

# Participação dos usuários no desenvolvimento de sistemas de informação: uma reflexão sob a óptica positivista e interpretativista

Mônica Ximenes

Rodrigo Oliveira

Marcilio Ferreira

Décio Fonseca

Centro Federal de Educação Tecnológica de Bambuí - CEFET

## RESUMO

*Nos dias atuais, é crescente o fato do desenvolvimento de sistemas de informação procurar considerar o contexto social e organizacional. O sucesso de um sistema depende do quanto os requisitos dos usuários são satisfeitos, derivados ou limitados por esse contexto. Este trabalho busca identificar os aspectos relevantes de dois diferentes tipos de abordagens na construção de sistemas de informação: positivista e interacionista. E lança o olhar para as atividades relacionadas à gestão de requisitos, que representam um papel fundamental no processo de construção de um sistema, especialmente por se constituírem o alicerce para as fases subseqüentes do desenvolvimento. Para ilustrar cada tipo de abordagem são apresentadas suas respectivas características, sempre com ênfase na importância da participação do usuário no processo de desenvolvimento dos sistemas. Ao final é apresentada uma análise dos pontos fracos e fortes de cada abordagem à luz das contribuições dos usuários.*

Palavras-chave: Sistemas de informação. Desenvolvimento. Participação de usuários.

## 1. INTRODUÇÃO

As tecnologias da informação (TI) têm participado do cotidiano das organizações e das pessoas que trabalham com sistemas de informação (SI) nas empresas e têm sido consideradas protagonistas de mudanças organizacionais. Conforme relata Ramos (2000), as TI são geralmente usadas para auxiliar a ação humana nas organizações, para permitir novas formas de realizar as atividades de negócio ou para implementar novas atividades. As aplicações de TI se constituem parte integrante da realidade de trabalho, reforçando a sua estrutura, as práticas de trabalho, as relações entre os atores organizacionais, e os significados partilhados, e permitindo o controle e a coordenação das várias atividades.

Ainda conforme Ramos (2000), existem relatos de que muitas vezes os sistemas de informação não atendem as necessidades dos usuários. Um problema que está muitas vezes relacionado com a adoção de aplicações que não satisfazem os requisitos para elas definidos, com custos de desenvolvimento que excedem os benefícios esperados da sua utilização, com a constatação de que a aplicação não atende as necessidades, expectativas e/ou interesses das entidades (organizações e grupos) afetadas pela sua adoção.

Para Furnival (1995), os casos de projetos para sistemas de informação automatizados que fracassam são muitos, em geral resultando em não-uso, sub-uso, 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. Segundo o autor, o que é difícil de entender é como tais sistemas chegam ao ponto de ser implementados.

De fato, nos dias atuais, o desenvolvimento de sistemas de informação tem procurado considerar o contexto social e organizacional. E o sucesso ou fracasso desse processo depende do quanto os requisitos dos usuários são satisfeitos, derivados ou limitados por esse contexto. Para entender e captar adequadamente esses requisitos algumas metodologias são utilizadas.

As tradicionais, de cunho positivista, alegam que os requisitos podem ser clara e precisamente especificados lingüisticamente desde o começo do projeto. Como observa Partridge (1986) *apud* Furnival (1995), estas metodologias definem a maneira ‘correta’ de desenvolver sistemas como aquela que primeiro especifica completamente o problema. Existem dúvidas de que os requisitos dos usuários podem mesmo ser determinados (e fixados) desde o começo do projeto, além da dificuldade de compreensão da comunicação formal baseada em diagramas e documentos.

Segundo Furnival (1995), as metodologias de cunho interacionista, como o desenho participativo, têm suas próprias técnicas de promover a participação dos usuários no processo do *design*. Essas técnicas variam de *workshops* de crítica (criticando o sistema atual), fantasia (imaginando um cenário ideal) ou de contar histórias (de suas experiências com sistemas) à re-projeção do departamento e do trabalho (*job and departmental re-design*) e à especificação dos requisitos de satisfação do trabalho. 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 laboratórios, 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.

De acordo com Ferreira e Lima (2005), a resistência dos usuários ao sistema é apenas a ponta de um *iceberg* cuja base foi construída a partir de premissas falsas ou incompletas. Ou seja, o verdadeiro problema se encontra, muitas vezes, na fase inicial do projeto onde são levantadas informações sobre as necessidades do sistema (quais funcionalidades o sistema deverá possuir para alcançar determinados fins), o negócio do cliente, isto é, o processo produtivo a ser informatizado, e o saber operatório dos operadores no controle e supervisão do processo. É a qualidade destas informações que determinará, em grande parte, o sucesso do software no que se refere à sua aceitação e eficiência. A área de Engenharia de Requisitos dedica-se ao levantamento eficiente dessas necessidades do cliente.

Todos estes potenciais problemas, decorrentes da adoção de aplicações das TI, realçam a importância de definir aplicações adequadas a realidades de trabalho percebidas como motivadoras e geradoras de sentimentos de satisfação no trabalho. A correta definição dos requisitos facilita o desenvolvimento de aplicações que se ajustam à ação humana, aumentando a probabilidade de satisfazer os seus usuários e contribuir para uma maior eficácia do trabalho (RAMOS, 2000).

Ao analisar a produção científica da década de 1990 na área de Sistemas de Informação, Hoppen (1998) mostra que houve um predomínio de estudos sobre o desenvolvimento e uso de sistemas de informação. Porém, apesar desta representatividade histórica, apenas 7,1% dos trabalhos sobre ‘uso de SI’ foram destinados a investigações sobre os usuários. Em se tratando do tópico ‘desenvolvimento de SI’, os usuários e sua participação não constituíram sequer uma subcategoria de análise. Tal fato denota preocupações, já que os usuários são diretamente afetados com a implementação de novos sistemas e o seu uso adequado é a base para a melhoria da eficiência e dos resultados organizacionais.

