

A participação dos usuários no desenvolvimento de sistemas de informação

Ariadne Chloë Furnival

Resumo

*As abordagens tradicionais de design de sistemas de informação (SIs) computadorizados fundamentam-se em certos pressupostos técnicos que geralmente não englobam fatores sociais e psicológicos da organização na qual o novo sistema será implantado. Os custos de se ignorar tais fatores são as vezes altos, pois usuários insatisfeitos podem resistir ao uso do sistema, ou usá-lo de uma maneira diferente daquela para a qual foi originalmente projetado. As limitações das metodologias de **design** tradicionais têm sido mitigadas pela adoção de metodologias **soft**, que baseiam-se na idéia de que somente a participação contínua dos usuários no desenvolvimento do novo sistema pode evitar problemas futuros, pois são os usuários – e não o analista – que detêm o conhecimento especializado do seu trabalho, o que é necessário para a construção de um sistema de alta qualidade e, portanto, produtivo.*

Palavras-chave

Sistemas de informação; Desenvolvimento; Participação de usuários.

INTRODUÇÃO

Os casos de projetos para sistemas de informação automatizados que fracassaram são muitos, em geral resultando em não-uso, subuso, ou até em sabotagem do sistema, pois este não atinge os objetivos para os quais foi projetado, ou os faz de uma forma diferente da prevista. Mas o que é difícil de se entender é como tais sistemas de informação (SIs) chegam ao ponto de serem implementados. Pois a produtividade pode até cair, após a implementação de um novo sistema computadorizado, sobretudo se a maneira como os funcionários trabalhavam, antes da implementação do sistema, não foi levada em consideração ¹.

Como pode ser possível que milhares de dólares sejam investidos em sistemas que não fazem o que deveriam fazer? Para começar a abordar este tópico, precisamos analisar o processo de *design* destes sistemas e os pressupostos básicos que têm sustentado as metodologias predominantes de análise de sistemas e seus conseqüentes projetos (*designs*) de sistemas. Este rápido apanhado das metodologias tradicionais nos levará naturalmente a um breve exame da filosofia que sustenta as metodologias de *design* que se apresentam como alternativas ao *design* tradicional. Tais abordagens, as quais surgiram na década de 80, focalizam a participação dos usuários no processo de *design* de um sistema. E ainda concluiremos com uma breve descrição do estudo de um caso, que envolveu a participação de bibliotecários no *design* de seu novo sistema computadorizado.

Dado ao fato que os bibliotecários e profissionais de informação de hoje têm, como notou Mey², que conviver com os profissionais de computação (um conjunto bastante abrangente que inclui os analistas de sistemas, programadores, vendedores, técnicos de manutenção e outros), a questão do papel dos usuários no desenvolvimento de sistemas computadorizados de informação (um papel quase nunca questionado quando o sistema é manual) é entendida como sendo de muita relevância para estudantes e profissionais de biblioteconomia e ciência da informação.

AS ABORDAGENS TRADICIONAIS DO DESIGN DE SIS

Será delineada aqui uma breve descrição das fases principais das metodologias tradicionais de análise de sistemas, segundo Flynn³.

A determinação dos requisitos

Esta é a primeira fase do *design* na qual os requisitos dos usuários são determinados. É evidente que muitos desses requisitos podem coincidir com aqueles do sistema atual. Por sua vez, esta fase contém as seguintes etapas:

- a definição do problema pela gerência da organização;
- um estudo de exeqüibilidade no qual várias alternativas para o sistema proposto seriam delineadas, bem como as previsões de custos e benefícios de cada uma;
- aquisição dos requisitos, mediante observação dos usuários, análise da documentação existente, entrevistas ou questionários;

O produto principal desta primeira fase é um documento denominado especificação dos requisitos .

Análise

Nos últimos anos, esta fase tem sido caracterizada pela abordagem de modelagem conceitual ou daquela orientada a objetos (*object-oriented*), acrescentando à especificação mais detalhes em uma linguagem mais precisa, formal e algumas vezes, matemática, mas, ao mesmo tempo, conservando os termos e conceitos dos usuários, isto é, a semântica dos usuários. Embora sendo precisa, a especificação desta fase modela o sistema projetado em um nível abstrato – ou seja, não contém detalhes da implementação para o computador ou da representação dos dados.

Design lógico

Como o objetivo principal desta terceira fase é a produção de um documento de *design* do sistema desejado no qual a implementação computacional será baseada, dá-se ênfase à representação dos conceitos dos usuários das fases anteriores em termos computacionais, que incluirão processos que alteram os dados, regras integradas aos processos etc. Assim, é nesta fase que o *design* dos procedimentos e dos processos dos usuários (i.e. das transações que o usuário desempenhará com o sistema) acontecerá.

Design físico

Esta é a última fase do *design* , e é nela que o *design* lógico da fase anterior é traduzido em projetos de *software* e *hardware* .

Implementação e testagem

É nesta fase que um sistema físico é produzido: O *hardware* e o *software* são adquiridos (no caso do *software*, adquirido ou escrito) e o treinamento dos usuários é realizado.

Nas atividades de testagem (atividades muito negligenciadas, alega Flynn⁴), além de checar a própria viabilidade prática dos módulos do *software* para detectar problemas de programação (*bugs*), o comportamento esperado do sistema é avaliado à luz da atitude real dos usuários diante do sistema, o que normalmente resultará em modificações.

