, é interessante que o processo indique objetivamente o modo como os artefatos de usabilidade serão construídos, de preferência se apoiado numa metodologia de concepção de interfaces com suporte ferramental subjacente.

O apoio ferramental contribui para redução dos esforços de elaboração, compreensão e uso dos artefatos de usabilidade e, por conseguinte, dos níveis de satisfação, corretude e acreditação dos mesmos por parte dos alunos. Esta constatação de que um apoio ferramental via software é importante para a manipulação dos artefatos é mais consolidada na visão funcional de desenvolvimento de software a exemplo das ferramentas para modelagem UML, gerência de projetos, ou de automatização, cujas práticas são mais encorajadas quando um suporte ferramental computacional é possível. Isso, inclusive, auxilia na aplicabilidade, disseminação e acreditação das técnicas subjacentes à essas ferramentas.

5.2. CONCLUSÕES QUANTO AO IMPACTO METODOLÓGICO E FERRAMENTAL SUGERIDO A PARTIR DO EXPERIMENTO

Conforme a experimentação da metodologia de concepção de interfaces MEDITE, com o apoio de sua ferramenta *FastInterface*, se pôde detectar ao longo do desenvolvimento do software no laboratório, os seguintes aspectos positivos:

- **Melhora na redução de esforço e da qualidade dos artefatos produzidos pelos alunos**, com destaque para o modelo de tarefas, que aparentava anteriormente ser de difícil elaboração e

compreensão. A facilidade de edição e operacionalização das tarefas, proporcionada pela ferramenta, fez com que os que a utilizaram apresentassem melhores níveis de especificação do universo de tarefas de seu projeto. O mesmo ocorreu com os protótipos de interface, em que os que utilizaram o novo suporte obtiveram protótipos com maior quantidade de telas e de elementos (vide os resultados quantitativos obtidos, relativos à quantidade de tarefas, tamanho do protótipo e taxa de elementos de interação por tela no protótipo). Os alunos que utilizaram o suporte declararam em relação aos outros menores níveis de esforço, o que se atribui a sugestão da interface inicial por meio da geração do protótipo a partir do modelo de tarefas, proporcionado pela ferramenta.

- **Esforço de aprendizado e confiança dos alunos**, em que àqueles que utilizaram o suporte, em sua maioria, declararam possuir confiança no desenvolvimento do modelo de tarefas em posteriores demandas de desenvolvimento de software. Isso se deve a melhor compreensão de como elaborar e interpretar o modelo de tarefas, proporcionado com o uso da ferramenta.
- **Correlação entre modelo de tarefas e protótipo**. Foi possível inferir que a geração automática pôde contribuir para o relacionamento entre o modelo de tarefas e do protótipo, que no caso do *FastInterface*, quanto maior o nível de especificação da tarefas maior a probabilidade do protótipo inicial gerado pela ferramenta conter mais elementos.
- **Satisfação dos clientes**. Estes relataram em maiores níveis que puderam ter uma compreensão de como a interface proverá o acesso às funcionalidades, por meio do entendimento e da satisfação com os artefatos apresentados por suas respectivas equipes de desenvolvimento. Essas constatações evidenciam que

é praticável o uso e avaliação desses artefatos pelos clientes, que passam a ser apresentados mais satisfatoriamente.

É importante salientar que outras métricas qualitativas e quantitativas relacionadas à visão funcional não indicaram presença de diferenças significativas no andamento dos projetos entre as equipes que usaram e não utilizaram MEDITE+*FastInterface*. Esta constatação deve ser feita, em observação a possibilidade de que a abordagem proposta e o uso da ferramenta poderiam num pior caso perturbar o desenvolvimento das equipes.

Desta forma, deduz-se que os resultados do experimento evidenciaram um impacto positivo entre os alunos e respectivas equipes que se beneficiaram do suporte metodológico e ferramental de MEDITE+*FastInterface*. Apesar da impossibilidade de generalizar os resultados estatisticamente, devido a problemas de natureza amostral, de configuração e de acompanhamento; não se pode desconsiderar a relevância do impacto positivo que o suporte proporcionou aos alunos e clientes, no aprendizado e emprego de técnicas voltadas para o desenvolvimento da visão de IHC no desenvolvimento de software.

5.3. CONSIDERAÇÕES ESPECÍFICAS À RESPEITO DA DISCIPLINA LABORATÓRIO DE ENGENHARIA DE SOFTWARE DA UFCG

A partir das observações realizadas durante o período do experimento, na disciplina LES da UFCG, foram produzidos guias com recomendações e sugestões para facilitar: (i) a compreensão, elaboração e uso dos artefatos de usabilidade, por parte dos alunos (Anexo E); e (ii) a aplicação da abordagem de suporte metodológico e ferramental MEDITE+*FastInterface* no laboratório de software, fornecendo subsídios aos professores para um aproveitamento mais bem-sucedido da abordagem junto aos alunos e clientes (Anexo F).

Entretanto, se faz necessário tecer alguns comentários específicos ao contexto da disciplina LES da UFCG com base na administração do suporte metodológico e ferramental sugerido por este trabalho, em relação:

- **À contextualização da disciplina na grade da UFCG.** Que, todavia, é apenas precedida por disciplinas com conhecimentos gerais de Engenharia de Software. Essa realidade não dificultou o uso da abordagem junto aos alunos durante as experimentações, desde que sejam apresentados, ainda que de forma expressa, os conceitos relativos à análise de tarefas e à prototipagem evolutiva. Entretanto, pode ser interessante que o laboratório seja precedido por uma disciplina de Interface Homem-Computador, que contemplasse conceitos gerais de projeto de interfaces. Posteriormente, a disciplina de laboratório de software poderia servir como integradora, na prática, utilizando o suporte metodológico proposto, dos conceitos sobre a visão funcional e de usabilidade de software. Ainda, poderia haver a antecipação de uma disciplina de IHC, que na UFCG todavia é opcional, em relação ao laboratório de software, o que contribuiria para uma prévia preparação dos alunos na lida com os artefatos de usabilidade; bem como para o esforço do professor do laboratório, o qual poderia, nesse caso, apenas se ater à recomendações de uso desses artefatos.
- **À configuração das equipes e o cliente.** Em LES a média de integrantes varia entre 3 e 4 alunos. Em geral, as equipes que relataram algum problema tinham dificuldades na colaboração entre seus integrantes ou no acesso a seus clientes e usuários. Ainda, é importante que o cliente esteja ciente de sua responsabilidade de cobrar e dar retorno sobre os artefatos de usabilidade. Durante a avaliação do modelo de tarefas e do

protótipo, o professor do laboratório poderia exigir da equipe que os mesmos terão validade somente com o parecer do cliente e que tais informações serão posteriormente coletadas com o mesmo, para efeito de avaliação da etapa de Planejamento.

- **Tempo e escopo para realização dos projetos.** Em LES, como uma disciplina de graduação, o tempo para realização dos projetos dura um semestre letivo. Não se verificou problemas quanto ao curto espaço de tempo, desde que o escopo dos projetos seja compatível; e que haja efetiva colaboração entre integrantes das equipes e seus clientes. Ainda, especificamente em relação ao tempo para obtenção dos artefatos da etapa de Planejamento, que inclui os artefatos de usabilidade, é recomendável que as equipes tenham duas semanas para fazê-lo, considerando a disponibilidade do cliente, que em LES é algo real. Durante o experimento, verificou-se que o suporte metodológico e ferramental proporcionou aos alunos uma obtenção mais imediata do modelo de tarefas e o protótipo, que fornece subsídios para redução do intervalo de tempo até a avaliação e refinamento dos mesmos, considerando o intervalo de duas semanas.

As considerações supracitadas foram baseadas no desenvolvimento das equipes dos alunos ao longo das experimentações do suporte metodológico e ferramenta de MEDITE+*FastInterface*, em que, de uma forma geral, as equipes que o utilizaram obtiveram melhorias em relação ao nível de especificação do modelo de tarefas e do protótipo de interface; além da facilidade de elaboração e aplicação desses artefatos. Desta forma, essas constatações fornecem subsídios para a recomendação do suporte ferramental de MEDITE+*FastInterface* no contexto de LES e até em outras disciplinas similares de laboratório de software.

5.4. TRABALHOS FUTUROS

Considerando o tempo restrito para concretização deste trabalho de dissertação existem algumas proposições de trabalhos futuros, enumeradas a seguir:

- Ampliar o experimento sobre aplicação da abordagem MEDITE+*FastInterface* em outros contextos de desenvolvimento ágeis e centrados no usuário, a fim de realizar um processamento estatístico que permita ampliar o grau de validade externa do experimento, sem restrições;
- Analisar e implementar sugestões de melhoria inferidas sugeridas pelos alunos com o uso de *FastInterface*. Dentre as sugestões destacam-se: (i) a abertura quanto às limitações, na edição do protótipo no módulo de prototipagem de média-fidelidade, quanto à conversão de objetos de interação sugeridos após a geração do protótipo; (ii) sugestão direta no módulo de análise da tarefa quanto a quais tarefas devem ser melhor descritas para sua transformação no protótipo, quando o usuário tentar gerar o protótipo e não conseguir, e assim reduzir ainda mais o esforço em sua obtenção;
- Avaliar os produtos finais das equipes, com o uso de técnicas de avaliação como inspeções de conformidade, observação direta do uso e sondagem da satisfação subjetiva. Essas avaliações poderiam ser realizadas de forma mais contínua entre os semestres letivos a fim de verificar se há continuidade do impacto positivo verificado por este trabalho, em nível de avaliação da usabilidade. Isso poderá ser realizado na medida em que a prototipagem em alta-fidelidade seja melhor suportada pela ferramenta *FastInterface*.

Referências

- AGUIAR, Y. P. C. *SMILE: uma ferramenta para a geração automática, edição e simulação de protótipo de interface do usuário*. Campina Grande, 2007. 201f. Dissertação (Mestrado) Departamento de Sistemas e Computação, Universidade Federal de Campina Grande, Campina Grande, 2007.
- AMBLER, S. *Agile Modeling: effective practices for extreme programming and unified process*. 1.ed. Wiley: Adobe E-Book, 2002.
- BANSKTON, A. *usability and user interface design in xp*. 2002. Disponível em: <<http://www.ccpace.com/Resources/documents/UsabilityinXP.pdf>>. Acesso em: 09 fev. 2010.
- BECK, K. *Extreme programming explained: embrace change*. 1.ed. [S.l]: Addison-Wesley, 2000.
- BERNARDINO, C. B. *Design interativo em processos ágeis de desenvolvimento de software*. Recife: UFPE, 2005.
- BOEHM, B. *Get ready for agile methods, with care*. IEEE Computer, v. 35, n. 1, p. 64-69, 2002.
- BRAGA, J. C. *Diretrizes para o Ensino Interdisciplinar de Engenharia de Software*. IN: FÓRUM DE EDUCAÇÃO EM ENGENHARIA DE SOFTWARE IN SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 2009, Fortaleza. Anais... Fortaleza: Sociedade Brasileira de Computação, 2009.
- BRIGGS, D. C. *Comments on Slavin: synthesizing causal inferences*. Educational Researcher, v. 37, n. 1, p. 15–22 37, jan./feb. 2008.

- BUSKIRK, R.; VAN MORONEY, B. W. Extending prototyping. *IBM Systems Journal*, v. 42, n. 4, 2003.
- CALVARY, G.; COUTAZ, J.; THEVENIN. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computer*, v. 15, n. 3, p. 289-308, 2003.
- CASTRO, J. F. B.; GIMENES, I. M. S.; MALDONADO, J. C. Uma proposta de plano pedagógico. In: CURSO DE QUALIDADE DE CURSOS DE GRADUAÇÃO DA ÁREA DE COMPUTAÇÃO E INFORMÁTICA, 2., 2000, Curitiba. Anais... Curitiba: SBC, 2000. p.251-270.
- CONSTANTINE, L. L. Process agility and software usability: toward lightweight usage-centered design. [S.l.]: Constantine & Lockwood, Ltd., 1999.
- CONSTANTINE, L. L.; LOCKWOOD, L. A. D. Software for use: a practical guide to the models and methods of usage-centered design. [S.l.]: Addison-Wesley, 1999.
- COYETTE, A.; KIEFFER, S.; VANDERDONCKT, J. Multi-fidelity prototyping of user interfaces. In: INTERACT, 11., Rio de Janeiro. Anais... Rio de Janeiro: Springer-Verlag, 2007. p. 149-162.
- CYBIS, W. et al. Uma abordagem ergonômica para o desenvolvimento de sistemas interativos. Campinas: SBC, 1999. Ata de conferência.
- EASYPROCESS - Processo de Desenvolvimento de Software para o DSC, 2006. Disponível em: <<http://www.dsc.ufcg.edu.br/~yp/>>. Acesso em: 10 abr. 2009.
- ENGELBERG, D.; SEFFAH, A. A framework for rapid mid-fidelity prototyping of web sites. In: *IFIP 17th Computer Congress: Usability*.

- Gaining a Competitive Edge Montréal. Québec. Anais... Canadá: Springer, 2002. p. 203-215.
- FERRÉ, X. Incorporating usability into object oriented process. In: INTERACT, 13., 2001, Japan. Anais... Japan:IOS Press. 2001.
- GOMES, A. et al. O ensino de interação humano-computador com ênfase na formação de competências relacionadas à inovação tecnológica: design da interação. In: WORKSHOP DE EDUCAÇÃO EM INFORMÁTICA (WEI), 17., Rio de Janeiro. Anais... Rio de Janeiro: CONGRESSO DA SBC, 2007. p. 125-134.
- GUERRERO, C. V. S. MEDITE - Uma Metodologia Orientada a Modelos para Concepção de Interfaces Ergonômicas. 2002. 79f. Dissertação (Mestrado) - Centro de Ciências e Tecnologia, Universidade Federal da Paraíba, Campina Grande, 2002.
- HAZZAN, O.; DUBINSKY, Y. Teaching a software development methodology: the case of. In: International Conference On Software Engineering, 16., Espanha. Anais... Espanha: IEEE Computer Society, 2003. p. 176-184.
- HISLOP, G. W. Software Engineering Education: Past, Present and Future. In: ELLIS, H. J. C.; DEMURJIAN, S. A.; NAVEDA, J. F. Software engineering: effective teaching and learning approaches and practices. [S.l.]: IGI Global, 2009.
- JACOBSON, I.; BOOCH, G.; RAMBAUGH, J. The unified software development process. [S.l.]: Addison-Wesley, 1999.
- JOHNSON, P. et al. Task-Related knowledge structures: Analysis. Modelling and Application. [S.n]: University of London, 1988.