O estudo sobre a participação dos usuários no desenvolvimento de sistemas de informação ainda é pouco explorado pela literatura nacional em Administração. Essa constatação está pautada na dificuldade de encontrar artigos sobre esse tema no principal evento da área (Encontro da Associação Nacional dos Programas de Pós-graduação e

Pesquisa em Administração - EnANPAD), entre 1998 e 2006, o que reforça o interesse pelo estudo.

Para Vasconcelos *et al.* (2003), a maneira com que os indivíduos se apropriam da ferramenta e atribuem-na sentido é que permitirá o seu uso dentro de uma função ora automatizadora, promovendo apenas o aprendizado de circuito simples, ou seja, apenas com maior eficiência da operação/procedimento; ora dentro de um paradigma mais amplo, de informatização, aprendizagem de circuito duplo e inovação, sendo este último uma abordagem de interação sustentável e produtiva entre pessoa e tecnologia.

Além disso, existem paradoxos inerentes ao discurso empresarial quando se utilizam de modelos (Pós) Industriais para justificar a implementação de novas tecnologias e sistemas de informação. Nesse sentido, a tecnologia não induz necessariamente um tipo de percepção específica nos diferentes grupos de atores sociais. Assim, considera-se que estes reagem à tecnologia a partir do sentido que atribuem à mesma, o que depende de suas experiências passadas e fatores ligados à sua identidade/convivência social (VASCONCELOS, *et al.* 2001).

## **2. DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO: FOCO EM ENGENHARIA DE REQUISITOS**

De acordo com Rezende (2005), os sistemas de informação, independentemente de seu nível ou classificação, objetivam auxiliar os processos de tomada de decisões na organização. Se os sistemas de informação não se propuserem a atender a esse objetivo, sua existência não será significativa para a organização.

Um sistema de informação que utiliza recursos da tecnologia da informação pode ter um ciclo de vida curto, de no máximo cinco anos, quando não sofre implementações. O ciclo de vida natural abrange as fases: concepção ou criação; construção ou programação; implantação (disponibilização); implementações (pequenos ajustes ou melhorias); maturidade (utilização plena do sistema); declínio, manutenção; morte ou descontinuidade. Quando as três primeiras fases são elaboradas de forma errada, a morte do sistema de informação é acelerada, principalmente se o sistema focar a gestão estratégica da organização (REZENDE, 2005).

A ênfase desse trabalho está na fase de concepção ou criação, que envolve aspectos tratados especialmente pela Engenharia de Requisitos. De acordo com Sommerville (2003), a engenharia de requisitos é um processo que envolve todas as atividades exigidas para criar e manter o documento de requisitos nos sistemas. Existem quatro atividades genéricas de processo de engenharia de requisitos que são de alto nível, ou seja, o estudo da viabilidade do sistema, a obtenção e a análise dos requisitos, a especificação de requisitos e sua documentação e, finalmente, a validação desses requisitos.

Algumas pessoas consideram a engenharia de requisitos como o processo de aplicação de um método estruturado, como a análise orientada a objetos. No entanto, embora os métodos estruturados tenham um papel a desempenhar no processo de engenharia de requisitos, existem muito mais aspectos na engenharia de requisitos do que os que são abordados por esses métodos. Eles não fornecem apoio efetivo aos estágios iniciais do processo de engenharia de requisitos, como a obtenção de requisitos (SOMMERVILLE, 2003).

Para Ramos (2000), tem vindo a designar-se por Engenharia de Requisitos (ER) a área científica que aborda a problemática em torno da definição de requisitos para aplicações das TI, no que se refere à configuração do equipamento, funcionalidade e informação

disponibilizadas e interação com os utilizadores, e desenvolve as técnicas e os métodos adequados à definição e gestão de requisitos que estejam de acordo com determinados critérios de qualidade.

Ainda de acordo com Ramos (2000), a especificação dos requisitos num documento formal implica a tradução dos requisitos diretamente implementáveis numa tecnologia. Esta tarefa possui um carácter instrumental na medida em que o problema se traduz na definição de um conjunto de regras que permita essa tradução de acordo com determinados critérios de qualidade técnica e linguística. Já a definição dos requisitos dos usuários assume um carácter mais complexo. Esta complexidade resulta essencialmente da complexidade das interações sociais da realidade de trabalho onde a aplicação vai ser integrada e de alterações ambientais que, de uma forma mais ou menos regular, coloca novos desafios, oportunidades e problemas com os quais os atores sociais devem ser capazes de lidar. No entanto, nem sempre aqueles atores organizacionais têm uma idéia clara de quais são aqueles desafios, oportunidades e problemas, ou compreendem a totalidade das interações sociais e seu impacto na organização. Por outro lado, apresentam muitas vezes percepções diferentes e até antagônicas sobre a realidade de trabalho, ou não têm qualquer idéia de como as tecnologias de informação podem ser usadas para atingir os objetivos de negócio.

