



Casos de uso essenciais ajudam a gerar interfaces que respeitam o usuário

Raul Sidnei Wazlawick

Universidade Federal de Santa Catarina – raul@inf.ufsc.br

Patrícia Wazlawick

Faculdade Antonio Meneghetti - patriciawazla@gmail.com

Eixo Temático: Ciência, Tecnologia, Inovação e Sociedade

1 Introdução: O problema das interfaces difíceis

Mesmo após vários anos de evolução das técnicas de desenvolvimento de sistemas amigáveis, ainda nos deparamos com interfaces complicadas de usar, mesmo em se tratando de grandes sistemas amplamente disponíveis, como os de governo eletrônico.

Acreditamos que essa dificuldade deve-se ao fato de que os projetistas de tais sistemas desenvolvem a arquitetura de interface baseada não nos processos, mas nos dados. E por que isso é ruim? Tentaremos explicar através de uma pequena história: “Todo dia pela manhã, levanto para fazer café. Guardo a chaleira em um balcão junto com as panelas, o coador de papel em outro balcão junto com os guardanapos e o pó de café em outro balcão junto com a farinha e açúcar. Ou seja, eu agrupo os itens necessários para preparar o café por semelhança física. Assim, quando vou preparar o café preciso abrir três balcões para procurar os itens necessários. A pergunta é: por que não guardo a chaleira, coador e pó de café juntos no mesmo balcão, já que sempre uso os três em conjunto?”

As breves considerações apresentadas neste trabalho, no que dizem respeito à área de tecnologia da informação, permitem compreender o quanto esta discussão está em relação direta com a formação não apenas técnica e de conhecimento do profissional desta área, mas essencialmente em relação à formação da postura de responsabilidade profissional orientada à funcionalidade do escopo pelo qual esta atividade profissional existe.

O fato mostrado na história acima ilustra a maneira como muitas interfaces de sistemas computacionais são projetadas: ações semelhantes em uma interface são colocadas juntas apenas porque são semelhantes. Mas não se colocam as ações organizadas organicamente em função dos processos mais comuns que os eventuais usuários vão utilizar.



Por isso alguns sistemas são tão difíceis de usar: os usuários simplesmente não encontram as funções que precisam porque elas não estão por perto. Já vimos mesmo casos de pessoas que precisavam entrar em uma área de um sistema para tomar nota (em papel) de algumas informações e depois redigitá-las em outra parte do sistema (absurdo).

2 A Solução: Casos de uso essenciais

Os *casos de uso essenciais* já resolvem estes problemas, mas ainda são relativamente mal compreendidos e pouco usados, pois a maioria dos livros didáticos apresenta essa ferramenta de forma muito genérica, deixando mal definido o conjunto de processos a considerar como caso de uso, com o que fica difícil detectar se a análise está completa e correta. Para minimizar este problema, elaboramos ao longo de anos de trabalho uma visão de casos de uso essenciais (Wazlawick, 2011) cuja ideia é basicamente descrita a seguir.

A funcionalidade a ser implementada é organizada em termos de um *processo de negócio* que um usuário executa. O analista deve se perguntar se o processo é completo, ou seja, se um eventual usuário poderia ir ao computador, ligá-lo, e após executar o processo, desligá-lo, pois teria alcançado algum objetivo de negócio. Essa definição descarta processos como “*login*”, que por si só não realiza nenhum objetivo de negócio, e processos parciais, que apesar de serem partes de casos de uso não são completos de acordo com a definição acima.

O caso de uso essencial deve ser isento de tecnologia. Então, o que o analista precisa fazer primeiro é descobrir as reais *necessidades* do usuário. Depois, ele vai projetar uma tecnologia que satisfaça essas necessidades. Muitos analistas já iniciam o levantamento de requisitos com desenhos de interfaces e já produzem um modelo de interação prematuro, porque ainda sabem pouco sobre as reais necessidades do usuário. Analistas erram ao dizer “o usuário precisa *desta* interface”: usuários precisam processar *informação*; a interface é só um *meio* para realizar este objetivo.

Casos de uso devem ser descritos como simples *trocas de informação*, algo no estilo “o sistema informa..., o usuário informa...”. Evita-se desta forma a colocação de processamento interno, desnecessário para compreender o processo interativo do sistema com o usuário. Evita-se também dessa forma a colocação de passos desnecessários que não



apresentam efetiva troca de informação, coisas como “o sistema solicita...” ou “o usuário acessa...”, que são meras operações de navegação em uma interface hipotética.

Cada caso de uso deve ter um *fluxo principal*, que é o processo modelo no qual tudo corre bem. Este processo típico e ideal é que vai nortear a estrutura principal da interface. Depois, os *fluxos alternativos* serão adicionados de forma sistemática, para mostrar que ações poderiam ser tomadas caso alguma falha ocorresse durante o processo. Tal elaboração coloca nas mãos do usuário soluções simples e acessíveis para seus maiores problemas.

Técnicas de análise baseadas em casos de uso já existem desde o início da década de 1990. Porém, a falta de sistematização na definição destes casos de uso é o principal fator que ainda vem impedindo que sejam uma ferramenta efetiva para a elaboração de interfaces melhores.

Um caso de uso essencial, assim criado, pode dar origem a uma boa interface pela colocação também sistemática dos campos e controles com dados a serem inseridos e recebidos pelo usuário em uma sequência lógica e visualmente acessível. Sequências alternativas como tratadores de exceção podem ser modelados colateralmente nestas interfaces, mas o ponto é que estarão presentes e visíveis próximos às ações de usuário que poderão eventualmente torná-los necessários.

3 Considerações Finais

A experiência com casos de uso essenciais mostra que sua utilização tende a produzir interfaces mais bem adaptadas às reais necessidades do usuário. O *Chaos Report* (Standish Group, 1995) mostra que mais de 50% das funcionalidades implementadas em sistemas nunca são usadas. Isso ocorre em grande parte porque os analistas insistem em apresentar soluções preconcebidas antes de entender realmente qual a necessidade do futuro usuário.

O levantamento de requisitos baseado em casos de uso essenciais pode, então, amenizar este problema, pois leva o analista a concentrar-se nas necessidades do usuário em termos de informações e não de estruturas de interface. De posse deste modelo de casos de uso essenciais, um bom projeto de interface pode ser criado, de forma a atender inicialmente o fluxo principal, mas também deixando a possibilidade de realizar todos os fluxos alternativos da forma mais imediata possível. Desta forma, podemos verificar que muitos problemas



RESPONSABILIDADE E RECIPROCIDADE

Valores Sociais para uma Economia Sustentável

poderiam ser resolvidos caso os analistas desenvolvessem antes, em si mesmos, uma *forma mentis* adequada e direcionada às necessidades práticas de seus clientes, de modo a antecipar os problemas, resolvê-los, e proporcionar uma operação mais condizente com o escopo e o resultado funcional de seu produto/sistema, em base às demandas do usuário. O analista deveria ser não apenas um profissional técnico, mas junto disso, um profissional com competências e habilidades empreendedoras, principalmente no contexto contemporâneo. Falamos aqui de desenvolvimento de uma postura profissional responsável, no mais completo sentido do termo. Teria de ser a formação dessa responsabilidade no profissional da área de tecnologia da informação uma inovação? Por que esta inovação é tão tardia?

Referências

Standish Group. (1995). **Chaos Report**. Disponível em www.projectsmart.co.uk/docs/chaos-report.pdf. Acesso em: 10 março 2010.

WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2. ed. Série didática da Sociedade Brasileira de Computação. Rio de Janeiro: Elsevier, 2011.