- JOHNSON, P.; JOHNSON, H.; WILSON, S. Rapid prototyping of user interfaces driven by task models. In: CARROL, J. M. (ed.). Scenario-Based Design: envisioning work and technology in system development. New York: John Wiley and Sons, 1995. p.209-246.
- KITCHENHAM, B. et al. Preliminary guidelines for empirical research in. IEEE, n. 28, p. 72-734, 2002.
- LEONE, P.; GILLIHAN, D. . A. R. T. Web-based prototyping for user sessions: medium-fidelity prototyping. In: Proceedings of the Society for Technical Communications 44th Annual Conference. 44., 2000, Toronto. Anais... Toronto: STC, 2000. p. 231-234.
- LETHBRIDGE, T. C. et al. Improving software practice through education: Challenges and future trends. In: Future of Software Engineering (FOSE '07). Servilha. Espanha: IEEE Computer Society, 2007.
- LEVINE, D. M. B. M. L.; STEPHAN, D. Estatística: teoria e aplicações - Usando o Microsoft Excel Português. [S.l.]: LTC, 2000.
- LIMA, G. A.; AGUIAR, Y. P.; LULA, B. Impacto do apoio metodológico e ferramental a aspectos de usabilidade no ensino prático da engenharia de software. In: WORKSHOP DE EDUCAÇÃO EM INFORMÁTICA (WEI), 24., Bento Gonçalves. Anais... . Bento Gonçalves: Congresso da Sociedade Brasileira de Computação, 2009. p. 537-546.
- LU, S.; PARIS, C.; LINDEN, K. V. Integrating task modelling into object oriented design process: a pragmatic Approach. In: IFIP 7th – Working Conference on Engineering for Human-Computer Interaction, Grécia. Grécia: IFIP, 1998.

- MEDEIROS, F. P. A.; LULA, B. J.; CORDEIRO, P. B. A graphical tool to support task description using taos formalism for ui design. In: ERCIM WORKSHOP, 7., 2002, Paris. Anais... Paris: [s.n], 2002. p. 45-51.
- MEDEIROS, H.; KAFURE, I.; LULA, B. J. TAOS: a task-and-action oriented framewok for user´s task analysis in the context of human-computer interfaces design. In: INTERNATIONAL CONFERENCE OF THE CHIELAN COMPUTER SCIENCE SOCIETY, 20., 2000, Santiago del Chile. Anais... Santiago del Chile: [s.n.], 2000.
- MOFFAT, K. et al. Participatory Design with Aphasic Individuals. Canada: [s.n.],2003.
- NAVARRO, E. O.; HOEK, V. D. A. On the role of learning theories in furthering software enginnering education. In: ELLIS, H. J. C.;DEMURJIAN, S. A.; NAVEDA, J. F. Software engineering: effective teaching and learning approaches and practices (Hardcover). [S.l.]: IGI GLOBAL, 2009.
- NIELSEN, J; BUSH, R. M.; DAYTON, T.; MOND, N. E.; MULLER; M. J.; ROOT, R. W. Teaching experienced developers to design GUI. In: CHI '92 Proceedings of the SIGCHI conference on Human factors in computing systems. Monterrey, CA, USA: ACM, 1992. p.557-570.
- OLIVEIRA, K. M. A.; AGUIAR, Y. P. C.; LULA, B. O uso de modelos e múltiplos protótipos na concepção de interface do usuário. Revista Princípia, João Pessoa, v. 15, n. 11, p. 15-29, 2007.
- QUEIROZ, E. R. D. Abordagem híbrida pra a avaliação da usabilidade de interfaces com o usuário. Campina Grande, 2001. 410p. Tese (Doutorado em Engenharia Elétrica) - Centro de Ciências Exatas e Naturais, Universidade Federal da Paraíba, Campina Grande, 2001.

- RETTIG, M. Prototyping for tiny fingers. *Communications Of The ACM*, v. 37, n. 4, 1994.
- RODRIGUES, C. E. C. L.; LULA, B.; SUÁREZ, P. R. Using a script model to preserve the consistency within an ui design enviroment. In: *INTERNATIONAL WORKSHOP ON TASK MODELS AND DIAGRAMS*, 4., 2005, Poland. Anais... Poland: ACM, 2005.
- RUBIN, J. *Handbook of usability testing: how to plan, design, and conduct effective tests*. New York: John Wiley & Sons, 1994.
- RUDD, J.; STERN, K.; INSENSEE, S. L. High-fidelity prototyping debate. *Interactions*, v.3, n. 1, jan. 1996.
- SBC. *Grandes Desafios da Pesquisa em Computação no Brasil - 2006 - 2016*. 2006. Disponível em: <<http://www.sbc.org.br/index.php?language=1&content=downloads&id=272>>. Acesso em: 01 mar. 2009.
- SHNEIDERMAN, B. *The New Computing. Ubiquity*, set. 2001. Disponível em: <<http://ubiquity.acm.org/article.cfm?id=763933>>. Acesso em: 10 jan. 2011.
- SILVA, A. C. da ; ANACLETO, J. C.; PENTEADO, R. A. D.; SILVA, S. R. Integrando A Visão de ES e a da IHC através da Aplicação de Padrões sobre o Modelo de Prototipação Descartável. In: *CLIHIC - Conferência Latino-Americana de Interação Humano-Computador*, 2005, Cuernavaca - México. Anais... México: ACM. 2005. p. 223 - 234.
- SILVA, M. A. G. et al. Novas tendências em educação de engenharia de software: um estudo de caso no domínio de Teste de Software. In: *Fórum de Ensino de Engenharia de Software*, 2008, Campinas. Anais... Campinas: Simpósio Brasileiro de Engenharia de Software, 2008. p. 22 -30.

SOARES, M. S. Uma experiência de ensino de engenharia de software orientada. In: Fórum de Educação em Engenharia de Software, 2008, Campinas. Anais... Campinas: SBC, 2008.

SOLINGEN, R.; BERGHOUT, E. The Goal/Question/Metric Method: a practical guide for quality improvement of software development. UK: The McGraw-Hill Companies, 1999.

SUÁREZ, P. R.; LULA, B. Applying knowledge management in ui desing process. In: TAMODIA '04 Proceedings of the 3rd annual conference on Task models and diagrams, 2004, Prague, Czech Republic. Anais... Prague, Czech Republic: ACM, 2004.

Travassos, G. H.; Gurov, D. & do Amaral, E. A. G. (2002). Introdução à Engenharia de Software Experimental. Technical Report RT - ES 590/02, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ, Rio de Janeiro/RJ..

Vasconcelos, C. R.; Garcia, F. P. & Turnell, M. F. Q. V. (2003). Integrando Usabilidade e Engenharia de Software: um modelo para o desenvolvimento de sistemas centrado no usuário. In WIHC-ES2003 - Integrating Human-Computer Interaction and Software Engineering Models and Processes, Rio de Janeiro/RJ. I Latin American Conference on Human-Computer Interaction - CLIHC 2003.

_____. XPU: Um Modelo para Desenvolvimento de Sistemas Centrado no Usuário. In: WORKSHOP DE TESES EM QUALIDADE DE SOFTWARE, 1., 2003, Fortaleza. Anais... Fortaleza: SBC, 2003.

WANGENHEIM, C. G.; SILVA, D. A. Qual conhecimento de engenharia de software é importante para um Profissional de Software?. In: FÓRUM DE EDUCAÇÃO EM ENGENHARIA DE SOFTWARE. In: XXIII

Simpósio Brasileiro de Engenharia de Software, 2009, Fortaleza.
Anais... Fortaleza: SBC, 2009. p. 1 – 8.

APÊNDICE A – TESTE T DE DUAS AMOSTRAS PRESUMINDO VARIÂNCIAS DIFERENTES PARA VARIÁVEIS QUANTITATIVAS

Este apêndice sumariza os resultados na administração do teste t de duas amostras presumindo variâncias diferentes para as variáveis quantitativas.

A.1. Quantidade de Tarefas (QT)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	19,41666667	39,75
Variância	73,17424242	691,1136364
Observações	12	12
Hipótese da diferença de média	0	
gl	13	
Stat t	-2,547831586	
P(T<=t) uni-caudal	0,012146023	
t crítico uni-caudal	2,160368652	
P(T<=t) bi-caudal	0,024292046	
t crítico bi-caudal	2,532637811	

REJEITAR HIP NULA

A.2. Taxa de Realização das Tarefas (TRT)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	0,447	0,369583333
Variância	0,055177455	0,015615356
Observações	12	12
Hipótese da diferença de média	0	
gl	17	
Stat t	1,007930329	
P(T<=t) uni-caudal	0,16380994	
t crítico uni-caudal	2,109815559	
P(T<=t) bi-caudal	0,327619881	
t crítico bi-caudal	2,458050719	

NÃO REJEITAR HIP NULA

A.3. Uso de Operadores (UO): Operador SEQ

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	4,833333333	7
Variância	38,51515152	30,72727
Observações	12	12
Hipótese da diferença de média	0	
gl	22	
Stat t	-0,901979345	
P(T<=t) uni-caudal	0,188418668	
t crítico uni-caudal	2,073873058	
P(T<=t) bi-caudal	0,376837336	
t crítico bi-caudal	2,40547274	

NÃO REJEITAR HIP NULA

A.4. Uso de Operadores (UO): Operador OR

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	2,416666667	2,333333333
Variância	13,17424242	13,15151515
Observações	12	12
Hipótese da diferença de média	0	
gl	22	
Stat t	0,056262489	
P(T<=t) uni-caudal	0,47782028	
t crítico uni-caudal	2,073873058	
P(T<=t) bi-caudal	0,95564056	
t crítico bi-caudal	2,40547274	

NÃO REJEITAR HIP NULA

A.5. Uso de Operadores (UO): Operador XOR

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	4,166666667	3,666666667
Variância	8,878787879	9,333333333
Observações	12	12
Hipótese da diferença de média	0	
gl	22	
Stat t	0,405863841	
P(T<=t) uni-caudal	0,344381549	
t crítico uni-caudal	2,073873058	
P(T<=t) bi-caudal	0,688763099	
t crítico bi-caudal	2,40547274	

NÃO REJEITAR HIP NULA

A.6. Uso de Operadores (UO): Operador AND

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	2,583333333	3,416666667
Variância	55,17424242	72,26515152
Observações	12	12
Hipótese da diferença de média	0	
gl	22	
Stat t	-0,25571578	
P(T<=t) uni-caudal	0,400273011	
t crítico uni-caudal	2,073873058	
P(T<=t) bi-caudal	0,800546022	
t crítico bi-caudal	2,40547274	

NÃO REJEITAR HIP NULA

A.7. Tamanho do Protótipo (TP)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	2,083333333	4,916666667
Variância	2,083333333	12,26515152
Observações	12	12
Hipótese da diferença de média	0	
gl	15	
Stat t	-2,591106551	
P(T<=t) uni-caudal	0,010228933	
t crítico uni-caudal	2,131449536	
P(T<=t) bi-caudal	0,020457867	
t crítico bi-caudal	2,489879694	

REJEITAR HIP NULA

A.8. Taxa de elementos de interação no protótipo (TEIP)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	3,625916667	6,597833333
Variância	5,923369174	11,67248161
Observações	12	12
Hipótese da diferença de média	0	
gl	20	
Stat t	-2,454268738	
P(T<=t) uni-caudal	0,0117022	
t crítico uni-caudal	2,085963441	
P(T<=t) bi-caudal	0,0234044	
t crítico bi-caudal	2,423116536	

REJEITAR HIP NULA

A.9. Nível de especificação do projeto funcional (NEPF)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	2,988166667	2,838083333
Variância	2,210118515	4,378075902
Observações	12	12
Hipótese da diferença de média	0	
gl	20	
Stat t	0,202553563	
P(T<=t) uni-caudal	0,42076516	
t crítico uni-caudal	2,085963441	
P(T<=t) bi-caudal	0,84153032	
t crítico bi-caudal	2,423116536	

NÃO REJEITAR HIP NULA

A.10. Nível de realização do projeto (NRP)

	<i>Grupo de Controle</i>	<i>Grupo Experimental</i>
Média	1,584333333	1,392166667
Variância	2,38353497	0,927274333
Observações	12	12
Hipótese da diferença de média	0	
gl	18	
Stat t	0,365848768	
P(T<=t) uni-caudal	0,359371085	
t crítico uni-caudal	2,100922037	
P(T<=t) bi-caudal	0,718742171	
t crítico bi-caudal	2,445005614	

NÃO REJEITAR HIP NULA

ANEXO A – PESQUISA SOBRE USO DE TÉCNICAS DE IHC NO ENSINO DA ENGENHARIA DE SOFTWARE

- **Objetivo:** levantar fatos e crenças sobre a aplicabilidade do uso de técnicas de IHC dentro do ensino da Engenharia de Software.
- **Contexto:** instituições relevantes em âmbito regional (NE)
- **Período:** junho à agosto de 2010
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs)
- **Amostra:** 30 professores de cursos de computação

Conteúdo e Sumário de Resultados

1. Indique abaixo o nome da instituição de ensino na qual você atua como docente.

Unifor, IFPB , UFCG, UFPB, UEPB, IESP, IFCE , UFPI, UFPE, Favip-PE, Unibratéc-JP, UFRN.

2. Você ultimamente tem se dedicado ao ensino em quais áreas?

IHC 26%

Engenharia de Software 67%

Outras 19%

3. Qual sua experiência no emprego de metodologias/processos de desenvolvimento CENTRADOS NO USUÁRIO?

Bastante 11%

Muita 33%

Nem muita, nem pouca 19%

Pouca 19%

Muito pouca 15%

Não sei do que se trata 4%

4. Dentre as metodologias/processos de desenvolvimento de software trabalhadas por você, algum processo de desenvolvimento CENTRADO NO USUÁRIO é aplicado num contexto prático (laboratório de software)?

Sim 44%

Não 48%

Não acho relevante 4%

Não sei 4%

5. Em alguma disciplina (sua ou de outros colegas) os alunos tem a oportunidade de desenvolver um projeto de software por completo atendendo a demandas de um cliente real?

Sim 56%

Não 41%

Não acho relevante 0%

Não sei 4%

6. Conceitos de IHC em seu curso são abordados de que forma? (marque todas as opções aplicáveis)

Dentro de uma disciplina de Engenharia de Software 37%

Em uma disciplina exclusiva de IHC 52%

De forma optativa 37%

De forma obrigatória 22%

7. Existe alguma disciplina que permita a aplicação pratica de conceitos de Engenharia de Software, que simula o desenvolvimento de um produto de software, por completo, em seu curso?