O resultado final do processo é um documento contendo os requisitos da aplicação, o qual servirá de base ao seu desenvolvimento ou aquisição, e permitirá aos utilizadores e outras entidades interessadas em verificar se ela satisfaz as necessidades, expectativas e interesses que motivaram a sua adoção. Este documento pode funcionar como um contrato legal entre os fornecedores da aplicação e as entidades nela interessadas. Para além dos requisitos da aplicação, este documento deverá conter uma descrição da nova realidade de trabalho, a qual integra a aplicação. Desta forma, o documento também representa um compromisso explícito de quem colaborou na sua criação para com um conjunto de novos conceitos, significados e práticas que serve de base à nova realidade de trabalho e que a aplicação ajuda a institucionalizar (RAMOS, 2000).

Conforme ressalta Vianna (2004), a visão de Sommerville (2003) sintetiza a Engenharia de Requisitos como o conjunto das seguintes atividades:

- **Elicitação de Requisitos:** Compreende o conjunto de atividades que a equipe de desenvolvimento emprega para elicitar ou descobrir as solicitações dos usuários, determinando as reais necessidades por trás das solicitações. É quando todas as necessidades, expectativas e recursos esperados do sistema serão levantados pelo Analista de Requisitos junto a cada *stakeholder*. Esta atividade normalmente requer uma forte comunicação entre o Analista e os *stakeholders*. Esta interação pode dar-se através dos mais diversos meios, como entrevistas, dinâmicas participativas, questionários, entre outros. Os requisitos devem ser levantados pela equipe do projeto, em conjunto com representantes do cliente, usuários-chave e outros especialistas da área de aplicação.
- **Análise e Negociação de Requisitos:** Nesta etapa está incluída a análise de todas as necessidades e expectativas levantadas na atividade anterior, que agora serão compiladas e ponderadas pelo Analista de Requisitos. Através da utilização de critérios de análise, como exequibilidade, prioridade, estabilidade ou benefício do requisito, ter-se-á como resultado a definição dos requisitos que fazem parte do escopo do sistema, e que devem ser devidamente registrados e continuados no processo de desenvolvimento. A determinação de quais requisitos serão realmente tratados no desenvolvimento deve ser estabelecida por meio de um consenso entre a equipe de desenvolvimento e os *stakeholders*, como consequência da negociação;

- **Documentação de Requisitos:** Compreende a atividade de especificação dos requisitos do sistema, ou seja, a “tradução” dos desejos e necessidades dos *stakeholders* para uma forma mais estruturada, que possa ser compreendida pelos demais participantes da equipe de desenvolvimento. Como o propósito da Engenharia de Requisitos é estabelecer um entendimento comum entre os usuários e a equipe de desenvolvimento sobre os requisitos do sistema a ser construído, os requisitos de software devem ser documentados. Tal entendimento comum serve como base para o estabelecimento de um acordo entre os usuários e a equipe de desenvolvimento, e como tal, a especificação é o documento central do processo, que orientará as fases subsequentes do desenvolvimento;
- **Validação de Requisitos:** Através desta atividade os requisitos previamente documentados ou especificados são validados pelos *stakeholders*, sob o ponto de vista de sua consistência e completude. O resultado perseguido é a aprovação dos requisitos pelos *stakeholders*, sinalizando sua continuidade dentro do processo de desenvolvimento, porém pode-se ter o eventual retrabalho de determinados requisitos.

De acordo com Espindola *et al* (2004), apesar do aparente fluxo entre as atividades acima descritas, não existe uma fronteira explícita entre elas. De fato, na prática existe muita sobreposição e interação entre uma atividade e outra. Os autores ressaltam ainda os tipos de entradas importantes para o processo, tais como: descrições do que os *stakeholders* necessitam para suportar suas atividades, informações a respeito do sistema que será implantado, padrões vigentes na organização a respeito de práticas de desenvolvimento de sistemas, regulamentos externos e informações gerais sobre o domínio de aplicação.

### 3. PERSPECTIVA POSITIVISTA E INTERPRETATIVISTA NO DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO

O desenvolvimento de SI, como qualquer outra atividade humana, necessariamente envolve princípios ou suposições explícitas ou implícitas. Dependendo dos princípios adotados, diferentes perspectivas podem ser identificadas e cada uma conduz a diferentes sistemas como resultado. Vale notar que diferentes filosofias podem levar a opções radicalmente diferentes em termos de características de projeto, estratégias de implementação, satisfação do usuário e uso do sistema (POZZEBON e FREITAS, 1997).

A inserção de sistemas de informação nas organizações pode ser realizada de diversas maneiras. No entanto, duas abordagens servem de base para o desenvolvimento e implementação de sistemas de informação (SI): positivista e interpretativista. A primeira é considerada tecnológica, sob a perspectiva *hard*, na qual a tecnologia é priorizada no processo de desenvolvimento dos sistemas. A segunda abordagem está relacionada a uma perspectiva *soft*, que incorpora características gerenciais no processo de desenvolvimento e implementação de SI's.

Segundo Reinhard (2005), autores como Michel Crozier estudaram as contradições entre os requisitos de lógica “radicalmente” consistente entre todos os níveis da organização e a realidade das grandes empresas, nas quais conflitos e agendas próprias convivem com uma estrutura apenas formalmente consistente. A transparência de informações que os novos sistemas oferecem explicita estas contradições e cria situações de conflito que precisam ser antecipadas e resolvidas, sob pena de inviabilizar a organização ou levar à rejeição (ou deturpação) do sistema implantado. Estas considerações levaram à constatação de que pode haver uma diferença entre o que o sistema permite fazer e o que convém que ele faça.