Manutenção

Segundo Layzell e Loucopoulos⁵, há três tipos de atividade que caracterizam esta fase pós-implementação:

- manutenção corretiva – que pode incluir, desde a correção de emergências em que o sistema pára de funcionar, até as correções mais rotineiras, pelas quais o programa é ajustado para corresponder às especificações;

- manutenção adaptativa – que responde a mudanças previstas no ambiente de processamento de dados da organização em que se localiza o sistema (e.g. devido a mudanças legais da organização, ou do país);
- manutenção perfectiva – cujo objetivo é o aprimoramento do sistema em termos de *performance*, ou seu ajuste para corresponder aos usos evolutivos dos usuários.

A fase de manutenção também é tradicionalmente ignorada⁶, o que é paradoxal, quando lembramos que a manutenção de um sistema constitui 80% de sua existência, dado que, em média, um sistema típico leva até dois anos para ser desenvolvido e terá uma vida aproximada de 10 anos antes de ser substituído.

Existem numerosas variações desta metodologia tradicional; por exemplo, cada fase pode ser iterada, ou o insumo de cada fase pode incluir a avaliação da própria fase de desenvolvimento pelos usuários (ou seja, retroalimentação (*feedback*)), o que faz com que esta abordagem se torne mais flexível.

Cabe perguntar: Quais são as falhas destas metodologias? Elas produzem sistemas bem sucedidos? Flynn⁷ define um sistema bem sucedido como aquele que satisfaz seus objetivos de qualidade e produtividade. O alvo de qualidade é aquele que para o fato de que o sistema atende seus requisitos, ao passo que o alvo de produtividade significa que o sistema foi desenvolvido dentro do prazo de tempo e dentro do orçamento previsto. Entretanto, relatos de sistemas malsucedidos proliferam. Flynn⁸ cita estatísticas de um relatório (de 1990) da firma de analistas no Reino Unido, Peat Marwick McLintock, que revela que “30% dos maiores projetos de computadorização ultrapassaram em uma escala elevada, seus orçamentos e seus cronogramas e, se completados, não fazem as tarefas para as quais foram projetados”. No mesmo contexto, dados levantados pelo Software Engineering Institute na Carnegie-Mellon University nos EUA em 1988 mostraram que 80% das 150 companhias de *software* (consideradas líderes no campo) avaliadas tiveram processos de desenvolvimento de projetos “caóticos”, com problemas sérios nas áreas de controle de mudanças. Problemas comuns com *software* tendem a concentrar-se nos problemas da não-entrega do *software* aos usuários, ou *software* entregue, mas que falha no sentido que não satisfaz todos os requisitos e conseqüentemente é rejeitado pelos usuários. Relatórios do Reino Unido mostram, por exemplo, que até 20% do investimento em projetos de sistemas são desperdiçados em *softwares* nunca entregues ou entregues mas não usados. Mais estatísticas demonstram a situação: uma estimativa mostra que até um bilhão de libras esterlinas extras são gastos em *softwares* deficientes - i.e. *software* que exigem muita manutenção não prevista; e um relatório de 1986 do governo inglês mostra que, de 20 projetos-piloto iniciados em 1982, apenas a metade funcionou como os usuários esperavam, ou foi plenamente aceita ou trouxe benefícios à organização; e 20% foram rejeitados. Nos EUA, as agências federais perderam, em 1989, \$17 bilhões em *software* fraco, em comparação com \$9 bilhões em 1982⁹. A autora não dispõe de dados deste tipo para o Brasil, mas supõe-se que a situação não seja muito distinta.

A partir desses abundantes exemplos de sistemas fracassados, os problemas podem ser categorizados como aqueles de qualidade e produtividade, conforme resumimos abaixo¹⁰:

Qualidade

- **Enfoque errado.** São escolhidas atividades erradas para se automatizar, pois o problema não é definido corretamente ou o sistema pode entrar em conflito com as metas e estratégias da organização.
- **Negligência da organização mais ampla:** Fatores psicológicos e sociais mais amplos podem ser negligenciados, como o grau de descentralização ou centralização da organização, ou o grau de aceitação ou “usabilidade” (*usability*) do sistema.

- **Análise incorreta:** As atividades corretas são identificadas, mas pode-se cometer erros na análise das necessidades de informação devido a técnicas fracas de desenvolvimento.
- **Motivos errados:** Tecnocratas ou fãs das novas tecnologias com influência na organização querem implementá-las (o chamado *technology push*), ou gerentes que querem estender seu poder e influência por meio do sistema computadorizado (o chamado *political pull*) têm muita influência na decisão de implementação do novo sistema.

Produtividade

Os problemas de produtividade são causados por requisitos que mudam devido a:

- **Nível de exigência:** Os usuários se tornam mais exigentes enquanto o projeto progride.
- **Eventos externos:** As mudanças de fatores externos como da tecnologia, da legislação, do mercado ou do ambiente político muitas vezes afetam diretamente as necessidades do sistema.
- **Implementação não exeqüível.** Aspectos não-factíveis dos requisitos são reconhecidos como tais apenas durante a implementação e testagem.