Sim 63%

Não 37%

8. Você acha importante que conceitos de projeto de interfaces/usabilidade sejam abordados de forma contextualizada em disciplinas de Engenharia de Software?

Concordo Totalmente 81%

Concordo 15%

Nem concordo, nem discordo 0%

Discordo 4%

Discordo Totalmente 0%

9. Dentro do contexto de desenvolvimento centrado no usuário, você acredita que os alunos geralmente tendem a apreciar mais artefatos funcionais, depreciando o valor dos artefatos relacionados ao projeto de interface/usabilidade?

Concordo Totalmente 26%

Concordo 59%

Nem concordo, nem discordo 4%

Discordo 11%

Discordo Totalmente 0%

10. Você acredita que os alunos saem com uma boa compreensão de técnicas de projeto de interface/usabilidade?

Concordo Totalmente 4%

Concordo 22%

Nem concordo, nem discordo 26%

Discordo 41%

Discordo Totalmente 7%

11. Aponte as razões que você acredita serem as mais influentes no sentido de possibilitar um ensino de Engenharia de Software que dê uma maior destaque a técnicas/métodos de projeto de interface/usabilidade

Cultural, muitos associam IHC como uma área opcional/especializada 70%

Complexidade, pois as técnicas e métodos de IHC se apresentam complexos inviabilizando qualquer abordagem, mesmo que superficial 19%

Escassez de tempo, pois outros assuntos acabam sendo mais priorizados, como aqueles relativos aos aspectos funcionais do software (gerência, planejamento, testes, arquitetura) 44%

Nenhuma 0%

12. Você acredita que questões relacionadas a fatores humanos são abordadas com que frequência no ENSINO de práticas de Engenharia de Software (de forma geral)?

Bastante satisfatória 0%

Muito satisfatória 7%

Nem muito, nem pouco satisfatória 19%

Deficiente 56%

Muito deficiente 19%

Não acho relevante 0%

13. Quais das técnicas/tópicos de IHC você já abordou ou tentou abordar em suas disciplinas?

Modelagem de tarefas 41%

Prototipagem 78%

Design iterativo/incremental/evolutivo 67%

Testes de usabilidade 37%

Nenhum 7%

Outros 4%

ANEXO B – QUESTIONÁRIOS ALUNOS

Questionário de Levantamento do Perfil (Alunos)

- **Título:** Conhecimento Prévio de Domínio, Afinidade e Confiança Inicial
- **Objetivo:** levantar características dos alunos
- **Contexto:** disciplina LES
- **Períodos:** semestres de 2008.2 à 2010.1
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs) ou aplicação direta.

1	Qual seu sexo?
	<input type="checkbox"/> Masculino <input type="checkbox"/> Feminino

2	Como você considera seu nível de experiência no desenvolvimento de sistemas computacionais?
	<input type="checkbox"/> Bastante experiente <input type="checkbox"/> Experiente <input type="checkbox"/> Nem experiente nem inexperiente <input type="checkbox"/> Inexperiente <input type="checkbox"/> Bastante Inexperiente

3	Você possui algum conhecimento acerca de metodologias de desenvolvimento (ex.: XP, RUP, YP) anterior ao curso de LES?
	<input type="checkbox"/> Sim <input type="checkbox"/> Não

4	Como você considera seu nível de experiência quanto à construção de interfaces em sistemas computacionais?
	<input type="checkbox"/> Bastante experiente <input type="checkbox"/> Experiente <input type="checkbox"/> Nem experiente nem inexperiente <input type="checkbox"/> Inexperiente <input type="checkbox"/> Bastante Inexperiente

5	Como você considera a importância do projeto de interfaces do usuário no desenvolvimento de produtos de software?
	<input type="checkbox"/> Essencial importante <input type="checkbox"/> Extremamente importante <input type="checkbox"/> Dispensável <input type="checkbox"/> Tanto faz

6	Como você considera seu nível de experiência na construção de protótipos de interface?			
	<input type="checkbox"/> Bastante experiente	<input type="checkbox"/> Experiente	<input type="checkbox"/> Nem experiente nem inexperiente	<input type="checkbox"/> Inexperiente <input type="checkbox"/> Bastante Inexperiente

7	Possui conhecimento prévio quanto à modelagem de tarefas do usuário?	
	<input type="checkbox"/> Sim	<input type="checkbox"/> Não

7	Você é "repetente" em LES?	
	<input type="checkbox"/> Sim	<input type="checkbox"/> Não

Questionário Pós Etapa de Planejamento (Alunos)

- **Objetivo:** levantar fatos e crenças sobre o manejo dos artefatos de usabilidade logo após o término da etapa da Planejamento de YP, em que os mesmos obrigatoriamente, seguindo indicação processo, devem estar disponíveis.
- **Contexto:** disciplina LES
- **Períodos:** semestres de 2008.2 à 2010.1
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs) ou aplicação direta.

Nome: _____

Equipe/Projeto: _____ Período: _____

MEDITE+FastInterface: sim não

1	Antes de realizar a etapa de planejamento do seu projeto você tinha alguma noção/conhecimento da complexidade do domínio do problema que o seu software está envolvido?	Importância (0-10)
	<input type="checkbox"/> Bastante conhecimento <input type="checkbox"/> Muito conhecimento <input type="checkbox"/> Nem muito nem pouco conhecimento <input type="checkbox"/> Pouco conhecimento <input type="checkbox"/> Nenhum conhecimento	

2	Reprodução dos requisitos obtidos na conversa com o cliente dentro do modelo de tarefas com os recursos que você utilizou foi:	Importância (0-10)
	<input type="checkbox"/> Muito fácil <input type="checkbox"/> Fácil <input type="checkbox"/> Nem fácil nem difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Muito difícil	

3	Compreensão do modelo de tarefas obtido pela sua equipe foi:	Importância (0-10)
	<input type="checkbox"/> Muito fácil <input type="checkbox"/> Fácil <input type="checkbox"/> Nem fácil nem difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Muito difícil	

4	Se você chegou a apresentar o modelo de tarefas parcial obtido a clientes/usuários, como foi o grau de esforço para fazer essa demonstração?	Importância (0-10)
	<input type="checkbox"/> Muito fácil <input type="checkbox"/> Fácil <input type="checkbox"/> Nem fácil nem difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Muito difícil <input type="checkbox"/> Não apresentei o modelo de tarefas ao cliente/usuário	

5	Acho que o cliente/usuário compreendeu suficientemente bem o modelo de tarefas obtido.					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

6	Acho que o modelo de tarefas auxiliou na compreensão das funcionalidades do sistema (na decomposição de suas funções).					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

7	Acho que o modelo de tarefas obtido contribuiu para uma melhor especificação e/ou compreensão de <i>user stories</i> e <i>testes de aceitação</i>.					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

8	Acho que a modelagem da tarefa, por ampliar a compreensão dos requisitos e a forma de utilização do sistema, contribui para uma especificação mais realista do planejamento de iterações e releases.					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

9	Como você se sente confiante em relação à qualidade do modelo de tarefas obtido?					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

10	Você acredita que o modelo de tarefas integra as discussões da equipe e auxilia na tomada de decisão sobre quais funcionalidades o software deverá conter?					Importância (0-10)
	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	—

11	A elaboração do protótipo de interface a partir do modelo de tarefa é:					Importância (0-10)
	<input type="checkbox"/> Muito fácil	<input type="checkbox"/> Fácil	<input type="checkbox"/> Nem fácil nem difícil	<input type="checkbox"/> Difícil	<input type="checkbox"/> Muito difícil	—

12	Indique o nível de esforço para construção do protótipo de interface com os recursos utilizados.	Importância (0-10)
	<input type="checkbox"/> Muito fácil <input type="checkbox"/> Fácil <input type="checkbox"/> Nem fácil nem difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Muito difícil	—

13	A compreensão da forma de utilização dos elementos da interface pelo cliente/usuário na apresentação do protótipo foi.	Importância (0-10)
	<input type="checkbox"/> Muito fácil <input type="checkbox"/> Fácil <input type="checkbox"/> Nem fácil nem difícil <input type="checkbox"/> Difícil <input type="checkbox"/> Muito difícil <input type="checkbox"/> Não se aplica	—

14	Fiquei satisfeito(a) com o protótipo de interface gerado	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	—

15	Acredito que houve influência do modelo de tarefas na concepção do protótipo	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	—

16	A obtenção do modelo de tarefas foi importante para a geração do protótipo	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	—

Questionário Pós Release Final (Alunos)

- **Objetivo:** levantar o conceito final dos alunos sobre o manejo com os artefatos de usabilidade e produto final.
- **Contexto:** disciplina LES
- **Períodos:** semestres de 2008.2 à 2010.1
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs) ou aplicação direta.

Nome: _____

Equipe/Projeto: _____ Período: _____

MEDITE+FastInterface: sim não

1	Durante o desenvolvimento do projeto quais dos artefatos de usabilidade abaixo chegaram a ser modificados/atualizados? (marque todas as opções aplicáveis)	Importância (0-10)
	<input type="checkbox"/> Perfil do usuário <input type="checkbox"/> Objetivos de usabilidade <input type="checkbox"/> Modelo de tarefas <input type="checkbox"/> Protótipo de interface	

2	Em algum momento os o escopo de funcionalidades mudou durante o desenvolvimento do projeto? Indique o grau dessas modificações.	Importância (0-10)
	<input type="checkbox"/> Bastante <input type="checkbox"/> Muito <input type="checkbox"/> Nem muito, nem pouco <input type="checkbox"/> Pouco <input type="checkbox"/> Muito pouco	

3	Dos artefatos de usabilidade abaixo, quais foram desenvolvidos em conjunto, com opiniões da maioria da equipe?	Importância (0-10)
	<input type="checkbox"/> Perfil do usuário <input type="checkbox"/> Objetivos de usabilidade <input type="checkbox"/> Modelo de tarefas <input type="checkbox"/> Protótipo de interface	

4	O modelo de tarefas auxilia na compreensão do "jeito de usar" um sistema	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	

5	O modelo de tarefas pode ser utilizado em testes de usabilidade	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
6	O modelo de tarefas e o protótipo permitiram uma melhor comunicação entre os membros da equipe (internamente) sobre o projeto do software	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
7	O modelo de tarefas auxiliou na compreensão do domínio do problema do software, ou seja na extração de suas funcionalidades	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
8	O modelo de tarefas permite uma melhor comunicação entre a equipe e o cliente	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
9	O modelo de tarefas permite detectar melhor quais funcionalidades são mais complexas de desenvolver	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
10	Quais dos artefatos de usabilidade abaixo você continuaria a utilizar ou recomendaria em projetos de software futuros? (marque todas as opções aplicáveis)	Importância (0-10)
	<input type="checkbox"/> Perfil do usuário <input type="checkbox"/> Objetivos de usabilidade <input type="checkbox"/> Modelo de tarefas <input type="checkbox"/> Protótipo de interface	
11	Se fosse pedido a você desenvolver o modelo de tarefas de um sistema você se sentiria habilitado considerando seu atual grau de compreensão?	Importância (0-10)
	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	

	O software desenvolvido está em boas condições de uso para os clientes/usuários?					Importância (0-10)
12	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	

	Os artefatos de usabilidade auxiliaram no processo de implementação do software.					Importância (0-10)
13	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	

	Você acredita que o cliente ficou satisfeito com o seu projeto sob quais aspectos? (marque todas as alternativas aplicáveis)					Importância (0-10)
14	<input type="checkbox"/> Testes	<input type="checkbox"/> Comunicação	<input type="checkbox"/> Interface	<input type="checkbox"/> Tempo de Desenvolvimento	<input type="checkbox"/> Cumprimento das expectativas sobre o projeto	

ANEXO C – QUESTIONÁRIOS

CLIENTES/USUÁRIOS

Questionário Release Intermediário (Clientes/Usuários)

- **Objetivo:** receptividade, compreensão e satisfação dos clientes com os artefatos de usabilidade e equipe de desenvolvimento.
- **Contexto:** disciplina LES, após o release intermediário em YP
- **Períodos:** semestres de 2008.2 à 2010.1
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs)

Você foi cliente de qual equipe: _____

O que você achou do modelo de tarefas da interface de seu projeto?					Importância (0-10)	
1	<input type="checkbox"/> Muito compreensível	<input type="checkbox"/> Compreensível	<input type="checkbox"/> Nem muito, nem pouco compreensível	<input type="checkbox"/> Pouco compreensível	<input type="checkbox"/> Incompreensível	
					<input type="checkbox"/> Não se aplica	

Você acredita que o modelo de tarefas auxiliou para uma melhor compreensão de funcionalidades de seu projeto?					Importância (0-10)	
2	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	
					<input type="checkbox"/> Não se aplica	

O protótipo da interface que lhe foi apresentado, por sua equipe, atendeu às suas expectativas?					Importância (0-10)	
3	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	
					<input type="checkbox"/> Não se aplica	

	Você acredita que esta equipe conseguirá atender às suas expectativas ao término do projeto?	Importância (0-10)
4	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
	Você acredita que a equipe compreende às suas demandas e expectativas, de forma geral, de que maneira?	Importância (0-10)
5	<input type="checkbox"/> Muito satisfatória <input type="checkbox"/> Satisfatória <input type="checkbox"/> Nem satisfatória, nem insatisfatória <input type="checkbox"/> Insatisfatória <input type="checkbox"/> Muito insatisfatória <input type="checkbox"/> Não se aplica	
	Você já visualizou alguma versão parcial de seu projeto? Em caso positivo qual sua avaliação em relação ao atendimento às suas expectativas?	Importância (0-10)
6	<input type="checkbox"/> Muito satisfatória <input type="checkbox"/> Satisfatória <input type="checkbox"/> Nem satisfatória, nem insatisfatória <input type="checkbox"/> Insatisfatória <input type="checkbox"/> Muito insatisfatória <input type="checkbox"/> Não se aplica	
	Você achou interessante e útil o uso do modelo de tarefas e do protótipo de interfaces em seu projeto?	Importância (0-10)
7	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
	O protótipo de interfaces permitiu uma melhor negociação com a equipe sobre a aparência final do produto?	Importância (0-10)
8	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	
	Você acredita que o desenvolvimento da INTERFACE do software se encontra:	Importância (0-10)
9	<input type="checkbox"/> Muito Adiantado <input type="checkbox"/> Adiantado <input type="checkbox"/> Normal <input type="checkbox"/> Atrasado <input type="checkbox"/> Muito Atrasado	