Estes conhecimentos sobre o comportamento organizacional passaram a orientar as ações dos analistas de sistemas responsáveis pelo projeto de implantação de sistemas de

informações e tinham por objetivo reduzir a elevada taxa de insucesso na implantação (REINHARD, 2005)

De acordo com Vianna (2004), a tarefa de implantação de um novo sistema de informação em uma organização é uma tarefa com um elevado nível de complexidade, e o seu sucesso depende fortemente dos requisitos do software. O Quadro 1 apresenta uma comparação entre a abordagem positivista e a interpretativista e oferece subsídios para uma análise acerca das diferenças entre as perspectivas *hard* e *soft* nas seções que a sucedem, com um olhar voltado para o desenvolvimento de sistemas de informação.

Quadro 1. Comparação entre Positivismo e Interpretativismo (Fonte: Weber, 2004).

<b>Suposições Metateóricas</b>	<b>Positivismo</b>	<b>Interpretativismo</b>
Ontologia	Pesquisador e realidade são separados	Pesquisador e realidade são inseparáveis
Epistemologia	Realidade objetiva existe além da mente humana	Conhecimento construído reflete metas, cultura, experiência, história
Objeto de Pesquisa	Objeto de pesquisa tem qualidades inerentes que existem independentemente do pesquisador	Objeto da pesquisa é interpretado à luz de estrutura de significado da experiência vivida pelo próprio pesquisador
Método	Estatísticas, Análise de conteúdo	Hermenêutica, estudo de caso, fenomenologia etc
Teoria da Verdade	Mapeamento um-a-um entre declarações do pesquisador e realidade	Verdade como conteúdo intencional: interpretações do objeto de pesquisa
Validade	Dados verdadeiros medem a realidade	Conhecimento construído
Confiabilidade	Replicabilidade: resultados das pesquisas podem ser reproduzidos	Pesquisadores reconhecem e endereçam implicações de sua subjetividade

### 3.1. A PERSPECTIVA POSITIVISTA NA ÁREA DE SI

Filosoficamente falando, o positivismo foca os fatos, troca o imaginário, o metafísico pelo real e observável, em busca de uma explicação do fenômeno em si, descartando suas origens e seus fins. O método positivista tem características como: imaginação subordinada à observação; descrição do fato real; quantificação e análise estatística; foco na precisão, rigor e clareza; privilégio do objeto; posição de neutralidade.

Segundo essa abordagem também conhecida como funcionalista, os analistas de sistemas devem descobrir os verdadeiros requisitos do sistema e modelá-los objetivamente. Eles adquirem o conhecimento através da busca de relações causa-efeito mensuráveis, objetivas e tratam de uma realidade organizacional que é independente da percepção ou crença do observador.

O funcionalismo é o paradigma dominante em praticamente todas as escolas de análise de sistemas. Nele há uma preocupação com o levantamento preciso de requisitos que buscam a identificação de “requisitos verdadeiros”. Para Pozzebon e Freitas (1997):

A postura mais freqüente encontrada entre os analistas de sistemas é a de especialistas aptos a modelar as necessidades dos usuários de sistemas. O resultado do seu trabalho é inquestionável: corresponde ao atendimento das necessidades em termos de informação dos usuários. Falhas no levantamento dos requisitos dos sistemas fatalmente são atribuídas às dificuldades dos usuários em “saber o que necessitam” ou em “transmitir o que desejam”.

De acordo com Ferreira e Lima (2005), as metodologias tradicionais, de cunho positivista, dividem o desenvolvimento em duas fases: planejamento e desenvolvimento propriamente dito. O planejamento deve ser o mais completo e detalhado possível; os desenvolvedores devem fazer um exaustivo levantamento de requisitos para depois se envolverem na criação da arquitetura do sistema (funcionalidades, características, soluções). O objetivo desta fase é prever o maior número possível de funcionalidades que o sistema deverá possuir para assim ser construído o alicerce do sistema.

A fase de levantamento de requisitos consiste em entrevistas com os clientes para a obtenção de informações sobre as necessidades do sistema e o funcionamento do processo. Após sua conclusão, é definido o escopo do projeto para posterior aprovação pelo cliente. Depois de acordado com cliente, os desenvolvedores partem para o desenvolvimento propriamente dito, ou seja, irão desenvolver as funcionalidades do sistema.

Para Ferreira e Lima (2005), as experiências práticas e as pesquisas têm demonstrado deficiências nesta metodologia de desenvolvimento de software. Quando ocorrem modificações ao longo do projeto, aparece uma grande resistência dos programadores na sua incorporação, pois, como o sistema já foi todo idealizado, uma modificação pode acarretar mudanças estruturais, impactando no prazo e no custo do projeto. Além disso, a interação com o cliente ou usuário fica restrita a esta fase inicial do projeto, ou ao momento de teste do protótipo, limitando consideravelmente as retro-alimentações e os *feedbacks* dos usuários no decorrer do desenvolvimento.

Por isso, o sistema produzido pouco atende as necessidades dos usuários, ou seja, é um sistema mais voltado para a representação do analista sobre a atividade do usuário do que realmente voltado para as reais necessidades destes. A origem destes problemas está relacionada ao método de levantamento de requisitos desta metodologia. O problema principal é a crença na capacidade dos seres humanos de prever todas as funcionalidades do sistema logo de início; é por isso que se limita a definição dos requisitos a esta fase, tentando explicitar tudo o que o cliente deseja no sistema. Esta crença não condiz com o funcionamento cognitivo dos seres humanos, que é situado e dinâmico, isto é, muda ao longo das interações com o sistema. Assim, não é possível aos futuros usuários ou clientes saberem a priori tudo o que o sistema precisa contemplar, tendo em vista que também desconhecem o funcionamento do software, suas possibilidades e limitações. Não raras vezes, os clientes demandam funcionalidades impossíveis de serem desenvolvidas devido à complexidade, e, noutras vezes, aquém das possibilidades do software.

