

# Integração de Padrões e Técnicas da Engenharia de Software no Desenvolvimento de Interfaces Humano - Computador com Qualidade

Murilo de Melo Jatobá  
Universidade Federal Fluminense

Graduação em Ciências da Computação

Pólo Universitário de Rio das Ostras – PURO

murilomelo555 at gmail.com

## Resumo

Este estudo apresenta possíveis aplicações de ferramentas e técnicas da engenharia de software visando o aumento da produtividade e da qualidade dos artefatos gerados no processo de desenvolvimento de interfaces humano-computador com qualidade. Este trabalho tem como objetivo, fornecer um arcabouço que visa colaborar com desenvolvedores com o interesse de produzir interfaces humano-computador com qualidade, usando para tal fim, de técnicas conhecidas e amplamente testadas na engenharia de software.

Palavras-Chave: engenharia de software, sistemas interativos, qualidade de software, interface humano-computador, frameworks.

## Abstract

*This study provides potential applications of tools and techniques in software engineering in order to increase productivity and quality of the artifacts generated during the development of human-computer interfaces with quality. This article aims to provide a framework that aims to collaborate with developers in the interest of producing human-computer interfaces with quality, using for that purpose, techniques known and widely tested in software engineering*

*Key words: software engineering, interactive systems, software quality, Human Computer interface frameworks.*

## I- Introdução

É de conhecimento geral que o desenvolvimento de interfaces responsivas e que agradam o usuário do sistema em questão é fundamental para o sucesso do sistema, assim como são responsáveis diretas pela produtividade, satisfação do cliente e diversas. Tais variáveis, muito determinantes no sucesso/fracasso de um sistema precisam ser mapeadas e estudadas mais a fundo.

Diversas questões podem ser levantadas a respeito do tratamento dado as interfaces pelos processos de desenvolvimento de software, por exemplo, “Qual momento do processo de

desenvolvimento deve ser reservado a Engenharia de Usabilidade ?”

Na engenharia de software, existem muitas publicações a respeito de processos de desenvolvimento amplamente testados pela comunidade e com qualidade comprovada no desenvolvimento de software, podemos citar a ISO/IEC 12207:2008 – Software Life Cycle Processes e o IEEE SWEBOK – Software Engineering Body of Knowledge, porém ao consultar qualquer uma podemos notar que não é reservado uma etapa para a engenharia de usabilidade.

A engenharia de Usabilidade é definida como uma abordagem metodológica e de natureza científica de produção, que objetiva a entrega de um produto usável ao usuário. Para isso, utiliza métodos para agrupar requerimentos, desenvolver e testar protótipos, avaliar projetos alternativos, analisar problemas de usabilidade, propor soluções e testes com usuário (Garner, 2003)[1]

Para a realização de um projeto de interface humano-computador de qualidade, é necessário um conhecimento sobre o domínio e o contexto no qual o software é utilizado, tal conhecimento é de difícil extração e por esse motivo são necessários testes, implementações de *frameworks* e ISO's de sucesso comprovado e inevitavelmente de muito contato com os *stakeholders* do sistema.

Para fins de criação de padrões e *frameworks* algumas normas ISO foram estabelecidas, como por exemplo a ISO 13407 que define a criação de protótipos de interface que devem ser testados pelos usuários finais do sistema e assim em intervalos de tempo determinados ciclos, a interface poderá ser melhorada atendendo a novos eventuais requisitos, é notória a semelhança com o modelo de processo de desenvolvimento de software, conhecido como iterativo e incremental. A ISO 13407 se implementada desde cedo no desenvolvimento, pode reduzir o risco de falhas conceituais do projeto, garantindo que, a cada ciclo, o sistema responda cada vez melhor às expectativas e necessidades dos usuários em suas tarefas. (Cybis,

Betiol & Faust, 2007).[2]

A integração entre as áreas de Engenharia de software e de Interface humano-computador é um processo complexo por diversos fatores tais como, equipes com formação diferentes e focos distintos, observadas estas dificuldades para integração foram elaborados padrões para a Integração Engenharia de Software e Interface humano-computador fornecendo assim um vocabulário comum às duas áreas além do compartilhamento de práticas de sucesso.

## II - Duas Perspectivas Sobre Qualidade

O objetivo de aplicarmos técnicas de IHC no processo de desenvolvimento de software é claramente, incrementar qualidade no software, mas qualidade em termos de software é um termo muito abstrato, sendo assim necessário recorrermos à uma definição. De acordo com a definição de Pressman[3], qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais. Essa definição demonstra a visão de qualidade de software com um enfoque mais funcional "requisitos funcionais e de desempenho" enquanto à área de interação humano-computador possui seu enfoque na qualidade da experiência dos usuários ao passo que utilizam o software. A princípio as duas áreas se mostram bastante distintas e portanto de difícil integração, porém se recorrermos a documentos relevantes para a engenharia de software, tais como o manifesto ágil[5] encontramos diversas interseções entre as duas áreas, o Manifesto Ágil afirma que a prioridade de um processo de desenvolvimento de software deve ser satisfazer o cliente através de entregas antecipadas e contínuas de software de valor ou seja, uma abordagem dirigida ao usuário bem similar à técnica de interface humano-computador conhecida como Design Centrado no Usuário (UCD, do inglês User-Centered Design).