	A equipe tem divergido quanto às suas solicitações					Importância (0-10)
10	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	
	A equipe tem apresentado dificuldades em compreender o que necessito no projeto					Importância (0-10)
11	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	
	A equipe consegue obter boas soluções quando há divergências/problemas no desenvolvimento de funcionalidades do software, especialmente aquelas funcionalidades que se apresentam mais complexas					Importância (0-10)
12	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo	<input type="checkbox"/> Discordo totalmente	
	Selecione os artefatos abaixo que foram atualizados com sua aprovação					Importância (0-10)
13	<input type="checkbox"/> Protótipo de interface	<input type="checkbox"/> Modelo de tarefas	<input type="checkbox"/> User stories e testes de aceitação	<input type="checkbox"/> Nenhum		

Questionário Pós Release Final (Clientes/Usuários)

- **Objetivo:** receptividade, compreensão e satisfação finais dos clientes com os artefatos de usabilidade, a equipe de desenvolvimento e o produto final.
- **Contexto:** disciplina LES
- **Períodos:** semestres de 2008.2 à 2010.1
- **Meio de distribuição/artefatos:** formulário eletrônico (GoogleDocs)

Você foi cliente de qual equipe: _____

Indique o grau de satisfação com a interface de seu produto final		Importância (0-10)
1	<input type="checkbox"/> Muito satisfeito <input type="checkbox"/> Satisfeito <input type="checkbox"/> Nem satisfeito, nem insatisfeito <input type="checkbox"/> Insatisfeito <input type="checkbox"/> Muito insatisfeito	

Indique o grau de satisfação com a sua equipe		Importância (0-10)
2	<input type="checkbox"/> Muito satisfeito <input type="checkbox"/> Satisfeito <input type="checkbox"/> Nem satisfeito, nem insatisfeito <input type="checkbox"/> Insatisfeito <input type="checkbox"/> Muito insatisfeito	

Indique o grau de satisfação com o nível de realização do projeto.		Importância (0-10)
3	<input type="checkbox"/> Muito satisfeito <input type="checkbox"/> Satisfeito <input type="checkbox"/> Nem satisfeito, nem insatisfeito <input type="checkbox"/> Insatisfeito <input type="checkbox"/> Muito insatisfeito	

Nos próximos projetos de software que eu encomendar farei questão pelo uso e apresentação do modelo de tarefas		Importância (0-10)
4	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	

Nos próximos projetos de software que eu encomendar farei questão pelo uso e apresentação de protótipos de interface		Importância (0-10)
5	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	

Você acha que a interface de seu produto é fácil de aprender e usar?					Importância (0-10)
6	<input type="checkbox"/> Concordo totalmente	<input type="checkbox"/> Concordo	<input type="checkbox"/> Nem concordo, nem discordo	<input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente	

ANEXO D – TUTORIAL *FASTINTERFACE*

- **Objetivo:** orientar o uso da ferramenta *FastInterface* para elaboração dos artefatos modelo de tarefa e protótipo de interface.
- **Contexto:** equipes inseridas no grupo de tratamento.
- **Período:** antes da geração do modelo de tarefas e do protótipo e da realização de qualquer artefato funcional inserido na fase de Planejamento de YP.
- **Meio de distribuição:** eletrônica (kit tutorial+ferramenta de software) disponível através do site <http://les.dsc.ufcg.edu.br/fastinterface>

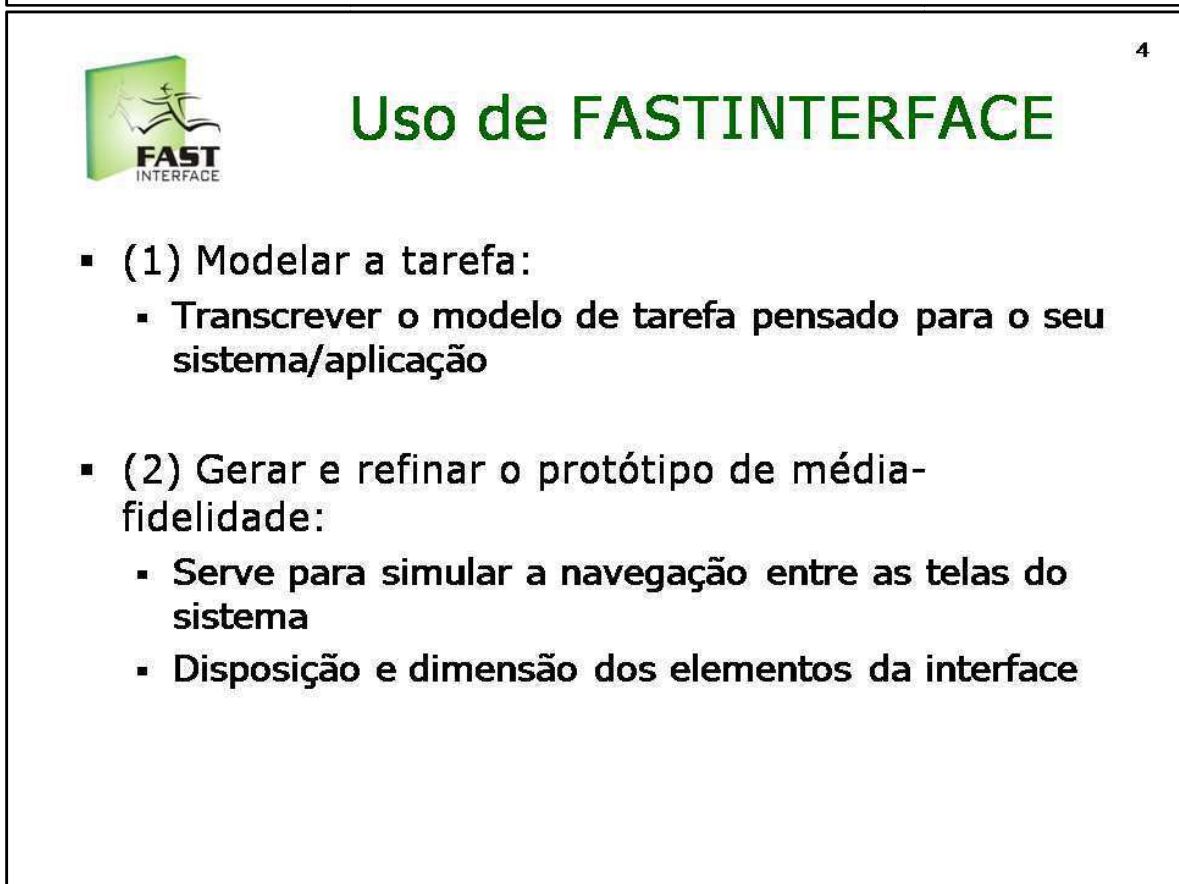
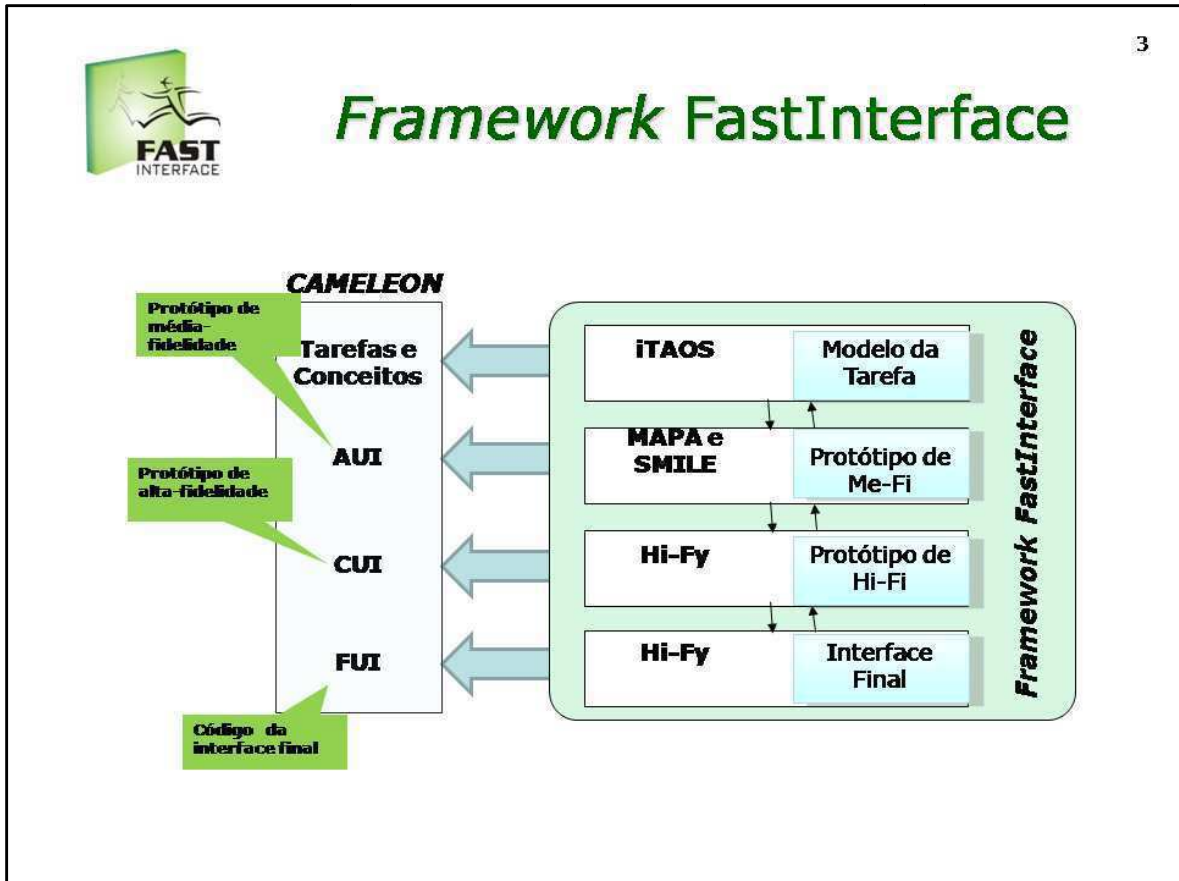


Giuseppe Anthony N. Lima
giuseppe@dsc.ufcg.edu.br



FastInterface

- Ambiente de Desenvolvimento Baseado em Modelos e Múltipla Prototipagem Evolutiva
- Disponível em:
fi.ifpb.edu.br





Uso de FASTINTERFACE

5

- (3) Gerar e refinar o protótipo de alta-fidelidade
 - Atributos estéticos, fontes, cores dos elementos da interface
 - Gerar código da interface final para posterior acoplamento c/ as funcionalidades

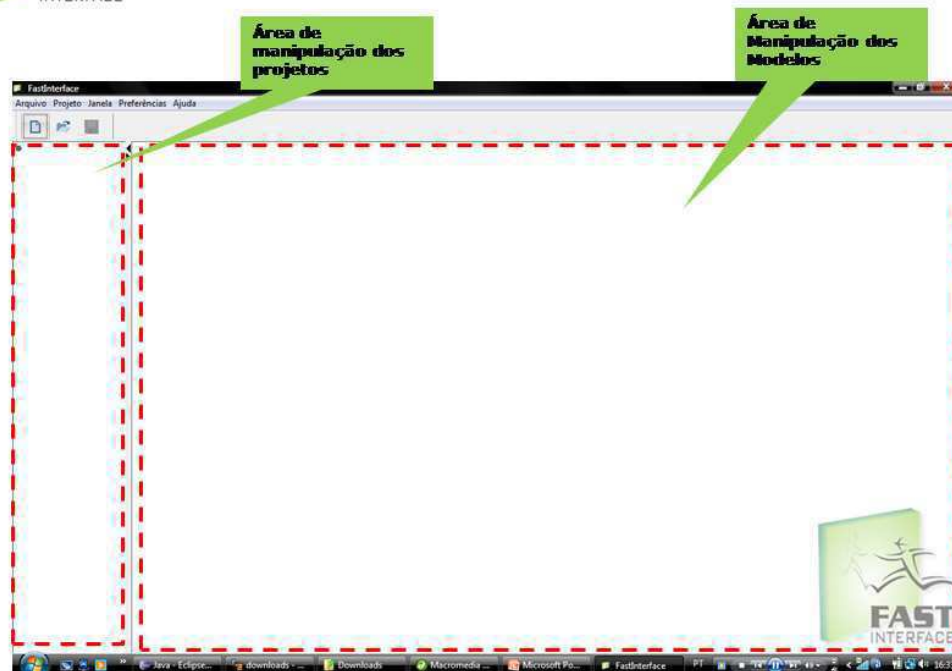
FASTINTERFACE facilita a geração da interface para facilitar o processo de transcrição do modelo de tarefa em um protótipo de interface, com suas telas e elementos de interação (áreas, botões, objetos de seleção, etc...)

A interface do produto final pode se basear ou não no protótipo gerado: ainda que o projetista opte por não utilizar o código gerado para o acoplamento com as funcionalidades do sistema final, o modelo de navegação e a idealização da interface são antecipados e obtidos sem muito esforço.



Uso de FASTINTERFACE

6



7



Uso de FASTINTERFACE

Criando um novo projeto

É possível criar um novo projeto clicando aqui...





8



Uso de FASTINTERFACE

Criando um novo projeto

Ou acessando via menu...



Esse formulário surgirá. Informe os dados de seu projeto nele. O botão procurar indique aonde o mesmo deve ser salvo.





Uso de FASTINTERFACE Modelo da Tarefa

9

Clique 2x aqui para começar a produzir...

Em seguida clique 2x aqui p/ produzir o modelo de tarefa...

Clique c/ o botão direito encima da tarefa (caixa amarela) p/ adicionar sub-tarefas...

Clique c/ o botão direito encima da tarefa (caixa amarela) p/ adicionar sub-tarefas...