Outra característica deste método de levantamento de requisitos é a intensificação do contato com o cliente na fase inicial e o distanciamento nas fases de desenvolvimento. Há uma separação temporal entre concepção e execução. Por este motivo, quando termina a fase de levantamento de requisitos, não tem mais sentido o contato com o usuário, configurando-se, desse modo, um longo distanciamento entre analistas e clientes durante o desenvolvimento do sistema propriamente dito, sendo o contato retomado apenas nos momentos de teste e na implementação dos sistemas. Esta distância contribui para a criação de um software cujas funcionalidades não atendem bem as reais necessidades dos usuários.

A construção de sistemas seguindo esta metodologia de desenvolvimento foi amplamente usada pela comunidade até a década de 80. E ainda o é em alguns casos. Esses processos rígidos, sustentados em muita documentação e burocracia, são considerados muito preventivos, tentando evitar situações dispendiosas ao invés de buscar a otimização. Procuram analisar exaustivamente os requisitos e eliminar erros antes do seu aparecimento na etapa de implementação.

O modelo em cascata, apresentado em Sommerville (2003), conhecido também como clássico ou linear, tem como premissa a metodologia apresentada anteriormente, pois se caracteriza por seguir um modo quase estritamente seqüencial das etapas de desenvolvimento. A fase de análise só tem início quando a fase de levantamento de requisitos é considerada concluída. Semelhantemente, a fase de projeto ou *design* só começa após a finalização da fase de análise e assim por diante. Contudo, Bezerra (2007) explica que os seres humanos têm sérias dificuldades em lidar, de uma só vez, com sistemas complexos. É difícil construí-los de uma forma puramente seqüencial. Ao invés disso, adota-se uma estratégia incremental e iterativa, tentando dividi-lo em grupos de pequenos problemas. Além do mais, atualmente, a volatilidade dos requisitos é um fato com o qual a equipe de desenvolvimento tem de conviver. A menos que o sistema a ser desenvolvido seja muito simples e estático, que são características raras nos dias de hoje, é praticamente impossível pensar em todos os detalhes a princípio. A existência de requisitos voláteis é mais uma regra do que uma exceção. Consequentemente o congelamento de requisitos é impraticável para organizações que precisam se adaptar, cada vez mais rapidamente, a mudanças.

### **3.2. A PERSPECTIVA INTERPRETATIVISTA NA ÁREA DE SI**

Segundo Rodrigues Filho et al (1999) algumas anomalias podem ser identificadas no paradigma funcionalista/positivista, o que nos remete a uma possibilidade de crise paradigmática como estabelece Kuhn (2001) em seu livro, “A estrutura das revoluções científicas”. Algumas anomalias do paradigma positivista são apresentadas por Klein (1985) apud Rodrigues Filho (1999):

- A construção de sistemas de informação como artefato técnico, ignorando a dimensão social;
- A definição da informação derivada de um dado objetivo, através de procedimentos formais e impessoais;
- Aderência ao modelo burocrático de organização como máquina, cujo racionalismo causa alienação e falhas na implementação dos sistemas de informação;
- Interpretação do desenvolvimento de sistemas de informação como um processo de engenharia, evitando a participação do usuário.

A impessoalidade presente em tais anomalias evidencia a necessidade de uma maior interação social, em que o usuário não seja evitado, mas seja um ator ativo no processo de desenvolvimento e implementação de SI's. Nesse sentido o paradigma interpretativo mostra-se como uma alternativa para resolução das anomalias acima citadas. Nessa nova perspectiva, a realidade a ser analisada para o desenvolvimento de um sistema de informações passa a ser vista como um texto, seguindo a lógica hermenêutica de interpretação da realidade. Assim, a realidade é socialmente construída e não objetivada e reducionista. As interações sociais passam a ser consideradas e os usuários passam a ter papel essencial na interação com os analistas para um desenvolvimento em conformidade com a realidade, considerando as suas peculiaridades e complexidade.

Para Pozzebon e Freitas (1997), no paradigma interpretativista, chamado por eles de relativismo social, o analista é tido como um facilitador. Dessa forma, os SI's emergem como uma construção organizacional da realidade, num processo de criação de sentido, o que nos remete a uma reflexão fenomenológica de análise organizacional. Assim, nas palavras do autor:

A função do desenvolvedor é interagir com a gerência para achar o tipo de sistema que faz mais sentido, uma vez que não existem critérios objetivos para distinguir bons e maus sistemas. Ele deve atuar como um facilitador, estimulando a reflexão, cooperação e aprendizado experimental. Os próprios negócios não induzem a uma objetiva realidade econômica, mas envolvem mudanças em leis sociais, convenções, cultura e atitudes. Todo sistema com aprovação das partes é legítimo.

Este paradigma é, do ponto de vista epistemológico, anti-positivista, pois, para ele a busca de uma explicação causal e empírica dos fenômenos sociais é ilegítima, podendo assim ser sobreposta pelo julgamento pessoal. Desse modo a complexidade, na sua reflexão de base ontológica, não pode ser considerada imutável ou simplificada em modelos, ao invés disto, deve ser socialmente construída. Nesse sentido as mudanças devem se guiar pelo conhecimento aceito, em consonância com a opinião geral, gerando mudanças que não afetem a harmonia social (HIRCHHHEIM & KLEIN, 1989 apud POZZEBON E FREITAS, 1997).