Métodos Ágeis e UCD são metodologias diferentes no que se diz à suas abordagens à respeito de coletas de requisitos, elaboração de projetos de forma antecipada e ciclos de *feedback* porém possuem a semelhança de manterem o cliente satisfeito durante o processo de desenvolvimento, sendo o processo de UCD mais demorado ou seja, são duas metodologias centradas no usuário.

Mesmo usando de abordagens diferentes, e tendo definições distintas para o termo "qualidade de software", ambas áreas utilizam de recursos que em muitos casos se complementam e assim contribuem para um software funcional e que satisfaça às necessidades de uso do cliente em questão.

## III - Análise Comparativa IHC x E.S

Os modelos desenvolvidos em ES dizem respeito à modelagem das entidades e funções da aplicação, enquanto que em IHC, os modelos são mais concretos, representando características pertinentes à interação entre um usuário e um SI (tais como: suas tarefas, estrutura navegacional entre as telas, etc)[6] ou seja, na ES há um foco na funcionalidade e o estudo das IHC buscam fornecer usabilidade às funcionalidades elicitadas nos artefatos de software gerados pela engenharia de software

Quanto aos modelos de ciclo de vida, na ES destacam-se os modelos em cascata (Royce, 1970) que evidencia a sequência em cascata de uma fase para outra e o espiral (Boehm, 1988), que representa um processo de desenvolvimento de software em muitos *loops* permitindo a avaliação e redução de riscos.

Em IHC, por sua vez, existem 2 modelos de ciclo de vida, são eles: Modelo em estrela (Hix e Hartson, 1993), cuja atividade central é a avaliação e o início de um processo de desenvolvimento de software pode acontecer por qualquer uma das outras atividades e o modelo de engenharia de usabilidade(abordado no início deste artigo).

Tabela I – Quadro Comparativo IHC x ES

Fatores	Engenharia de Software	Interação Humano-Computador
Origem	Década de 70	Década de 80
Foco	Qualidade do processo e do produto	Qualidade de uso(satisfação do cliente ao usar o produto)
A modelagem recai sobre	As entidades e funções da aplicação	Tarefas, interação, navegação
Modelos de Ciclos de Vida	Cascata, espiral	Estrela e engenharia de usabilidade
Características do produto	Devem atender requisitos funcionais e não funcionais	Devem Atender requisitos de usabilidade
Utilização de Padrões	De Projeto(apoio a implementação e modelagem)	De usabilidade(apoio ergonômico à modelagem das interfaces)

## **IV – Iniciativas usadas para integrar IHC e ES para gerar interfaces considerando requisitos funcionais**

O termo requisito não se aplica de forma consistente a indústria de software. Isso se deve ao fato de em alguns casos os requisitos serem encarados como uma declaração abstrata de uma função que o sistema deve fornecer ou uma restrição do sistema. Em outros casos ele é uma descrição detalhada de uma função do sistema (SOMMERVILLE, 2008)[7].

Definido o conceito de requisito funcional de um sistema de software Sutcliffe (2002)[8] descreve um método para fazer a modelagem e validação de requisitos baseada em cenários. O objetivo desta fase é identificar os requisitos com a participação do usuário, numa abordagem de projeto centrado no usuário. Os requisitos identificados através dos cenários são discutidos e expressos em storyboards. O próximo passo é analisar a viabilidade dos requisitos de usabilidade fazendo um estudo sobre as opções de projeto e em seguida são usados protótipos para fins de validação dos requisitos.

## **V - Vantagens da Integração das áreas em questão**

A engenharia de software junto com as técnicas de gestão de projetos oferecem um extenso ferramental para trabalharmos com interfaces em sistemas interativos, pelo fato de se utilizar de ferramentas já testadas e de sucesso comprovado pelas mais diversas áreas do conhecimento humano.

Outro fator relevante é a necessidade de se estabelecer um padrão para criações de interfaces, sempre buscando produzir software atendendo a todas funcionalidades pedidas (engenharia de software) e fazendo que essas funcionalidades possam ser efetivamente usadas pelos usuários do sistema acelerando o processo quando possível e apoiando os clientes em suas atividades rotineiras(Interação Humano-Computador).

## Referências Bibliográficas :

[1]: MOLISANI, Danielle; KAMINSKI, Paulo. "FERRAMENTA PARA CLASSIFICAÇÃO DE EMPRESAS QUANTO À CAPACIDADE DE DESENVOLVIMENTO DE PRODUTOS ". Artigos USP. Disponível em:<[http://sites.poli.usp.br/d/pme2600/2007/Artigos/Art\\_TCC\\_049\\_2007.pdf](http://sites.poli.usp.br/d/pme2600/2007/Artigos/Art_TCC_049_2007.pdf)> Acesso em 01 de maio de 2014.

[2]: SECCO, Lillian; "AVALIAÇÃO DE USO DE INTERFACE DE SOFTWARE PARA BARES". Monografia De Conclusão do Curso de Graduação em Sistemas da Informação da UNOCHAPECO. Disponível em <<http://www5.unochapeco.edu.br/pergamum/biblioteca/php/images/00006E/00006EC5.pdf>> Acesso em 01 de maio de 2014.

[3]: SELLERI, Fernando; "ASPECTOS SOBRE QUALIDADE EM SOFTWARE LIVRE". Anais Depto. Ciências da Computação UNEMAT. Disponível em: <<http://www.sucesumt.org.br/mtdigital/anais/files/AspectossobreQualidadeemSoftwareLivre.pdf>>