Uso de FASTINTERFACE Modelo da Tarefa

10

Se seu modelo de tarefa ficar um pouco grande ative o Overview

Salve seu modelo de tarefa aqui

Passe o mouse encima p/ visualizar o nome completo da tarefa

Operadores de 2º nível podem ser adicionados entre tarefas irmãs desde que o operador de 1º nível seja seq ou and e possua 3 filhos ou mais

zoom

O modelo de tarefa é apresentado num formato menor (Overview). Utilize o quadrado vermelho (segurar e arrastar) para editar/visualizar a área que desejar



Uso de FASTINTERFACE Modelo da Tarefa

11

Operadores

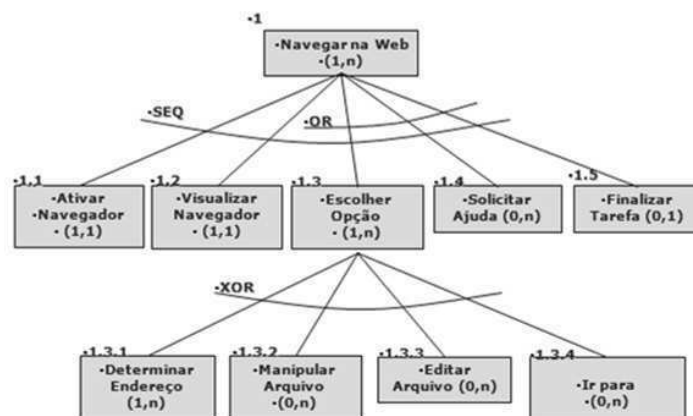
SEQ	Execução seqüencial de sub-tarefas e/ou ações
OR	Execução de pelo menos uma das sub-tarefas e/ou ações
XOR	Execução de apenas uma das sub-tarefas e/ou ações
AND	Execução de todas as sub-tarefas e/ou ações, independentemente da ordem
SIM	Execução simultânea e independente de sub-tarefas e/ou ações
PAR	Execução concorrente (i.e. com pontos de sincronização) de sub-tarefas e/ou ações



Uso de FASTINTERFACE Modelo da Tarefa

12

Exemplo de Modelo de Tarefa sobre TAOS:





Uso de FASTINTERFACE Protótipo de Média-Fidelidade

13

The screenshot shows the FASTINTERFACE application window. On the left, a tree view displays a hierarchy of spaces: 'space1.1' (Access Sistema) branches into 'space1.1.1' (Estatue Login) and 'space1.1.2' (Estatue Avaliação). 'space1.1.1' further branches into 'space1.1.1.1' (Sobre Empresa) and 'space1.1.1.1.1' (Sobre Empresa). 'space1.1.2' branches into 'space1.1.2.1' (Nova Avaliação) and 'space1.1.2.2' (Continuar Avaliação). On the right, a wireframe of a screen titled 'Consultor' is shown, containing buttons for 'Acessar Sistema', 'Calcular Empresa', 'Ajuda', and 'Fechar Sistema'. Below these are several 'Objeto de interação' (interaction objects) and a 'Visão' (view) area.

Annotations in green callouts point to various elements:

- Tela ou espaço**: Points to the top right area of the wireframe.
- Visão**: Points to the 'Visão' label in the wireframe.
- Objeto de interação**: Points to three different interaction objects in the wireframe.
- Visão**: Points to the 'Visão' label in the wireframe.
- Objeto de interação**: Points to another interaction object in the wireframe.
- Área ou modo Árvore/Tree. Contém a estrutura das telas/espacos do seu sistema.**: Points to the tree view on the left.
- Clique 2x para ativas a aba de protótipo de média fidelidade.**: Points to the 'Protótipo Me-F' tab in the top menu.
- ZOOM**: Points to the zoom slider at the bottom of the tree view.



Uso de FASTINTERFACE Protótipo de Média-Fidelidade

14

- **Espaços**
 - São as telas da sua interface
- **Visões**
 - São áreas da tela que agrupam objetos de interação de um contexto comum
- **Objetos de interação**
 - São os elementos da interface de fato: imagem, botão, caixa de seleção, rótulo, etc...



Uso de FASTINTERFACE Protótipo de Média-Fidelidade

15

- **ESPAÇO**
 - **Inserir/Remover:** clique c/ o botão direito encima do espaço (um nodo) da árvore de telas na área *Árvore/Tree*.
- **VISÕES**
 - **Inserir:** clique no botão "Inserir visão no Espaço"
 - **Remover:** Clique c/ o botão direito encima da visão e escolha remover (Modo Esboço/Sketch)
- **OBJETOS DE INTERAÇÃO**
 - **Inserir:** utilize a barra de botões referente aos objetos de interação
 - **Remover:** clique c/ o botão direito encima da visão na qual deseja inserir um objeto de interação.
 - **Modificar:** clique c/ o botão direito encima do objeto de interação que deseja modificar. (ex.: transformar uma radiobutton em um checkbox)



Uso de FASTINTERFACE Protótipo de Média-Fidelidade

16

The screenshot shows the FASTINTERFACE software interface with several callout boxes explaining the functions of various toolbars and buttons:

- Inserir uma visão em um espaço/tela.** (points to the 'Inserir Visão' button in the top toolbar)
- Navegue pela ordem entre os esboços.** (points to the navigation buttons in the top toolbar)
- Modo Sketch/Esboço Estendido** (points to the 'Modo Esboço' button in the top toolbar)
- Salve seu protótipo me-fi aqui.** (points to the 'Salvar' button in the top toolbar)
- Exporte como imagem seus protótipos aqui.** (points to the 'Exportar Imagem' button in the top toolbar)
- Simule/Pare a navegação do seu protótipo aqui.** (points to the 'Simular' and 'Parar' buttons in the top toolbar)
- Inserir objetos de interação habilitados p/ uma visão.** (points to the interaction object buttons in the top toolbar)
- Exporte como imagem seus protótipos** (points to the 'Exportar Imagem' button in the top toolbar)
- zoom** (points to the zoom slider in the bottom toolbar)
- Visões e alguns objetos de interação são dimensionáveis, refine seu protótipo me-fi como desejar.** (points to the main workspace)
- A visão funcionalidade/functionality só pode ser editada através da tela/espaco inicial.** (points to the main workspace)



Uso de FASTINTERFACE Protótipo de Alta-Fidelidade

17

Navegue pela ordem entre as telas

Aba do protótipo de alta-fidelidade (hi-fi)

Clique 2x para ativar a aba de edição do protótipo de alta-fidelidade

Área ou modo Árvore/Tree. Contém a estrutura das telas/espacos do seu sistema.

Edição da disposição dos elementos não é possível! (deve ser feito no me-fi)

O foco nesta fase é a especificação dos atributos estéticos (cor, fonte, etc.)



Edição do Protótipo de Alta Fidelidade (Hi-fi)

18

Toolbar de Edição do Modelo CUI

Exportar Código

Se o botão remeter a outra tela clique 2x e irá diretamente p/ a tela

Botão direito abre a edição do elemento

Caixa de edição aberta



Resumo das Principais Funções de FASTINTERFACE

- Dar suporte à análise e modelagem da tarefa;
- Implementar o mapeamento entre os elementos do modelo da tarefa (MT) e os elementos do modelo da interação (interface abstrata - AUI);
- Permitir a edição e simulação do modelo da interface abstrata a partir da edição e simulação de protótipos de média fidelidade;
- Dar suporte à geração e edição de protótipos de alta fidelidade da interface;
- Geração do código da interface final;



Dúvidas?
giuseppe@dsc.ufcg.edu.br

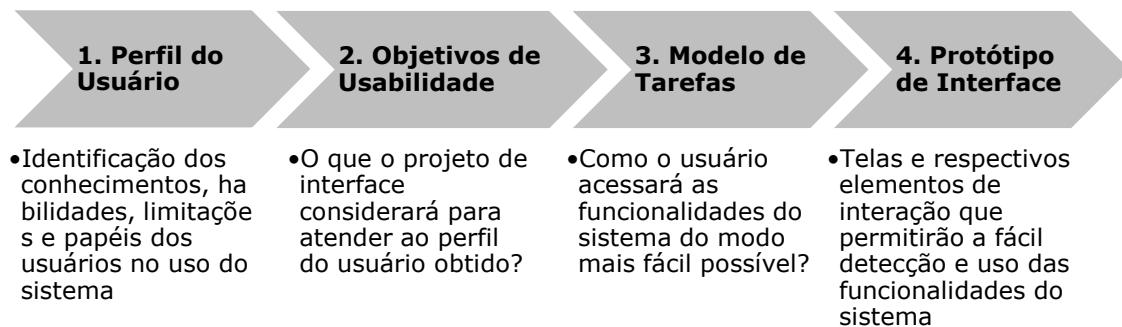
ANEXO E – GUIA PARA ALUNOS: MANIPULAÇÃO DE ARTEFATOS DE USABILIDADE EM YP

Apresentação

Este guia tem como principal objetivo orientar os alunos do laboratório de software no que se refere à produção dos artefatos de usabilidade indicados pelo processo YP, quando desenvolvidos sob os preceitos da metodologia de concepção de interfaces MEDITE, com o uso do suporte ferramental da ferramenta *FastInterface*. Este guia está organizado sob o formato de perguntas e respostas.

1. Qual a seqüência de obtenção dos artefatos de usabilidade?

Os artefatos relacionados ao projeto de interface em YP são: (1) perfil do usuário; (2) objetivos de usabilidade; (3) modelo de tarefas; (4) protótipo de interface. Um artefato serve de entrada para o outro. Portanto, é interessante procurar desenvolvê-los na ordem correta, conforme é demonstrado abaixo.



2. Devo dispensar os artefatos de usabilidade de YP se meu projeto não inclui explicitamente a implementação de uma interface com o usuário?

Alguns projetos se restringem à implementação de módulos internos, ou seja, se focam apenas no desenvolvimento, refatoramento ou aprimoramento da lógica de negócio.

Se você se inclui nesses casos procure saber do cliente se essa lógica de negócio já possui ou possuirá uma interface, ainda que o seu projeto não tenha que implementá-la.

Nesses casos ao menos desenvolva o perfil do usuário e o modelo de tarefas, pois eles podem lhe ajudar a obter uma compreensão melhor do domínio do problema. Essa compreensão pode lhe auxiliar mentalmente na detecção dos detalhes das funcionalidades do sistema e como elas serão implementadas na especificação de *user stories*, testes de aceitação, projeto arquitetural e modelo lógico de dados.

3. O projeto que irei desenvolver se trata de um aprimoramento de uma versão já existente de um produto de software. O que devo considerar na produção dos artefatos de usabilidade?

Se você se inclui nesse caso terá que estudar como a versão anterior atendia às expectativas dos usuários e no que ela ficava a desejar.

Você pode experimentar a versão anterior ou software similar e tentar especificar o seu modelo de tarefas, detectando aquelas tarefas que estão causando problemas de dificuldade de uso.

Procure fazer o modelo de tarefas da nova versão priorizando a remodelagem, ou seja, descrevendo melhores formas de acessar as funcionalidades existentes, caso isso não esteja atendendo às expectativas dos atuais usuários.

Entretanto, se novas funcionalidades forem incluídas e não há necessidade de remodelar o acesso a funcionalidades existentes, inclua tudo em um único modelo de tarefas. Nesse caso, tente respeitar o formato de acesso das novas tarefas em relação às tarefas já existentes.

4. Em que momento produzir cada um dos artefatos de usabilidade dentro de YP?

A produção dos artefatos de usabilidade em YP se inicia na fase de Conversa com o Cliente, tendo o seu fechamento na fase de Inicialização do processo.

Fase de YP	Artefato	Quando produzir?	Saída
Conversa com o Cliente	Perfil do usuário	Após conversa com cliente e futuros usuários	Descrição do perfil do usuário dentro do Documento de Visão
	Objetivos de Usabilidade	Após obtenção do Perfil do usuário	Quadro de objetivos e modos de alcance
Inicialização	Modelo de Tarefas	Após obtenção dos objetivos de usabilidade Antes de qualquer artefato funcional especificado na fase de Inicialização (especificação de US e TAs, Projeto Arquitetural e Modelo Lógico de Dados)	Modelo hierárquico de tarefas
	Protótipo de Interface	Após obtenção do modelo de tarefas Antes ou paralelamente a demais artefatos funcionais especificados na fase de Inicialização	Telas com respectivos objetos de interação para acesso as funcionalidades do sistema

Durante a realização da primeira conversa com o cliente, em que o domínio do problema do software a ser desenvolvido deverá ser definido, não se esqueça de já obter uma descrição do próprio cliente sobre o perfil dos usuários que efetivamente utilizarão o sistema.

DICA

Procure conversar com os futuros usuários para obter um delineamento do perfil com maior segurança. O ideal é que essa conversa aconteça no ambiente onde os mesmos utilizarão o sistema. Assim, você poderá extrair mais informações sobre as características do ambiente de trabalho e das pessoas que se beneficiarão do uso do software que você irá desenvolver.

DICA

Recomenda-se que os artefatos funcionais especificados na fase de Inicialização de YP sejam produzidos APÓS a obtenção do modelo de tarefa. O protótipo de interface deve ser desenvolvido logo após o modelo de tarefas ou paralelamente aos outros artefatos funcionais.

A recomendação é baseada na premissa de que o modelo de tarefas pode auxiliar numa melhor compreensão do domínio do problema, já que nele serão modeladas as formas de acesso às funcionalidades do sistema (tarefas do sistema). Assim, será mais fácil compreender o universo de funcionalidades do sistema e sua complexidade de implementação.

5. Como produzir cada um dos artefatos de usabilidade?

Para obtenção de cada um dos artefatos de usabilidade de YP, deve-se considerar algumas questões relevantes para facilitar esse processo.

	Artefato	Questões Relevantes
Conversa com o Cliente	Perfil do usuário	<ul style="list-style-type: none"> • Quais as características do grupo de usuários que usará o sistema (faixa etária, sexo, escolaridade, etc.)? • Existe algum usuário que tenha alguma característica física especial? • Os usuários já interagem com sistemas de computador? Com que frequência? Há quanto tempo? • Qual a plataforma computacional que utilizam com maior frequência? • Os usuários já interagiram com sistemas de computador similares ao que será desenvolvido (software similar ou versão anterior do sistema)? • Há necessidade de conhecimento de alguma simbologia, idioma e/ou de vocabulário específico para uso do sistema? • Como é o ambiente de uso do sistema (níveis de ruído, iluminação, estresse)?
	Objetivos de Usabilidade	<ul style="list-style-type: none"> • Com base no delineamento do perfil do usuário, o que será agregado ao projeto de interface para atender às suas necessidades e expectativas? • Como fazer com que o usuário acesse as funcionalidades do sistema mais facilmente? Há alguma necessidade especial? • Como fazer com que o usuário aprenda e memorize mais? • Como a interface será atrativa para seus usuários? O que será considerado? • Como o usuário terá que lidar com eventuais erros cometidos por ele ou pelo sistema? • Como o usuário obterá respostas de ações realizadas pelo sistema que venham a demorar? • O usuário poderá acessar alguma documentação/manual de instalação/uso do sistema? Como isso será disponibilizado?
Inicialização	Modelo de Tarefas	<ul style="list-style-type: none"> • Quais as tarefas que podem ser realizadas no sistema (funcionalidades)? • Qual o conjunto de passos para que cada uma dessas tarefas possa ser realizada pelo usuário? • O modelo de tarefa foi bem compreendido pelo cliente/usuários e portanto tem a sua aprovação?
	Protótipo de Interface	<ul style="list-style-type: none"> • A disposição das telas facilita o acesso as funcionalidades do sistema? • Os elementos de cada tela e sua posição facilitam a compreensão de como a tarefa poderá ser completada? • O protótipo foi bem compreendido pelo clientes/usuários e portanto tem a sua aprovação?