AS FALHAS DAS ABORDAGENS TRADICIONAIS

As principais críticas às metodologias tradicionais de análise de sistemas dirigem-se ao pressuposto básico no qual elas se sustentam. A saber: que os requisitos podem ser clara e precisamente especificados lingüisticamente desde o começo do projeto. Como nota Partridge¹¹, estas metodologias definem a maneira 'correta' de desenvolver sistemas como aquela que primeiro especifica completamente o problema. Este pressuposto está relacionado com os principais produtos do processo de *design* de um sistema, que são os documentos e o conhecimento que podem ser usados na tomada de decisões cruciais sobre o futuro sistema¹². Tal exigência de documentação formal dos requisitos tem raízes na origem histórica das metodologias tradicionais, pois estas metodologias surgiram no contexto de grandes projetos governamentais nos quais especificações escritas fizeram parte do processo de concorrência para o contrato: firmas concorrentes trabalhariam com a mesma especificação dos requisitos para gerar uma estimativa de orçamento e cronograma para ganhar o projeto¹³. Além de duvidar do fato de que os requisitos dos usuários podem mesmo ser determinados (e fixados) desde o começo do projeto, as críticas às abordagens tradicionais focalizam o aspecto da comunicação formal no próprio processo do *design* como sendo uma fonte de obstáculos ao desenvolvimento de sistemas bem-sucedidos.

Por exemplo, Kensing & Munk-Madsen¹⁴ notam como esta predileção pela comunicação escrita no processo do *design* baseia-se em um modelo implícito de comunicação no qual ela é vista como alguma coisa criada em um lugar (e.g. na cabeça do *designer*) e que daí é transportada até aos receptores (e.g. os usuários). Note-se que, neste modelo, os receptores são passivos e que "a comunicação bem sucedida é determinada pela capacidade do emissor (i.e. o *designer*) em formular uma mensagem rigorosa e completa"¹⁵. Portanto, métodos predominantes como os de Yourdon ou de Jackson dedicam muita atenção à produção de especificações rígidas pelo analista numa linguagem altamente abstrata e formal, pois a linguagem natural, na qual os requisitos foram originalmente expressos, é nebulosa e ambígua demais - aberta a múltiplas interpretações do mesmo fato¹⁶.

Por sua vez, as raízes históricas deste rigor de descrição encontram-se no modelo cartesiano do conhecimento, que tem predominado desde o Renascimento. Este modelo, que permeia o modo de

pensar ocidental, fundamenta-se no dualismo, segundo o qual existe um mundo interior de experiências (a mente) e um mundo externo de objetos (realidade externa). A linguagem,

conforme este modelo, reflete (objetivamente) o mundo externo de objetos¹⁷, em vez de construí-lo, como entendemos agora, principalmente graças ao trabalho do lingüista Saussure. A partir do Renascimento, é dada a primazia à objetividade e ao quantitativo, colocando, ao mesmo tempo, a subjetividade e o qualitativo em um segundo plano¹⁸.

Como já vimos, as implicações desta ontologia são evidentes na linha predominante da análise de sistemas. Supõe-se que o “problema” (i.e. a necessidade de um sistema novo) tem uma base lógica, que pode ser expressa em uma linguagem precisa e resolvida com soluções computacionais. Sem dúvida, tais metodologias e soluções tinham seu valor algumas décadas atrás quando os sistemas de processamento em lote (*batch systems*) predominaram (e portanto, as questões de interfaces interativas e amigáveis não foram tão urgentes quanto são hoje em dia com sistemas *on-line*), e os usuários eram principalmente profissionais com uma formação em computação ou engenharia¹⁹. Ou seja, a distância entre o universo do discurso do *designer* do sistema e o dos usuários potenciais não era muito grande: falavam a mesma linguagem.

Hoje em dia, o cenário é bem diferente: os usuários interagem diretamente com sistemas *on-line*, e eles são mais expostos a uma variedade de *softwares* comerciais no seu dia-a-dia, o que faz com que eles se tornem mais discernentes, críticos e exigentes em frente dos *softwares*, além de serem mais independentes na instalação nos seus microcomputadores.

Os modelos tradicionais têm sido obrigados a incorporar estas novas características das comunidades de usuários, que sabem muito bem o que querem. Entretanto, não é suficiente simplesmente planejar/projetar sistemas que coloquem os usuários em uma posição central (os chamados *user-centred systems*) por meio de interfaces interativas e amigáveis. O argumento principal dos críticos em relação aos métodos tradicionais é que o grau de “usabilidade” (*usability*) do sistema final depende do grau de integração dos usuários ao próprio processo de *design* do sistema. Notava-se uma tendência dos *designers* que seguiam os métodos tradicionais a ignorar os chamados “fatores humanos” dos sistemas, uma negligência embutida nas próprias metodologias com suas predileções para linguagens formais e especificações abstratas. Conseqüências: os requisitos dos usuários mudam com freqüência à medida que apreciam como o sistema projetado irá influenciar o próprio caráter do seu trabalho (e, portanto, suas necessidades de informação); ressentimentos diante da possível ameaça que o novo sistema representa à segurança do seu emprego; subutilização, boicote ou até sabotagem do novo sistema.

AS ALTERNATIVAS

Neste contexto do reconhecimento geral das falhas e fragilidades das metodologias tradicionais, surgiram metodologias de *design* de sistemas que pressupõem como fundamental a participação dos usuários no processo de *design*: as chamadas metodologias de *participatory design*. Um modelo de comunicação que sustenta a maioria das metodologias de *design* participativo (DP) é o proposto pelos biólogos chilenos Maturana e Varela²⁰, no qual é dada ênfase ao que acontece com os receptores quando o emissor entrega a mensagem. Para eles, “a comunicação depende não do que é transmitido, mas do que acontece à pessoa que a recebe.” Ou seja, os receptores participam do ato de comunicação; eles se envolvem na atividade de comunicação com o emissor. Como afirmam Maturana e Varela: “A comunicação é criada por pessoas que interagem”²¹.