Ultimamente, devido à complexidade cada vez maior dos sistemas, os modelos mais utilizados para o desenvolvimento são os que usam a abordagem incremental e iterativa, mesmo que, apesar de todos os seus problemas, o modelo em cascata ainda venha sendo utilizado durante anos. O modelo incremental e iterativo foi proposto como uma resposta aos problemas encontrados no modelo em cascata. Um processo de desenvolvimento segundo esse modelo divide o desenvolvimento de software em iterações. Em cada iteração, são realizadas as atividades de análise, projeto, implementação e testes para uma parte do sistema.

Uma vez alocados os requisitos a uma iteração de desenvolvimento, estes requisitos são analisados, projetados, implementados, testados e implantados. Na próxima iteração, um outro subconjunto dos requisitos é considerado para ser desenvolvido, o que produz uma nova versão (ou incremento) do sistema que contém extensões e refinamentos sobre a versão anterior. Desta forma, o desenvolvimento evolui em versões, através da construção incremental e iterativa de novas funcionalidades até que o sistema completo esteja construído [Bezerra, 2007].

Segundo Bezerra (2007), o modelo incremental e iterativo incentiva a participação do usuário nas atividades de desenvolvimento. Isso porque os usuários desde o início do projeto já possuem uma versão do sistema para utilizar. De fato, as versões iniciais do sistema não contêm a implementação de todas as funcionalidades necessárias. Porém, essa situação é muito melhor do que aquela em que o usuário recebe o sistema todo de uma vez, somente no final do projeto, conforme ocorre no modelo em cascata.

Desenvolvida no início da década de 90, *Extreme Programming* (XP) é uma das mais modernas das metodologias de desenvolvimento disponíveis e amplamente utilizadas pelas empresas. XP faz parte das metodologias leves, ou ágeis, de desenvolvimento de software. É baseada em um processo simples, flexível, disciplinado e marcado pela conscientização e envolvimento de todos para o desenvolvimento, pois se sustenta no conhecimento tácito do time do projeto no intuito de lidar mais facilmente com as mudanças constantes de requisitos e reduzindo o custo global de desenvolvimento (BECK, 1999).

O processo de desenvolvimento com XP é essencialmente interativo. As correções vão sendo feitas à medida que o projeto progride, com constantes conversas entre os clientes e os desenvolvedores. Desta maneira, pequenas alterações na rota põem o projeto na direção certa. É importante que haja um cliente no ambiente de desenvolvimento, isto é, um usuário do sistema que participe de maneira mais ativa no projeto. Esta pessoa se encarrega de tirar dúvidas sobre o funcionamento da empresa e participa das dificuldades do dia-a-dia do desenvolvimento. Assim, o diálogo entre os usuários e os desenvolvedores torna-se mais uniforme, os desenvolvedores passam a ser encarados como parte da equipe e o usuário não é mais considerado como “aquele chato que não sabe o que quer”. Um sistema que tem a participação do usuário tem mais chance de dar certo, por representar melhor as necessidades do cliente e também por ter um defensor de sua implantação, parte ativa do processo de criação do sistema. Por esta razão que um dos valores básico preconizado por XP é a comunicação. É fundamental haver comunicação entre o desenvolvedor e o cliente. O desenvolvedor não conhece o negócio do cliente e o cliente não conhece as dificuldades inerentes à tarefa de desenvolvimento. A comunicação deve ser clara e aberta para que os desenvolvedores tenham a mesma visão dos usuários do sistema.

#### **4. ÊNFASE NA PARTICIPAÇÃO DO USUÁRIO NO DESENVOLVIMENTO DE SISTEMAS MAIS EFICIENTES E MAIS ADAPTADOS À REALIDADE**

Nas subseções seguintes serão abordados temas, problemas e características do desenvolvimento de sistemas que enfatizam a importância da participação do usuário que detém o conhecimento do negócio, tendo em vista a valorização de suas idéias como fator preponderante no envolvimento e na responsabilidade com o produto final. Com isso, pressupõe-se uma menor resistência a mudanças, uma maior aceitação da tecnologia, facilidade de uso e grande eficiência de operação.

##### **4.1. PARTICIPAÇÃO NA MUDANÇA DE SISTEMA**

Segundo a perspectiva interpretativista, a participação dos usuários é tida como elemento essencial para o sucesso do sistema implementado, considerando assim as peculiaridades do trabalho e as necessidades das pessoas que irão utilizar os sistemas, novos ou modificados. Já a perspectiva *hard*, trabalha numa lógica mais técnica, associada à área de Engenharia de Software, na qual a análise é essencialmente objetiva e a prioridade são os processos em detrimento das necessidades individuais dos funcionários da empresa.

Para Furnival (1995), as limitações das metodologias de design tradicionais têm sido solucionadas pela adoção de metodologias *soft*, “que se baseiam na idéia de que somente a participação contínua dos usuários no desenvolvimento do novo sistema pode evitar problemas futuros”. Nesse sentido é reconhecido o conhecimento especializado dos usuários sobre seu trabalho. Assim, o usuário é percebido como ator fundamental para a construção de um sistema de alta qualidade e produtivo.

Percebe-se no argumento acima que teoricamente a participação tem sido valorizada em estudos sobre o desenvolvimento e implementação de SI. Porém, diversos sistemas implantados não têm o devido sucesso, como ratificam Furnival (1995) e Rodrigues Filho *et al* (1999). Um dos problemas mais comuns é a resistência de funcionários a novos sistemas implantados devido a uma falta de identificação com os propósitos dessas novas aplicações. Nesse sentido, a participação dos usuários busca, através de uma interação efetiva, uma melhor compreensão por parte dos funcionários das potencialidades e possíveis benefícios dos sistemas de informação que são implantados.