6. Quando atualizar os artefatos de usabilidade?

Sempre que possível. O ideal é que o modelo de tarefas e o protótipo de interface sejam concebidos e aprovados pelo cliente dentro da fase de Inicialização do processo. Caso o modelo de tarefas e o protótipo não tenham sido demonstrados ao cliente se

corre o risco de desenvolver toda a interface sem atender às suas expectativas, o que implicará em retrabalho.

Em alguns casos funcionalidades são adicionadas após a obtenção do modelo de tarefas ou do protótipo, ou, o acesso a funcionalidades existentes (grafo de tarefas relacionadas no modelo de tarefas) devem ser mais bem descritos, para facilitar o aprendizado da funcionalidade e seu uso no sistema.

7. Como iniciar a construção de meu modelo de tarefas?

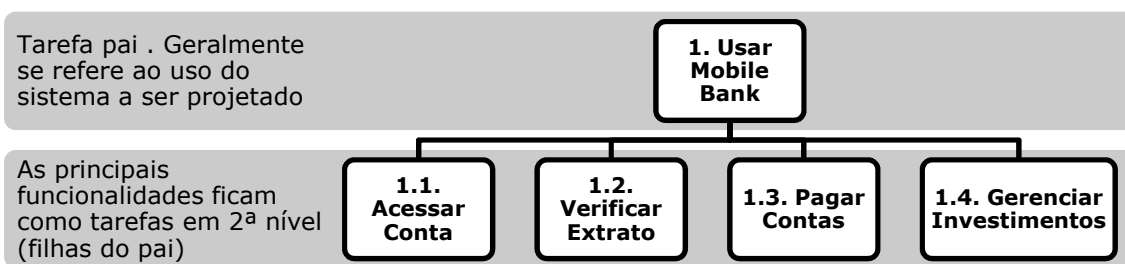
Antes de utilizar o módulo de modelagem de tarefas da ferramenta *FastInterface*, a seguinte pergunta deve ser realizada: de quais tarefas é composto o principal conjunto de funcionalidades de meu sistema? Ao responder a essa pergunta se lembre dos requisitos funcionais que foram definidos a partir de conversas com o cliente e os usuários.

Ao elaborar o modelo de tarefas procure respeitar as características, limitações e expectativas dos usuários, conforme você especificou nos artefatos de perfil do usuário e de objetivos de usabilidade, em etapa anterior. Lembre-se que o modelo de tarefas especifica o formato de acesso às funcionalidades do sistema, considerando as características de seus usuários.

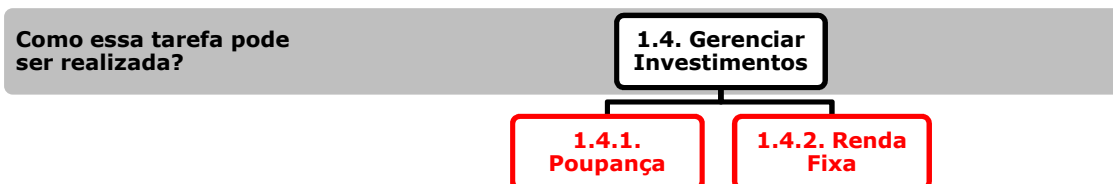
Para exemplificar a modelagem de tarefas tomaremos como exemplo o domínio de problema de acesso a banco por celular, o sistema *MobileBank*. Inicialmente poderíamos enumerar o seguinte conjunto de funcionalidades a serem acessadas pelo usuário por esse sistema, em alto nível (não se preocupe ainda em como detalhá-las):



Supondo que essas seriam as principais funcionalidades do sistema, você poderia começar a elaborar o seu modelo de tarefas dispondo a tarefa pai, 1º nível, como sendo usar o sistema, "Usar MobileBank". Ainda, as principais funcionalidades poderiam ser suas tarefas filhas, ficando as mesmas em 2º nível.

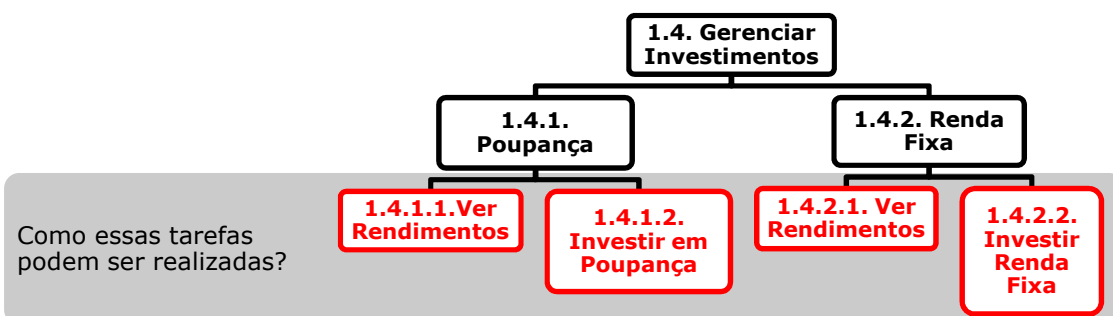


A partir dessa primeira configuração deveremos especificar se essas tarefas possuem outras sub-tarefas ou especializações. Por exemplo, a tarefa "Gerenciar Investimentos" poderia ter como sub-tarefas (tarefas filhas) as modalidades de investimento "Poupança" e "Renda Fixa". Portanto, se vê que a tarefa "Gerenciar Investimentos" se especializou e que assim descendermos um nível na hierarquia (abaixo em vermelho) para incluir suas especializações.



No entanto, observe que essas especializações aparentam ser muito abstratas. Por exemplo, o que o usuário pode fazer com as tarefas "Poupança" e "Renda Fixa"? Ou seja, o que terá de fazer para trabalhar com ambas? Pela disposição atual ainda não é possível responder a essas questões. Sendo assim, precisamos detalhar ambas as tarefas.

Na tentativa de detalhar melhor a realização das tarefas "Poupança" e "Renda Fixa" disponibilizamos 2 sub-tarefas para ambas, para a visualização de rendimentos e para a realização de investimentos. Observe a disposição dessas 2 sub-tarefas na figura abaixo (em vermelho).

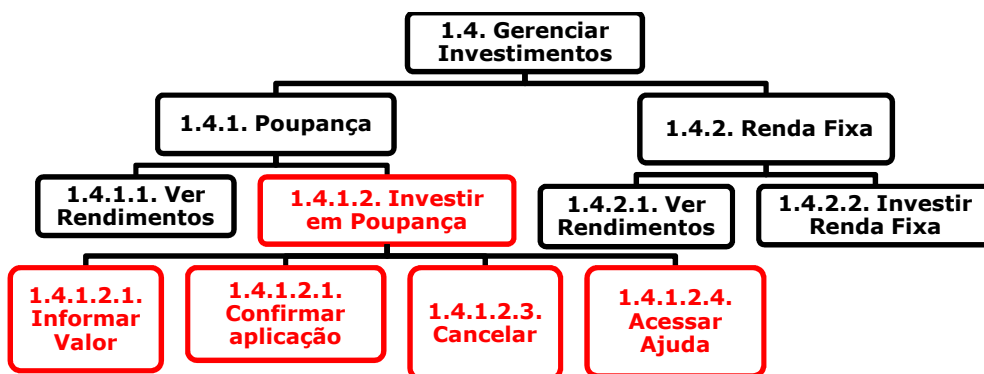


Percebe-se que para concretizar a tarefa “Investir em Poupança” temos que incluir um conjunto de ações que o usuário deve realizar, por exemplo: “Informar o Valor”, “Confirmar aplicação”, “Cancelar” e “Acessar Ajuda”. Sendo assim, vamos descer mais um nível para especificar as ações que descrevem a concretização da tarefa “Investir em Poupança” (em vermelho).

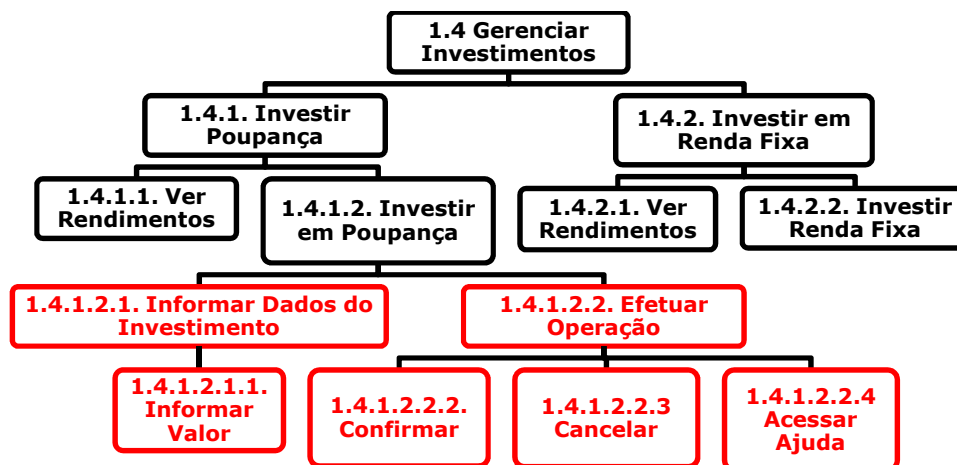
DICA

Sempre que tiver dúvida se já está na hora de parar a especificação de uma tarefa se coloque no lugar do usuário e pergunte se ainda falta especificar mais passos para cumpri-la!

Por exemplo, se você fosse o usuário o que ainda teria de fazer para realizar a subtarefa “Poupança” ou a subtarefa “Renda Fixa”? Até o momento ainda não especificamos nada de concreto, pois ainda não indicamos os passos que o usuário pode fazer para alcançar o objetivo de realizar ambas as subtarefas.



Você poderia ter pensado em realizar a tarefa “Investir em Poupança” de outra forma. Por exemplo, para realizá-la você poderia ter colocado as tarefas “Informar Dados de Investimento” e “Efetuar Operação” como suas filhas. Veja que nesse caso a pergunta “Como realizar essas tarefas?” se aplica novamente. Assim, devemos detalhar as ações necessárias para concretizar a realização delas, conforme exemplifica o grafo de tarefas abaixo (em vermelho).

**DICA****Toda vez em que desejar especializar uma tarefa...**

Adicione novas subtarefas como suas filhas, preferencialmente numa ordem seqüencial da direita para esquerda.

8. Como especificar a frequência com que uma tarefa pode ser realizada durante o uso do sistema?

Para especificar a frequência com que uma tarefa ocorre, você deverá indicar a ocorrência da mesma. As ocorrências são as seguintes:

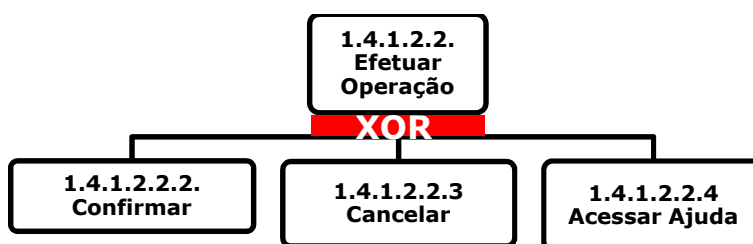
Ocorrência	Quando usar
(0,0)	Raramente é utilizada, pois indica que a tarefa nunca será realizada (acessada pelo usuário).
(0,1)	Indica que a tarefa pode não ser acessada nenhuma vez pelo usuário. Se for realizada, será apenas por uma vez.
(0,N)	Indica que a tarefa pode não ser realizada nenhuma vez pelo usuário. Se for realizada poderá repetida por várias vezes.
(1, 1)	Indica que a tarefa é obrigatoriamente acessada uma vez e somente por uma vez pelo usuário
(1, N)	Indica que a tarefa é obrigatoriamente acessada uma vez podendo ser repetida por várias vezes pelo usuário

Por padrão, ao inserir uma nova tarefa, no módulo de modelagem de tarefas de *FastInterface*, a mesma assume a ocorrência (0,1). Portanto, ao criar uma tarefa observe a ocorrência com que ela se realizará. Altere a ocorrência acessando as propriedades da tarefa (botão direito sobre a tarefa).

9. Como determinar a lógica de realização de tarefas com os operadores?

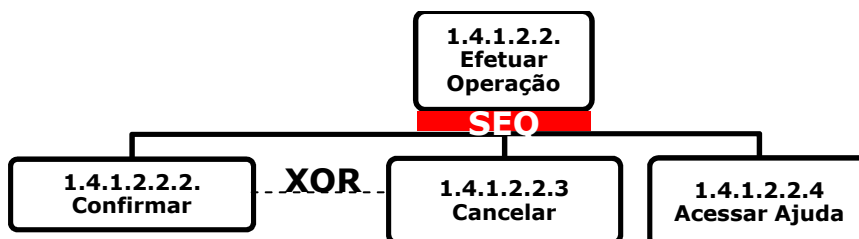
A lógica deve considerar tarefas em mesmo nível: tarefas irmãs. Portanto, pegue a tarefa pai que elas têm em comum e aplique um operador lógico que atenda a sua especificação.

Por exemplo, no grafo abaixo temos a tarefa "Efetuar Operação". Podemos "Confirmar" OU "Cancelar" OU "Acessar a Ajuda" de forma exclusiva.



Especificando de outra forma, poderíamos usar operadores em 2º nível, em que o operador é sempre aplicável entre uma tarefa irmã e a outra (à esquerda ou à direita).

Assim, poderíamos usar o operador em 1º nível como sendo SEQ. Em segundo nível aplicarmos XOR entre "Confirmar" e "Cancelar". Assim estaremos dizendo que o usuário pode ao "Efetuar Operação", confirmá-la ou cancelá-la e na seqüência acessar a ajuda.



Abaixo segue o quadro com as possibilidades de uso de operadores lógicos.