O que cabe ressaltar aqui é que para DP, os usuários e os *designers* são *ambos* enunciadores, e isto parece lógico quando levamos em conta que qualquer *design* de um sistema é baseado em dois domínios²²: o trabalho atual dos usuários e as opções tecnológicas (tanto do *hardware* e

software, quanto da organização do próprio trabalho). E, como dizem Kensing e Munk-Madsen, “as interações entre estes dois domínios leva a um terceiro: um novo (ou mudado) sistema computadorizado e mudanças no conteúdo e na organização do trabalho dos usuários”²³.

Estas interações acontecem, somente se houver oportunidades para a equipe do *design* e a comunidade de usuários aprenderem o máximo possível sobre seus respectivos domínios de trabalho. Os usuários deveriam ter chances de adquirir conhecimento das opções tecnológicas e organizacionais, ficando expostos a várias alternativas, pois, ao se tornarem cientes da série de possibilidades ofertadas, podem participar da tomada de decisões de um ponto de vista consciente. Por exemplo, Ehn²⁴ afirma que, no Projeto Utopia, na Noruega*, os futuros usuários e os *designers* do sistema que está sendo projetado fazem visitas em conjunto a laboratórios de pesquisa e a escritórios de vendedores, para tomar consciência da série de possibilidades tecnológicas disponíveis, o que constitui atividades importantes no começo do projeto. Da mesma forma, Greenbaum e Madsen²⁵ comentam como queriam integrar as visitas de vendedores no seu projeto nos EUA (*Participatory Design Project*) para que os usuários pudessem ganhar experiência direta (*hands-on experience*) com os vários sistemas. Entretanto, neste caso, estas pesquisadoras se defrontaram com as atitudes tradicionais dos vendedores, pois não lhes agradou a idéia de deixar que os usuários não-administrativos mexessem com os sistemas para adquirir esta experiência direta, o que revela, segundo essas autoras, que os vendedores tendem a ver apenas os gerentes como o mercado potencial para sistemas automatizados, ignorando o fato de que são eles que provavelmente menos o usarão no seu dia-a-dia, o que não é verdadeiro dos funcionários.

Cada uma das várias metodologias de DP tem suas próprias técnicas de promover a participação dos usuários no processo do *design*, que variam de *workshops* de crítica (criticando o sistema atual), fantasia (imaginando um cenário ideal) ou de contar histórias (de suas experiências com sistemas)²⁶ à reprojeção do departamento e do trabalho (*job and departmental re-design*) e a especificação dos requisitos de satisfação do trabalho, como é feito na metodologia de DP chamado ETHICS de Enid Mumford²⁷. O que é notável é que estes tipos de mecanismos de participação estão longe dos “contatos” feitos com os usuários nas metodologias tradicionais, nas quais os *designers* tendem a identificar as informações relevantes por meio de enquetes, relatórios, questionários, entrevistas, ou talvez por meio de experiências em laboratórios²⁸, ou via um usuário selecionado, ou um gerente que apresenta as reivindicações da comunidade inteira dos usuários. Ou seja, à medida que os usuários colaboram com os analistas por meio destes contatos, o *designer* está também absorvendo o que é visto como “o necessário” do domínio dos usuários. Entretanto, os problemas com tais metodologias já têm sido bem documentadas na literatura sobre psicologia, que aponta para o fato de que as pessoas não agem normalmente como o fazem em experiências de laboratório e nem mesmo como quando estão sendo filmados ou observados no seu lugar de trabalho, desempenhando suas tarefas rotineiras: o ser humano é autoconsciente demais sob tais condições²⁹. E adquirir noções de usuários por meio de enquetes, questionários e relatórios é evidentemente muito parcial, pois tais instrumentos são bastante deterministas no sentido de que as pressuposições e focos de interesse vão permear e determinar as perguntas feitas.

O que é talvez mais surpreendente é o conselho de Kensing e Munk-Madsen: o *designer* não deveria confiar completamente nos requisitos escritos e falados diretamente pelos usuários. Eles afirmam isto, não devido a algum tipo de desconfiança intrínseca do analista em relação aos usuários, mas devido à natureza de conhecimento. Cada um de nós tem um corpo de conhecimento *proposicional* - ou seja, conhecimento que pode facilmente ser expresso em linguagem natural e de uma maneira mais ou menos não ambígua (como, por exemplo, um fato

* Os países escandinavos são pioneiros na área de DP, devido às leis sindicais dos anos 70 que promovem o direito de os sindicatos participarem em decisões tecnológicas que afetariam diretamente o trabalho dos seus membros.

geográfico). Mas também possuímos um corpo substancial de conhecimento tácito, que tem suas muitas raízes no senso comum, em uma cultura específica e no conhecimento prático. Ehn³⁰ e sua equipe de *designers* encontraram muito deste tipo de conhecimento enquanto trabalhavam no projeto para um sistema para tipógrafos. Estes usuários conseguiram explicar aos *designers* fatos relacionados às ferramentas que usaram (i.e. conhecimento proposicional), mas não conseguiram explicitar o porquê da escolha de um determinado *layout* de uma página: como explicar em palavras estas qualidades sem mostrá-las com exemplos? Foi o filósofo Heidegger que primeiro levantou tais pontos na sua obra *Being and Time* (1962), na qual criticou a tradição filosófica ocidental por ter superestimado a relevância da contemplação descompromissada e a importância de representações da realidade na mente, ao passo que a importância de nossa visão prática e envolvida com o mundo foi relegada ao segundo plano³¹.