Segundo O'Brien (2006) a implantação de um novo sistema, geralmente dá origem a um processo conhecido como curva de aprendizagem. Nesse sentido, o pessoal que opera e utiliza o sistema cometerá erros simplesmente por ainda não estarem familiarizados com ele. Embora tais erros diminuam com o passar do tempo, na medida em que os usuários adquirem certa experiência, para o autor, tais erros indicam, na verdade, áreas em que o sistema precisa melhorar.

Nesse sentido, a mudança de sistema pode ser vista como fator relevante na aceitação de uma nova tecnologia por parte dos usuários, bem como na sua utilização de forma produtiva. Tal processo de mudança deve, segundo O'Brien (2006) ser acompanhado de uma revisão pós-implantação para garantir que os sistemas recém-implementados alcancem os objetivos. Assim, o sistema deve ser acompanhado incluindo-se atividades de revisão e auditoria periódica do sistema para garantir o alcance de objetivos, além disso, deve-se monitorar de forma continuada o novo sistema para a detecção de potenciais problemas ou mudanças necessárias.

#### **4.2. USUÁRIO FINAL, MUDANÇA, PARTICIPAÇÃO E ENVOLVIMENTO**

Os processos de mudança são geralmente dificultados pela resistência das pessoas em aceitar novas experiências e mudanças nas suas rotinas, seja de trabalho ou de sua vida pessoal. Na implementação de sistemas de informação não é diferente, como ratifica O'Brien (2006), as mudanças nas tecnologias podem gerar receio e resistência por parte do usuário final dos sistemas. Para esse autor a chave para solucionar tais resistências consiste em utilizar-se de educação e treinamento adequados.

Para O'Brien (2006), ainda mais importante é o envolvimento do usuário-final nas mudanças organizacionais e no desenvolvimento de novos sistemas de informação. Nesse sentido a participação do usuário no desenvolvimento de projetos de sistemas de informação, antes da sua implementação é particularmente importante para a redução do potencial de resistência do usuário-final. Essa concepção de envolvimento ajuda a garantir que os usuários assumam a autoria e que seu produto final atenda a todas as suas necessidades. Assim, seja qual for o grau de elegância técnica e de eficácia no processamento de dados de um determinado sistema, ele não será efetivo se frustrar ou incomodar os seus usuários.

O envolvimento dos usuários é definido por Barki (1989) como sendo "um estado psicológico subjetivo relativo à importância e relevância pessoal de um sistema para o usuário". Esse autor distingue esse conceito do de participação que é para ele: "o conjunto de comportamentos ou atividades desempenhadas por um usuário durante o desenvolvimento de um sistema de informação". Nesse trabalho o autor argumenta que até então as pesquisas têm abordado com maior frequência a participação dos usuários e em menor quantidade o seu envolvimento no desenvolvimento de sistemas de informação.

Para esse autor o envolvimento dos usuários é diretamente relacionado ao resultado dos sistemas de informação produzidos e tem sido observado como determinante para: atitudes dos usuários, qualidade dos sistemas e sucesso dos sistemas.

#### **4.3. MODELO DE ACEITAÇÃO A TECNOLOGIA (TAM)**

A utilização e o convívio de usuários com produtos e serviços de base tecnológica é o ponto de partida para as investigações a cerca da aceitação a tecnologia. Um modelo bastante utilizado em estudos dessa natureza é o *Technology Acceptance Model* (TAM), desenvolvido por Davis (1986), como uma adaptação do modelo Theory of Reasoned Action (TRA) ou teoria da ação racionalizada. Tem por objetivo fornecer explicações sobre as causas determinantes da aceitação dos computadores de forma geral, bem como explicar o

comportamento dos usuários frente às diversas tecnologias ligadas a informática, prevendo e explicando a pesquisadores e interessados porque um sistema pode ser aceito ou invalidado, assim como pode dar orientações para devidas correções (COSTA FILHO e PIRES, 2005).

Ainda segundo Costa Filho e Pires (2005), o estudo parte da aceitação baseada na esfera cognitiva e afetiva, desse modo o modelo TAM se sustenta em dois construtos importantes para a aceitação a tecnologia, que são a utilidade percebida (*percieved utility*) e a facilidade de uso percebida (*perceived ease-of-use*)

#### 4.4. DESENHO PARTICIPATIVO

Segundo Rodrigues Filho e Ludmer (2005), “os modelos racionais tradicionais ou modelos “hard” têm sido incapazes de lidar com o mundo organizacional cada vez mais complexo e turbulento, com perspectivas humanas conflitantes”. Nesse contexto os autores argumentam que outras opções (paradigmas) devem ser buscados para que o contexto tecnológico não seja enfatizado em detrimento do humano na área de Sistemas de Informação.

Ainda segundo esses autores, a metodologia de Desenho Participativo (DP) foi evidenciada inicialmente nos países escandinavos e tem ganhado popularidade na América do Norte e em outros países. Esta metodologia de desenho participativo enfatiza muito mais a participação das pessoas no desenvolvimento de sistemas de informação em detrimento de questões técnicas. Assim, o DP “reconhece o papel central do usuário no processo de desenho de sistemas e enfatiza as oportunidades para que o usuário possa influenciar o desenvolvimento”.

A caracterização do desenho participativo é também abordada por Furnival (1995) quando afirma que:

Tais abordagens, as quais surgiram na década de 80, focalizam a participação dos usuários no processo de design de um sistema. 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.