Ocorrência	Quando usar
SEQ	Execução seqüencial de sub-tarefas e/ou ações
OR	Execução de pelo menos uma das sub-tarefas e/ou ações
XOR	Execução de apenas uma das sub-tarefas e/ou ações
AND	Execução de todas as sub-tarefas e/ou ações, independentemente da ordem
SIM	Execução simultânea e independente de sub-tarefas e/ou ações
PAR	Execução concorrente (i.e. com pontos de sincronização) de sub-tarefas e/ou ações

*Ação é uma sub-tarefa concreta, sem filhos

10. As tarefas que especifiquei em meu modelo de tarefas são *user stories*?

É natural fazermos a relação entre as tarefas especificadas e as *user stories* do sistema, pois ambas auxiliam na especificação de funcionalidades.

No entanto, o modelo de tarefas objetiva especificar como o usuário deverá proceder para acessar as funcionalidades do sistema com o menor esforço possível. Esse modelo também objetiva auxiliar na seleção dos elementos que estarão presentes no desenvolvimento da interface e de sua navegação.

Já as *user stories* objetivam descrever os parâmetros para que as funcionalidades do sistema efetivamente possam ser implementadas e aceitas, numa visão mais interna de seu desenvolvimento.

Portanto, as tarefas não descrevem como as funcionalidades serão implementadas, apenas como elas poderão ser acessadas pelo usuário. Da mesma forma, as *user stories* não indicam como as funcionalidades serão acessadas, mas objetivam descrever os requisitos e as restrições de como essas funcionalidades serão implementadas para serem aceitas.

11. Se o modelo de tarefas e as *user stories* trabalham com as funcionalidades do sistema, o que deve ser construído primeiro?

Recomenda-se a construção do modelo de tarefas antes das *user stories* e testes de aceitação, já que é muito mais fácil desenvolver a compreensão do sistema pensando primeiro em como o usuário trabalhará com o mesmo.

Se a obtenção do modelo de tarefas é feita de forma coerente com o perfil do usuário, você pode, a partir das tarefas mais abstratas, selecionar o universo de *user stories* e testes de aceitação de seu sistema.

As pré e pós condições especificadas na realização de cada tarefa podem auxiliar na determinação dos testes de aceitação relativos a uma *user storie* associada diretamente a uma funcionalidade do sistema. Por exemplo, usar a tarefa "Aplicar Investimento em Renda Fixa" para ser descrita como uma *user storie*.

Podemos perguntar se há alguma pré-condição para essa tarefa ser concluída com sucesso. Elencamos várias: informar um valor múltiplo de 500, valor não pode ser vazio, etc. Essas pré-condições poderiam ser eleitas para serem alguns dos testes de aceitação dessa *user storie*. As pós-condições também podem ser consideradas, em que podemos perguntar quais as possibilidades de resultado na conclusão da tarefa "Investimento em Renda Fixa". Por exemplo: não há saldo em conta para tal, valor não é múltiplo de 500, investimento realizado com sucesso, etc. Todas essas pós-condições também podem ser relacionadas na especificação dos testes de aceitação da respectiva *user storie*.

12. Como demonstrar o modelo de tarefas e validá-lo junto ao cliente?

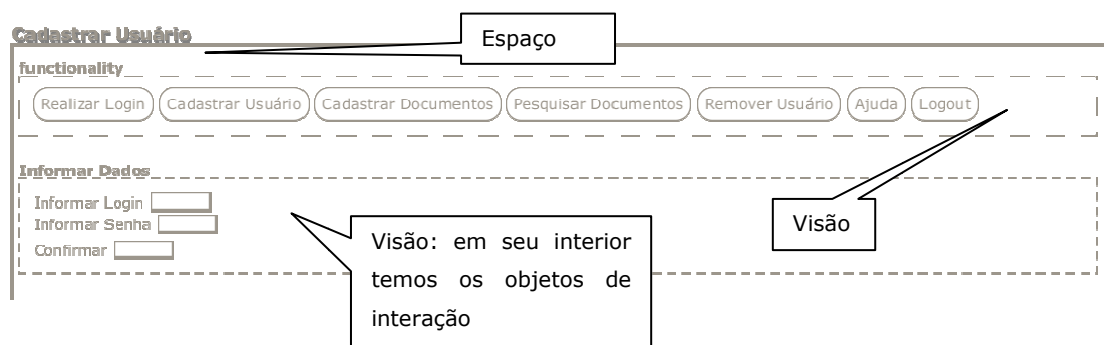
Procure se certificar junto aos usuários sobre o modelo de tarefas. Se não houve disponibilidade tente fazer isso ao menos com o cliente. Tente demonstrar o modelo de tarefas lendo o que ele especifica no uso do sistema. Exemplo de como verbalizar isso:

- Ao "Realizar Investimentos" o usuário poderá fazê-lo como "Poupança" ou como "Renda Fixa".
- Ao "Realizar Investimentos" como "Poupança" o usuário deverá informar o valor da aplicação e poderá nesse momento "Confirmar" ou "Cancelar" ou "Acessar a ajuda". O usuário poderá acessar a ajuda nenhuma ou várias vezes ao realizar a aplicação.

A intenção ao verbalizar o modelo de tarefas para o cliente é para que o mesmo possa determinar (espontaneamente ou não) se a forma de usar o sistema está coerente, ou seja, se todas as funcionalidades estão especificadas dentro das expectativas.

13. Como realizar a construção do protótipo de interface com *FastInterface*?

Ao construir o protótipo de interface com *FastInterface* você já deverá ter modelado a tarefa com o auxílio da ferramenta, em etapa anterior. Lembre-se que o modelo de tarefas será transformado em um protótipo de interface de média-fidelidade, em que serão sugeridas as telas do sistema (espaços) e os elementos de interação (objetos de interação) agrupados em áreas específicas da tela (visões).

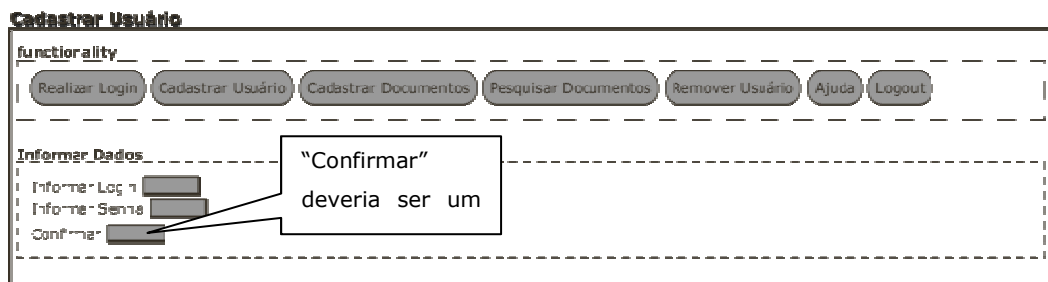


É importante que o modelo de tarefas tenha sido apresentado ao cliente, para se certificar que o mesmo contempla todo o formato de acesso às funcionalidades elencadas para o sistema. Caso o modelo de tarefas seja alterado o protótipo será gerado automaticamente de novo.

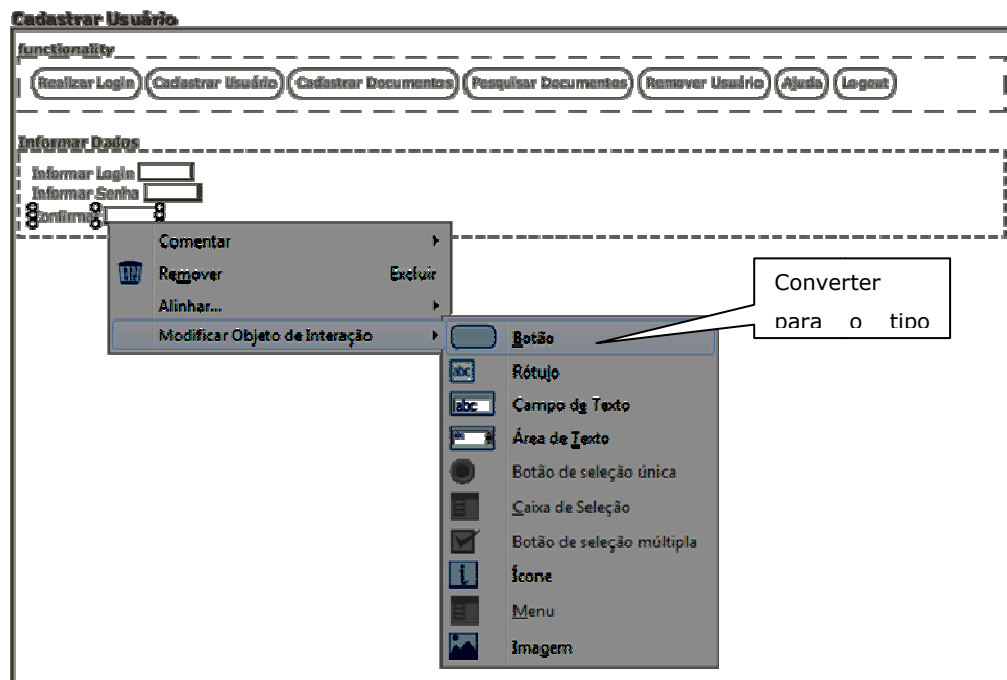
De posse do protótipo sugerido verifique a disposição de seus espaços e respectivos elementos. Se necessário, reposicione os elementos nas telas geradas (espaços)

14. Fiz o modelo de tarefas e gerei o protótipo de interface com o *FastInterface* mas o protótipo sugerido não atende às expectativas?

Em algumas telas da interface o protótipo gerado por *FastInterface* pode ter sugerido objetos de interação (rótulos, listas de seleção única ou múltipla, caixas de seleção, botões, etc.) que não atendem às expectativas. Por exemplo, o espaço "Cadastrar Usuário", contém a visão "Informar Dados". O objeto "Confirmar" veio como um campo de texto quando deveria ser um botão.



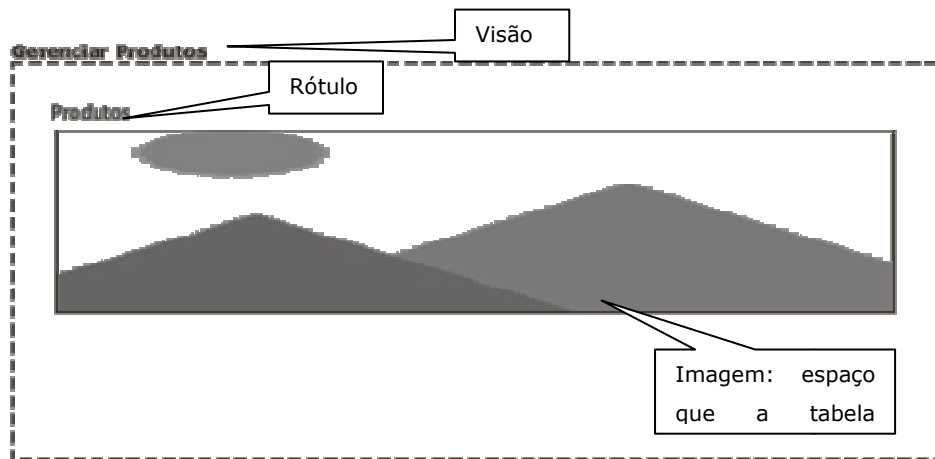
Nesse caso, você deve tentar modificar o objeto de interação.



15. Desejo incluir um tipo de objeto na tela que o *FastInterface* não disponibiliza. Como fazer para incluir esse elemento?

Inclua um rótulo (texto) e em seguida uma imagem para ilustrar o objeto que não é disponibilizado pela ferramenta. O rótulo deve conter como texto o que o objeto representa e a imagem ilustrará a posição e a área do objeto.

Exemplo, uma tabela de produtos precisa ser colocada dentro de uma das visões de uma tela qualquer do protótipo. Como não há representação equivalente devermos incluir um objeto de interação com o nome "Produtos" e incluir outro objeto imagem para ilustrar a área e a posição que essa tabela ocuparia.



16. Como validar o meu protótipo com *FastInterface*?

Exporte as visualizações dos espaços. Leve os arquivos de imagem para demonstrar ao usuário ou apresente o protótipo através da própria ferramenta.

Procure saber do cliente se ele está de acordo com as representações especificadas nas telas do protótipo e com a disposição dos elementos para o acesso das funcionalidades do sistema. Procure explicar para ele a navegação necessária para acessar as funcionalidades e usar o sistema.

17. Como implementar o código da interface?

Procure obedecer às disposições obtidas no protótipo de interface que você obteve automaticamente e refinou com a ferramenta *FastInterface*. Caso tenha alguma dúvida na implementação refaça o caminho necessário lendo no modelo de tarefas o grafo correspondente para acessar algumas das funcionalidades.

Lembre-se que o ideal é que o modelo de tarefas e o protótipo tenham sido aprovados pelo cliente e se possível também pelos usuários para que não haja equívocos durante a implementação do código da interface.

18. Como preparar o teste de usabilidade usando os artefatos de usabilidade de YP?

Para realizar um teste de usabilidade é necessário realizar um planejamento do quais funcionalidades virão a ser testadas, preparar o material do teste e selecionar os usuários de teste. Abaixo seguem algumas dicas para realizar esses procedimentos.

- **Para selecionar as funcionalidades contempladas durante a realização do teste**

- Utilize o modelo de tarefas para selecionar os grafos de tarefas que representam as principais funcionalidades. Em seguida escolha de 5 a 10 grafos de tarefas que serão candidatas a serem incluídas no teste.
- Converse com o cliente para selecionar dentre os grafos de tarefas pré-selecionados as 6 principais tarefas que efetivamente serão incluídas no teste.
- **Para obter o roteiro do teste**
 - Para cada uma das tarefas elencadas precisaremos descrever os passos que o usuário terá que desenvolver para completar cada uma delas.
 - Faça uma roteirização dos passos necessários para realização de cada tarefa. Veja o grafo no modelo de tarefas correspondente a tarefa que está sendo descrita para auxiliar nessa roteirização.
 - Estime o tempo em minutos, necessário para completar cada tarefa. Imagine que o usuário usará o seu sistema e terá apenas como guia o roteiro que você deu para ele.
- **Para selecionar os usuários de teste**
 - O ideal é que os indivíduos que irão interagir com o seu sistema durante o teste atendam as especificações obtidas no artefato perfil do usuário. Caso não seja possível, convide alguém que nunca tenha tido contato com o sistema, como, por exemplo, outro colega que não fez parte da equipe de seu projeto.