As implicações da existência desse corpo de conhecimento tácito para a equipe de *design* de um sistema são que esta não pode se colocar na posição de julgar a relevância de uma proposição (falada ou escrita) dos usuários, se não tiver uma experiência direta com aquilo que está sendo referido, o que quer dizer que o *designer* deveria experimentar o trabalho do usuário para que pudesse penetrar na dimensão tácita do conhecimento dos usuários. (O mesmo se aplica àqueles *designers*/analistas que já têm experiência com desenvolvimento de sistemas em organizações parecidas: as semelhanças não podem ser presumidas, ou seja, as particularidades de cada organização não podem ser ignoradas³².) As implicações para os usuários são que o uso de artefatos de *design* como maquetes, protótipos, bonecos (*mock-ups*) etc. do proposto sistema torna-se muito valioso para a comunicação efetiva. Ehn³³ enfatiza que o uso de tais artefatos concretos não ajuda no sentido cartesiano (ou seja, a maquete refletindo a realidade), mas estimula a imaginação dos usuários e dos *designers*; fazendo-os lembrar de casos paradigmáticos e evocando possibilidades futuras. A metodologia da confecção do protótipo em si não é sempre recomendada, pois fixa certas facetas do *design* muito cedo no projeto—facetas que são depois difíceis de abandonar (física ou mentalmente). Também, às vezes, desvia a atenção de perguntas cruciais como: “Precisamos mesmo de um sistema computadorizado?”³⁴ ou “Queremos um (novo) sistema computadorizado?”. Com um protótipo computadorizado na frente para avaliar, é improvável que tais perguntas surjam.

HARD VS. SOFT

As metodologias de *design* que envolvem os usuários são normalmente agrupadas sob a categoria de abordagens *soft* (uma metodologia participativa desenvolvida por Checkland na década 80 chama-se *Soft Systems Methodology*), em contraposição às metodologias *hard* ou tradicionais. Esta denominação evoca a idéia de que as metodologias *hard* concentram-se mais nas soluções técnicas, ao passo que as *soft* tentam abordar as questões mais complexas relacionadas à estrutura de poder da organização, e a fatores psicológicos e políticos da organização - ou seja, fatores fundamentalmente humanos. Entretanto, não é necessariamente o caso de se adotar uma categoria com a exclusão da outra; depende muito do tipo de problema sendo abordado e, claro, do tipo de usuários que utilizará o sistema. Já mencionamos que, nas primeiras décadas da computadorização de SIs, os usuários tendiam a ser os próprios programadores ou analistas, informatizando principalmente as atividades funcionais da organização (e.g. o pagamento de salários, a checagem de estoque). Ao longo das décadas, testemunhamos a informatização de atividades menos estruturadas e rotineiras, com o concomitante surgimento de Sistemas de Informação Gerencial (MISs), Sistemas de apoio a tomada de decisões (DSSs) e Sistemas especialistas (ESs). Podemos assim deduzir que os requisitos de tais sistemas são menos fáceis de delinear, exigindo, portanto, mais participação dos usuários no seu desenvolvimento. Entretanto, não deveríamos nos iludir sobre o fato de atividades rotineiras e estruturadas acarretarem requisitos bem conhecidos: temos sempre que nos lembrar de que os requisitos podem metamorfosear-se ao longo do processo do *design* e que os grupos de usuários não são homogêneos. A própria constituição dos grupos de usuários de um

sistema computadorizado para bibliotecas e centros de informação exemplifica este fato: temos não apenas os usuários que são os bibliotecários ou funcionários, mas também os usuários propriamente ditos da biblioteca - ou seja, o público-alvo, que em si é um grupo muito heterogêneo.

IMPACTO ORGANIZACIONAL DE SISTEMAS COMPUTADORIZADOS E DP

Não é objetivo deste trabalho delinear as principais vertentes da área de “impacto organizacional” de sistemas automatizados de informação. Todavia, avaliaremos a questão do ponto de vista do DP, principalmente com relação à definição dos requisitos dos usuários, pois o que muitas vezes surpreende na sua participação ativa no processo de *design* é que as metas da organização nem sempre coincidem com as deles. A organização pode querer aumentar a velocidade da produção por meio do sistema enquanto o usuário pode ter como prioridade o melhoramento da qualidade de sua condição de trabalho, desempenhando suas tarefas com mais precisão, profundidade e eficiência. A participação dos usuários no *design* pode trazer à luz tais tipos de “inconsistências”.

O fato de existirem grupos de usuários bem diferentes dentro da mesma organização, cada um com aspirações “políticas” ou profissionais distintas, pode também ser esclarecido durante o DP. Portanto, o contato com apenas alguns usuários “chaves” selecionados pela gerência, como nas metodologias tradicionais, pode acarretar distorções (não intencionais) da imagem da comunidade dos usuários da organização, resultando em uma possível insatisfação com o sistema produzido por parte dos usuários que não participaram.

Tudo isto nos lembra a característica básica de qualquer tecnologia, que é determinada pelo ambiente social no qual é introduzida, e não o inverso. Se subestimamos o papel dos fatores sociais na moldagem da tecnologia, ou se ignoramos a variedade de contextos sociais que mediam seu uso³⁵, caímos na retórica falaciosa de determinismo tecnológico, que enfatiza o impacto da tecnologia em nossas vidas, como se fosse um impacto uniforme, monolítico e cego às várias realidades sociais nas quais é projetado e introduzido. A realidade organizacional na qual o sistema é introduzido determinará o aspecto do seu impacto. Isto implica que o sistema somente será bem sucedido se combinar com os fatores sócio-políticos da organização, e não meramente com as tarefas/funções para as quais foi desenvolvido. É necessário ter consciência das mudanças estruturais na organização que um novo sistema possa catalisar e/ou necessitar. Algumas pesquisas mostram que as organizações, embarcando em projetos de computadorização, não estão fazendo isto e, como consequência (e paradoxalmente, em nossos dias de preços baixos de *software* e *hardware*), os custos do que Bjorg-Andersen denota *orgware* estão aumentando³⁶.