Não se esqueça de verificar todo o material para registro do teste (captura de tela, registro das expressões e verbalizações do usuário, pontos eletrônicos de conversa com o usuário de teste, etc.) com antecedência.

Ainda, não se esqueça de explicar para o usuário como se dará a condução do teste

- **Explique o que contém o roteiro de teste** que você elaborou e como ele procederá para usá-lo como guia
- **Peça para que ele verbalize o que está fazendo e pensando** ao tentar cumprir as tarefas do teste.
- **Peça para ele informar quando começou e terminou cada tarefa** do roteiro de testes
 - Caso ele demore mais do que o tempo estimado para realização da tarefa, o interrompa e explique como a mesma poderia ter sido terminada.
 - Tente registrar as dificuldades e causas que atrapalhar a conclusão da tarefa (eg. dificuldade de localização de um item, um erro do sistema,

etc.) antes de iniciar a próxima tarefa a ser realizada no roteiro de testes.

ANEXO F – GUIA PARA PROFESSORES: APLICAÇÃO DO SUPORTE METODOLÓGICO E FERRAMENTAL DE USABILIDADE NO CONTEXTO DE LABORATÓRIO DE SOFTWARE

Apresentação

Este guia tem como principal objetivo orientar os professores de laboratório de software, e de disciplinas similares, que desejem realizar a aplicação do suporte metodológico e ferramental provido por *FastInterface* no apoio à produção de artefatos de usabilidade. As recomendações e sugestões presentes neste guia foram baseadas nas experiências de adequação da ferramenta obtidas ao longo da execução do experimento de verificação do impacto desse suporte ferramental na disciplina Laboratório Engenharia de Software da UFCG, que utiliza o processo ágil e centrado no usuário easYProcess. Para facilitar a localização das informações, este guia está estruturado no formato de perguntas e respostas.

1. Quais os requisitos que a disciplina de laboratório de software deve possuir para aplicação do suporte metodológico e ferramental de *FastInterface*?

É recomendável que o laboratório de software seja precedido por disciplinas que abordem conteúdos de Engenharia de Software ligados a análise e modelagem de software, pois para o uso do *FastInterface* é recomendável que os alunos já possuam conhecimento sobre como extrair funcionalidades a partir de um domínio de problema. Esse é um pré-requisito importante na produção do modelo de tarefas, esse que serve de entrada para o projeto de interfaces com *FastInterface*, que de forma geral descreve como as funcionalidades do software serão utilizadas pelo usuário.

2. Quais os requisitos que o processo de software adotado no laboratório deve possuir para aplicação do suporte metodológico e ferramental de *FastInterface*?

Geralmente, o laboratório de software é guiado por um processo de desenvolvimento em que se recomenda a adoção de processos ágeis, principalmente por questões do curto tempo de um semestre letivo; e centrados no usuário, para que sejam

abordadas questões de usabilidade de interface de software com ênfase nos clientes e usuários. Para o emprego do *FastInterface* é recomendável que esse processo indique no mínimo as seguintes etapas na concepção da interface: (i) modelagem de tarefas; e a (ii) prototipagem de interfaces. Esses artefatos são diretamente suportados por *FastInterface*. Tal como na disciplina LES, da UFCG, com *easYProcess*, o laboratório de software deve exigir a presença de um cliente de software com demandas reais.

3. Como preparar os alunos para a aplicação de *FastInterface* no laboratório de software?

Minimamente, os alunos devem ser orientados em relação às notações para construção de modelos de tarefas (operadores de tarefa e nível de realização das tarefas) em que um exemplo de modelagem pode ser realizado com a própria ferramenta, o que seria ideal; ou a partir da demonstração de um modelo pré-concebido, usando a função de exportação de modelo de tarefas como imagem, presente na própria ferramenta. Em relação à prototipagem da interface, os alunos devem ser orientados a refinarem o protótipo automático gerado pela ferramenta, para adequar o layout e os tipos de objetos de interação da futura interface. O tutorial apresentado no Anexo D, pode ser disponibilizado aos alunos como uma referência rápida sobre a ferramenta (uma versão digital deste e de outros materiais de apoio estão disponíveis no site da ferramenta).

4. Como integrar todos membros da equipe na construção do modelo de tarefas e do protótipo de interface?

Em geral, esses artefatos correm o risco de serem desenvolvidos por apenas um membro da equipe do projeto. Até para garantir um melhor nível de descrição do modelo de tarefas, e para que todos participem, pode-se sugerir aos alunos que cada membro da equipe elabore o modelo de tarefas de acordo com a concepção que teve do domínio do problema e dos usuários. Ao final, todos os membros podem se reunir para integrar "grafos" de seus respectivas versões num único modelo de tarefas, de forma a convergir num modelo de tarefas mais completo e eficiente. Isso pode auxiliar na redução de dificuldades na descrição de realização das tarefas. Ainda, neste formato, em relação ao protótipo, gerado após a versão integrada do modelo de tarefas, a equipe pode refiná-lo de forma conjunta.

Outra sugestão, para incentivar a participação de todos, seria fazer com que cada membro de equipe desenvolvesse sozinho o modelo de tarefas e o respectivo protótipo, o que ocasionaria numa oferta de diversas versões do projeto da interface.

Estas versões poderiam ser discutidas junto ao cliente, em que o melhor formato seria selecionado. Neste caso, o cliente deverá ter maior disponibilidade de tempo.

É importante salientar que independentemente da configuração ambos os artefatos, modelo de tarefa e protótipo de interface, sejam validados pelo cliente, preferencialmente de forma seqüencial.

5. Quais as características ideais que o software a ser produzido pelas equipes do laboratório deve possuir para uso de *FastInterface*?

De uma forma geral, softwares aplicativos que exijam a produção de interfaces gráficas tanto em plataformas *web* quanto *standalone*. Considerando os casos em que os integrantes das equipes estarão analisando a tarefa e produzindo protótipos pela primeira vez, recomenda-se a aplicação com maior ênfase em domínios de negócio mais simples, como aqueles ligados à sistemas de informação. Em geral, os alunos conseguem desenvolver o raciocínio de realização das tarefas do sistema de forma mais fácil nesses casos, pois o domínio se restringe em grande parte à operações sobre os dados de entidades, auxiliando num aprendizado mais correto sobre o emprego das notações do modelo de tarefas.

6. Que tipo de projetos de software NÃO DEVEM se utilizar do suporte metodológico e ferramental de *FastInterface*?

Projetos de softwares voltados somente à implementação de serviços em camadas internas ou que contemplem apenas lógica de negócios, sem elementos de interface gráfica. Opcionalmente, em softwares cuja interação seja por meio de interfaces textuais, recomenda-se que o modelo de tarefas seja desenvolvido com a ferramenta, em que na obtenção do mesmo, e por conseguinte do protótipo, se deve buscar uma descrição e representação mínimas de como essa interface textual poderá ser acessada (*e.g.* como se invocam comandos e do que será exibido em tela enquanto os mesmos iniciam, executam e terminam)

7. Quais as características que as equipes devem possuir para o uso do suporte ferramental?

Equipes entre 3 e 4 alunos, a depender da complexidade do projeto.

8. Sobre quais aspectos as equipes do laboratório podem efetivamente se beneficiar com o uso do suporte metodológico e ferramental sugerido?

Os principais benefícios em relação às equipes e ao aprendizado dos alunos são: (i) maior produtividade e segurança na elaboração dos artefatos modelo de tarefas e

protótipo de interface, devido à facilidade de manipulação do modelo de tarefa e da obtenção do protótipo de interface inicial, automaticamente gerado, exigindo poucos esforços em seu posterior refinamento; (ii) facilidade de comunicação com o cliente, pois esses artefatos apresentam uma representação visual e notacional que facilita as demonstrações e uso desses artefatos com o mesmo; (iii) o aprendizado de técnicas básicas de projeto de interface, contribuindo para uma cultura de desenvolvimento de software que encoraja a antecipação de aspectos de projeto de uso do mesmo.

9. Como clientes efetivamente podem se beneficiar com o uso do suporte metodológico e ferramental sugerido?

Os artefatos modelo de tarefas e protótipo de interfaces passam a ser graficamente melhor representados com o suporte ferramental e notação presente em ambos é de mais fácil compreensão, conforme preconiza a metodologia MEDITE. Entretanto, recomenda-se alertar aos alunos, presumindo que os clientes não possuem capacidade para compreender sozinhos os modelos, que na apresentação dos artefatos gerados com *FastInterface* narrem o modelo de tarefas enquanto o demonstram. Em relação ao protótipo a mesma conduta deve se reproduzir, em que o de exportação em HTML (módulo de alta-fidelidade) pode ser utilizado para demonstrar ao cliente o formato de acesso às funcionalidades através da navegação das telas da futura interface.

10. Caso exista outra ferramenta de software similar, quais os requisitos que a mesma deve possuir para o manejo eficiente da usabilidade no laboratório de software?

Esta ferramenta deve preconizar uma associação entre a modelagem de tarefas e a prototipagem da interface, preferencialmente possibilitando a geração automática do protótipo para redução do esforço inicial na obtenção do mesmo. O modelo de tarefas deve possuir uma esquematização visual com notações e elementos de simples compreensão para interpretação do modo de realização das tarefas, o que se observa em modelos hierárquicos de tarefa. É interessante que o protótipo seja construído a partir de objetos de interação predefinidos, pois algumas ferramentas de prototipagem são baseadas em representações desenhadas à "mão-livre", que geram esforço na produção deste artefato, em especial no caso de alunos que não possuem habilidades em design gráfico. A possibilidade de simulação do uso da interface projetada é um recurso que também deve ser observado, pois pode auxiliar bastante na demonstração da futura interface junto aos clientes e usuários.

11. Como verificar o desempenho dos artefatos de usabilidade produzidos pelos alunos logo após a etapa de Planejamento de YP?

Após a fase de Planejamento os alunos deverão apresentar versões consistentes dos artefatos obtidos, inclusive os de usabilidade.

Em relação aos artefatos perfil do usuário e objetivos de usabilidade as equipes podem demonstrar as principais informações presentes nos mesmos em sala de aula.

Já os artefatos modelo de tarefas e o protótipo de interface carecem de maiores cuidados em relação à validação por parte do próprio cliente. É interessante que haja a aplicação de instrumentos de avaliação que extraíam, ainda que de forma subjetiva, a compreensão que o cliente teve após as demonstrações feitas desses artefatos por suas respectivas equipes de desenvolvimento.

Sugere-se como instrumento de verificação a aplicação de um breve questionário diretamente com os clientes abordando perguntas objetivas relativas à compreensão e satisfação que eles obtiveram desses artefatos e da futura interface, conforme o modelo A, disposto no final desse guia.

Caso haja problemas com algumas das equipes de acordo com as respostas de seus respectivos clientes no questionário, como a não apresentação dos artefatos, os alunos poderão ser notificados para sanarem os problemas dentro de um prazo. Por exemplo, as equipes deverão resolver os problemas relativos ao modelo de tarefas/protótipo de interfaces, até a verificação da próxima iteração do projeto. Após esse prazo, o questionário poderá ser reaplicado junto aos clientes para verificação final.

12. Como verificar o desenvolvimento da implementação da interface durante as iterações do projeto?

As equipes devem ser informadas de que o artefato *big chart*, gerado a cada término de iteração deve sumarizar a quantidade de telas da interface já implementadas.

Para facilitar o acompanhamento é interessante também que as equipes disponham a quantidade de telas esperadas para servir de medida relativa às telas que já foram implementadas. As telas esperadas podem ser estimadas através do protótipo de interface de média-fidelidade obtido na fase de planejamento.

Alternativamente, as equipes podem especificar em suas *user stories* quais telas que serão implementadas/atualizadas para que o conjunto de funcionalidades da respectiva *user storie* se dê por completada.

Caso a equipe não tenha implementado o que havia estimado em nível de interface se deve chamar a atenção da mesma. É interessante adotar esse modelo para não deixar que a interface seja implementada somente ao final.

13. Como avaliar a interface após a disponibilização da Release Final?

Quem melhor pode responder a essa pergunta são os próprios clientes dos projetos. Portanto, se sugere a aplicação de um segundo questionário com o objetivo de coletar o grau de satisfação dos clientes em relação à interface final do sistema. O modelo B, presente no final desse guia, contém uma seleção de perguntas que podem ser feitas aos clientes para obter essas informações e avaliar melhor o desempenho da interface final.

Modelo A - Questionário Pós Etapa de Planejamento para Aplicação com Clientes

Qual o nome do projeto de software do qual você é cliente?
De que forma o modelo de tarefas de seu projeto lhe foi apresentado pela equipe?
<input type="checkbox"/> Muito compreensível <input type="checkbox"/> Compreensível <input type="checkbox"/> Nem muito, nem pouco compreensível <input type="checkbox"/> Pouco compreensível <input type="checkbox"/> Incompreensível <input type="checkbox"/> Não sei, pois o modelo de tarefas não me foi apresentado
O protótipo da interface que lhe foi apresentado atendeu às suas expectativas?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não sei, pois o protótipo de interface ainda não me foi apresentado
Com o protótipo de interface que lhe foi apresentado foi possível obter uma boa noção de como será o uso do sistema?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não sei, pois o protótipo de interface ainda não me foi apresentado
Você acredita que sua equipe conseguirá atender às expectativas quanto à interface do sistema até o término do projeto?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente

Modelo B - Questionário Pós Release Final para Verificação da Satisfação dos Clientes com Interface Final

Qual o nome do projeto de software do qual você é cliente?
Indique o grau de problemas que a interface final do sistema apresenta
<input type="checkbox"/> Há problemas. Eles NÃO COMPROMETEM o uso do sistema <input type="checkbox"/> Há problemas. Eles COMPROMETEM PARCIALMENTE o uso do sistema <input type="checkbox"/> Há problemas. Eles COMPROMETEM TOTALMENTE o uso do sistema <input type="checkbox"/> Há problemas. Eles COMPROMETEM AS FUNCIONALIDADES PRINCIPAIS do sistema <input type="checkbox"/> Não há problemas <input type="checkbox"/> Não sei, ainda não me foi apresentado o sistema final.
Você está satisfeito com o modo como o sistema será utilizado (fácil uso e aprendizado)?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não sei, pois o protótipo de interface ainda não me foi apresentado
Você está satisfeito com a apresentação da interface do sistema (layout, estilos, terminologia)?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente
De uma forma geral a equipe atendeu às expectativas quanto ao que se esperava da interface do sistema?
<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo <input type="checkbox"/> Nem concordo, nem discordo <input type="checkbox"/> Discordo <input type="checkbox"/> Discordo totalmente