DP É INEQUIVOCAMENTE UMA ABORDAGEM MELHOR?

Por que uma empresa escolheria uma metodologia com a participação dos usuários, quando já foi comprovado que estas metodologias de *design*, como concentram os esforços nas fases do começo do ciclo da vida (i.e. a determinação dos requisitos)³⁷, são mais caras e prolongam o processo do *design*³⁸? Mais ainda: o fato de que os usuários participam do processo não necessariamente garantirá que o problema real será determinado, ou que o sistema será necessariamente melhor. Só porque os usuários são especialistas nas suas tarefas atuais não significa, inequivocamente, que manterão esse mesmo nível de especialização sobre as tarefas no futuro, principalmente porque as novas tecnologias mudarão a própria natureza delas. Um outro problema levantado³⁹ é que não é sempre que a gerência da organização concorda com a idéia da participação dos seus funcionários no processo de *design*, o que poderia ser contraprodutivo ao longo do tempo. Ele nota que, se ainda não existe uma prática de os funcionários tomarem decisões na instituição, então talvez a hora de iniciá-los nesta atividade não seja durante o planejamento de um novo sistema, dada as pressões de tempo, verba etc.

Anderson⁴⁰ levanta o que considera a questão crítica da participação do usuário: a falta do seu

conhecimento técnico. Sem dúvida, esta é uma área crítica de qualquer projeto de informatização em que a equipe de *design* teria potencial ilimitado para explorar sua superioridade nos assuntos técnicos. Bjorn-Andersen e Markus já levantaram esta questão no contexto de uma análise do poder do analista/*designer* em um projeto com possibilidade de ser altamente técnica (como a automatização de uma biblioteca): nas metodologias tradicionais, o *designer* pode (inconscientemente ou conscientemente) alterar o comportamento ou até os requisitos dos usuários, ao fazer certas perguntas, e ao não fazer outras, por exemplo. A ótica da interpretação da situação de que o *designer* dispõe ao iniciar o projeto vai, sem dúvida, influenciar o tipo de questões feitas aos usuários. Acontece geralmente que estas perguntas não abordam questões de satisfação no emprego, ou do próprio *design* ideal das tarefas, ou até questões mais amplas (e “psicológicas”) relacionadas à organização e à estrutura do trabalho. Uma outra faceta deste “jogo de poder” pode ser encontrada no fato de que o analista/*designer* é influenciado pela disponibilidade de certo *software* e outros recursos, o que pode contribuir para uma alteração das necessidades dos usuários.

Poderíamos aqui afirmar que, ao problematizar a falta de conhecimento técnico dos usuários, Andersen (como muitos profissionais da linha tradicional) supervaloriza a solução computacional para qualquer sistema. (A crítica de tal escola de pensamento é encapsulada na seguinte pergunta irônica: *Technology is the answer, but what is the question?**) Mumford⁴¹ afirma que sua metodologia, ETHICS, não necessariamente visa a uma solução computacional para o “problema”, mas enfatiza a obtenção de um equilíbrio entre os aspectos sociais e técnicos do sistema. Mumford reforça que os especialistas em qualquer *design* de um novo sistema são os usuários, e não o *designer*, pois são eles que conhecem detalhadamente as tarefas do dia-a-dia do seu trabalho - ou seja, a base do sistema existe na cabeça deles. Podemos afirmar, então, que este conhecimento especialista compensa qualquer deficiência técnica que os usuários porventura tenham. Visto por esta ótica, é o *designer* que deveria participar no projeto dos usuários, e não vice-versa. Os custos extras decorridos da participação dos usuários são discutivelmente recuperados posteriormente na aceitação e conseqüente uso efetivo do sistema pelos usuários, e isto particularmente ocorre quando o sistema vai ser usado por cem ou mais usuários (como é o caso de sistemas em bibliotecas)⁴².

Isto nos leva à questão da ética de participação. Para Mumford, a participação dos usuários é ética, pois as pessoas têm um direito moral de controlar seus destinos, e este controle deveria ser aplicado tanto no lugar do trabalho, quanto em outras situações da vida. Um outro argumento pró-DP é que este motiva os usuários, acarretando um aumento na produtividade e eficiência do sistema eventual. Como nota Booth⁴³, o alvo final das abordagens escandinavas é aumentar a rentabilidade por meio deste melhoramento da qualidade e produtividade do trabalho (via DP). Entretanto, um dos pioneiros do DP escandinavo, Niels Bjorn-Andersen, também nos adverte contra o uso do DP como um meio de cooptar e manipular a mão-de-obra, obrigando-a a aceitar mudanças de trabalho indesejáveis: o DP torna-se um meio de neutralizar a resistência potencial ao sistema, pois os usuários são corresponsáveis por ele⁴⁴. O alvo final do DP é melhorar a qualidade de vida dos funcionários na organização por meio do enriquecimento de seu trabalho, usando a tecnologia para contribuir na realização deste alvo, e não usando-a só por usá-la⁴⁵.

ESTUDO DE CASO DE DP

Nos meados da década de 80, iniciou-se um projeto para desenvolver um novo sistema computadorizado para a rede das bibliotecas públicas da cidade de Sheffield, no norte da Inglaterra (que, até então, tinha um sistema de processamento em lote, utilizando o computador de grande porte o – *mainframe* – da Prefeitura)⁴⁶. Entretanto, a gerência não conseguiu finalizar as especificações para a “customização” de um “pacote” de um sistema integrado no mercado, principalmente devido à resistência por parte dos usuários (i.e. os bibliotecários) a tanta

* “A tecnologia é a resposta, mas o que é a questão?”

computadorização. Esta resistência foi devida, por sua vez, às péssimas experiências com o sistema em lote antiquado com que tiveram que lidar. Daí, uma equipe de pesquisadores da universidade da cidade propôs técnicas de desenvolvimento do novo sistema que contariam com a participação dos bibliotecários. Esta proposta foi aceita pela gerência da rede, e, ao longo dos três anos seguintes, foram realizados “círculos de estudos” (durante os quais os participantes – membros do *staff* e pesquisadores – delinearam não apenas seus requisitos, mas também definiram uma rede de apoio entre os bibliotecários, bem como ganharam acesso a novas idéias e informações das opções tecnológicas), paralelamente a reuniões de uma equipe de *design* cujos participantes provinham de ambos o *staff* e a gerência das bibliotecas. Digna de nota foi a equipe de *design* que experimentou mais dificuldades, pois os participantes do *staff* (i.e. os bibliotecários) sentiram-se isolados, por estarem em minoria (o resto da equipe de *design* sendo da gerência). Por causa disso, a constituição da equipe de *design* foi alterada: mais bibliotecários foram incluídos, o que piorou a situação temporariamente devido à resistência, agora por parte da gerência, ao novo *status* fortalecido dos bibliotecários.

Os aspectos mais relevantes que surgiram da participação dos bibliotecários neste projeto (que se chamou Human Centred Office Systems Project - HCOSP) foram os seguintes:

- foram identificados requisitos pertinentes baseados diretamente no contato que os bibliotecários tinham com usuários dos serviços da biblioteca-requisitos que os pesquisadores alegam que teriam sido ignorados;
- os bibliotecários foram além das especificações meramente técnicas para incluir questões significativas de treinamento, igualdade de oportunidade e o próprio estruturamento (*design*) do seu trabalho;
- os esterótipos entrincheirados de bibliotecárias passivas foram desmistificados, pois os participantes (99% de mulheres) contribuíram com sugestões práticas e bem informadas sobre o *software*.

Este caso de DP representa um entre muitos que deram certo. Hull⁴⁷ também relata um caso de funcionários administrativos de uma firma na Inglaterra que aprenderam o básico de programação para que pudessem construir as partes de um sistema que afetaria diretamente o desempenho do seu trabalho.

CONCLUSÕES

Podemos interpretar as críticas específicas dirigidas às metodologias tradicionais de análise de sistemas e do *design* de sistemas de informação automatizados e a conseqüente promoção de metodologias “alternativas” como fazendo parte da agenda pós-moderna de questionar os paradigmas existentes. No âmbito da ciência da informação, já estamos testemunhando uma mudança do paradigma da informação centrada no dado (*data-centred paradigm*) para o centrado no usuário (*user-centred paradigm*). Quanto ao desenvolvimento de sistemas novos, a ignorância da possibilidade de os usuários participarem e contribuírem com este processo muito provavelmente resultará em usuários insatisfeitos e, por conseqüência, em baixa produtividade e qualidade de trabalho. Permitindo-se que a visão dos usuários permeie e guie o processo de desenvolvimento, evitar-se-iam cenários do tipo: este aspecto do problema não pode ser abordado, pois é muito complexo e, além do mais, não existe uma solução técnica para ele. Esta atitude, nota Cooley⁴⁸, é típica das metodologias tradicionais que abstraem cientificamente de tal maneira que as questões complexas, humanas, somem, como por encanto. Se nos lembrarmos do fato de que qualquer sistema é uma entidade sociotécnica (como na abordagem de Mumford),

não apenas com seu lado técnico, mas também com seu lado social (humano) , então, o *design* participativo parece ser o caminho mais natural e holístico. As justificativas para DP encontram-se na seguinte colocação de Cooley⁴⁹, com a qual encerramos:

“Decisions we make in respect of technological developments during the next five or ten years will have profound effects upon the way our society develops; the manner in which human beings relate to machines and to each other; and the relationship between human beings, their built environment and nature itself.” *

* “As decisões que tomarmos com respeito às novidades tecnológicas durante os próximos cinco ou dez anos terão efeitos profundos na maneira com o qual nossa sociedade evolui, a maneira na qual os seres humanos se relacionam com as máquinas e um com o outro, e o relacionamento entre os seres humanos, seu ambiente contruído e com a própria natureza”.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AVISON, D.E. *Information systems development: a database approach*. Oxford, Blackwell Scientific Publications, 1992, 369p. p.13-14.
2. MEY, E.S. Bibliotecários e analistas de sistemas: a convivência necessária. *R. Biblioteconomia. Brasília*, v.16, n,1, p.75-81. jan./jun. 1988.
3. FLYNN, D.J. *Information systems requirements: determination & analysis*. London, McGraw-Hill, 1992. 422p. p.97-108.
4. *Ibidem* p.105.
5. LAYZELL, P.; LOUCOPOULOS, P. *Systems analysis and development*. Chartwell-Bratt, 1989. 273p. p.18.
6. FLYNN, D. J. *op.cit.* p.106.
7. FLYNN, D. J. *op.cit.* p.12.
8. *Ibidem*.
9. *Ibidem* p. 292-3.
10. *Ibidem* p.12-5 & p. 110-112.
11. PARTRIDGE, D. Will AI lead to a super software crisis? *In: GILL, K.S. (ed.), Artificial intelligence for society*. London, John Wiley, 1986, p.32.
12. KENSING, F. & MUNK-MADSEN, A. PD: structure in the toolbox. *Communications of the ACM*, v. 36, n.4, p.78- 85.
13. GRUNDIN, J. Obstacles to participatory design in large product development organizations. *In: SCHULER, D., NAMIOKA, A. Participatory design: principles and practices*. London, LEA Publishers, 1993, p.99-119. p.113.
14. KENSING, F. & MUNK-MADSEN, A. *op.cit.* p.78.
15. *Ibidem* p. 79.

16. FLOYD, C. Outline of a paradigm change in software engineering. *ACM SIGSOFT Software Engineering Notes*, v.13, n.2, p.25-38. 1988.
17. EHN, P. Scandinavian design: on participation and skill. *In: SCHULER, D., NAMIOKA, A. op.cit.* p.41-77. p.62.
18. CHAUI, M. Filosofia Moderna. *In: CHAUI, M. et.al., Primeira filosofia, lições introdutórias.* São Paulo, Editora brasiliense, 1987. p.70-2.
19. GRUNDIN, J. *op.cit.* p.99-119.
20. KENSING, F. & MUNK-MADSEN, A. *op.cit.* p.79.
21. *Ibidem.*
22. *Ibidem.*
23. *Ibidem.*
24. EHN, P. *op.cit.* p.57-8.
25. GREENBAUM, J., MADSEN, K.H. *In: SCHULER, D., NAMIOKA, A. op.cit.* p.289 - 298. p.295.
26. *Ibidem* p.290-6.
27. MUMFORD, E. The participation of users in systems design: an account of the origin, evolution and use of the ETHICS method. *In: SCHULER, D., NAMIOKA, A. op.cit.* p.257-270.
28. GRUNDIN, J. *op.cit.* p.99 & p.105.
29. BOOTH, P. *An introduction to human-computer interaction.* London, Lawrence Erlbaum, 1991.268p. p.126
30. EHN, P. *op.cit.* p.67-8.
31. WINOGRAD, T., FLORES, F. *Understanding computers and cognition: a new foundation for design.* Reading, Addison Wesley, 1987. 207p.. p.27-37.
32. KENSING, F. & MUNK-MADSEN, A. *op.cit.* p.80.
33. EHN, P. *op.cit.* p.66-7.
34. KENSING, F. & MUNK-MADSEN, A. *op.cit.* p.84.
35. TURKLE, S. Computers and the human spirit. *In: FINNEGAN, R., SALAMAN, G., THOMPSON, K. (eds.), Information Technology: Social Issues - A Reader.* London, Hodder & Stoughton, 1987. p.264.
36. BJORG ANDERSON, N. *apud* BOOTH, P. *op.cit.* p.190.
37. FLYNN, D. J. *op.cit.* p.306.

38. MANTEI, M., TEOREY, T.J. Cost/benefit analysis for incorporating human factors in the software lifecycle. *Communications of the ACM*. v.31, n.4, : 428-439, April. 1988.
39. ANDERSON, E. E. Managerial considerations in participative design of MIS/DSS. *Information & Management*. v.9, p.201-207, 1985.
40. *Ibidem* p.202.
41. MUMFORD, M. *op.cit.* p.259.
42. MANTEI, M.,TEOREY, T.J. *op.cit.* p.437.
43. BOOTH, P. *op.cit.* p.181.
44. MUMFORD, M. *op.cit.* p.259.
45. BOOTH, P. *op.cit.* p.181.
46. CLEMENT, A. Computing at work: empowering action by 'low-level users'. *Communications of the ACM*, v.37, n.1, p.53-63, January 1994.
47. HULL, R. *In praise of WIMPS: a social history of computer programming*. Hebden Bridge, Alice Publications, 1992. 145p. p.56.
48. COOLEY, M. Human centred systems: an urgent problem for systems designers. *AI & Society*, v.1, p 37-46, 1987.
49. *Ibidem* p.38.

The participation of users in the development of information systems

Abstract

The traditional approaches to the design of computerized information systems are grounded in certain technical presuppositions that are usually unable to take on board social, political and psychological aspects of the organization in which the new system will be implemented. The costs of ignoring these dimensions can sometimes be high, since dissatisfied users can resist use of the system, or use it in ways not foreseen by the original design of the system. The limitations of the traditional, so-called "hard" approaches to design have been mitigated by the emergence and adoption of radically different, "soft" methodologies, which are based on the central idea that only the continued participation of the users in the system's development can avoid future problems, given that only the users - and not the systems analyst - possess the specialist knowledge pertaining to the nature of their work that is necessary to construct a high quality, and therefore productive, system.

Keyword

Information systems development user involvement

**Ariadne Chloë Furnival (email: chloe@power.ufscar.br)
 Professora Assistente do Núcleo de Biblioteconomia e Ciência da Informação da Universidade Federal de São Carlos (UFSCar). Bacheler em Comparative American Studies e Master of Arts (M.A.) em Literatura, ambos pela Universidade de Warwick, Inglaterra. Master of Science (M.Sc.) em Computação pelo Instituto de Ciência e Tecnologia da Universidade de Manchester (UMIST), Inglaterra.**