Para contextualizar esta metodologia de desenvolvimento faz-se necessário relacioná-la a uma análise da engenharia de requisitos, a exemplo de Rocha (2000) que identificou três tipos de abordagens relacionadas às perspectivas “hard” (positivistas) e “soft” (interpretativistas) de desenvolvimento, descritas abaixo:

- **Abordagens Tecnológicas:** aquelas que realçam a visão objetiva dos requisitos e os aspectos técnico/tecnológicos dos SI. Englobam os conhecidos métodos "hard", tradicionais ou estruturados.
- **Abordagens Sócio-organizacionais:** aquelas que enfatizam a visão intersubjetiva dos requisitos e os aspectos sociais e organizacionais dos SI. Englobam os métodos que vulgarmente são conhecidos por métodos "soft", interpretativistas ou construtivistas.
- **Abordagens Sócio-tecnológicas:** aquelas em que há uma situação intermédia, em que existe uma combinação de algo de cada uma das duas abordagens anteriores.

Geralmente focam-se primeiro nas questões sócio-organizacionais, deixando as questões técnico/tecnológicas para uma segunda fase.

Para esse autor a última abordagem é considerada, a partir de uma análise dos principais modelos de maturidade da área de SI, a mais adequada e relacionada com atividades de Engenharia de Requisitos (ER) de maturidade superior. Assim, o uso de uma abordagem sócio-técnica, que prioriza questões sócio-organizacionais, dentre elas a participação dos usuários no desenvolvimento e implementação de SI's é um caminho em direção a uma maturidade dos sistemas de informação desenvolvidos/implementados (ROCHA, 2000).

## 5. CONCLUSÃO

Diante do exposto, é possível perceber que se tenta rotular a metodologia tradicional como uma visão puramente objetivista e orientada ao produto, sem relevante preocupação com os aspectos subjetivos dos requisitos. No entanto, as novas abordagens da engenharia de software, que seguem um desenvolvimento iterativo durante todo o ciclo de vida do desenvolvimento do sistema, apontam para uma participação do usuário em maior escala que tempos atrás. Uma vez que os processos ágeis pressupõem a construção de forma iterativa e incremental em todas as etapas do processo. Dessa forma, a abordagem rotulada como *hard* não pode mais ser considerada essencialmente positivista.

As abordagens tradicionais de desenvolvimento de software fundamentam-se em certos conceitos técnicos que, geralmente, não englobam fatores sociais e psicológicos da organização na qual o novo sistema será implantado. Os sistemas são situados em ambientes organizacionais, isso inclui políticas e procedimentos que são, por sua vez, regidos por questões políticas, econômicas, sociais e ambientais mais amplas. Se o ambiente organizacional não for adequadamente compreendido, o sistema poderá não atender as necessidades da empresa e ser rejeitado pelos usuários e pelos gerentes. Dentre os fatores humanos e organizacionais que afetam o sistema destacam-se as mudanças nos processos de trabalhos, modificação na estrutura do poder político da organização e diminuição da habilidade dos usuários ou modificação do modo de trabalho. O custo de ignorar tais fatores é, às vezes, alto, pois usuários insatisfeitos podem resistir ao uso do sistema ou usá-lo de uma maneira diferente da qual foi originalmente projetado levando o sistema a não atingir seus objetivos com sucesso.

As limitações das metodologias tradicionais têm sido suavizadas pela adoção de metodologias ágeis de software, que se baseiam na idéia da participação contínua dos usuários no desenvolvimento do novo sistema. Nestas metodologias o usuário é parte integrante da equipe de desenvolvimento, desta forma, estabelece-se uma relação de transparência e confiança desde o princípio e mantém-se até o final. Por outro lado, o envolvimento mais estreito do usuário no processo de desenvolvimento e o seu constante *feedback* em relação às versões que são liberadas resultam em uma quantidade de trabalho significativa para fazer as alterações e correções necessárias. Assim, se o re-trabalho resultante dos requisitos em evolução não for bem gerenciado, ele pode ser um fator de desaceleração do ritmo do projeto.

No entanto, o que de fato acontece na prática é uma combinação de influências de vários paradigmas em cada situação, sendo normalmente um deles predominante. Ou seja, o tipo de abordagem vai estar de acordo com as características do sistema, da equipe e/ou empresa contratada para implantá-lo, bem como da experiência e disponibilidade espacial e temporal dos usuários para participar ativamente do processo. É importante lembrar que, visando alcançar eficiência, apenas usuários finais que possam colaborar ricamente com o sistema devem ser designados para fazer parte da equipe. Eles devem se sentir responsáveis

pelo resultado final do projeto. Usuários indecisos e com dificuldades de expressar ou retratar os processos práticos podem confundir a equipe e atrasar a elaboração do produto final.

Apesar das abordagens rotuladas de interacionistas promoverem uma aproximação entre a equipe de desenvolvimento e os usuários - muito construtiva na medida em que o grupo de desenvolvimento toma um maior contato com a realidade do usuário, e por outro lado, o usuário tem conhecimento do ambiente de desenvolvimento e de eventuais restrições tecnológicas -, existem também pontos fracos. Como mencionado acima, se grupo participante da dinâmica não for homogêneo, haverá discussões paralelas, que não agregam nenhum valor à especificação do sistema, e pode ser difícil a obtenção da convergência do grupo acerca de determinados assuntos.

## 6. REFERÊNCIAS

BARKI, Henri; HARTWICK, Jon. *Rethinking the user Involvement*. In: MIS Quartely, March, 1989.

BECK, Kent. *Extreme Programming Explained*. Addison-Wesley Pub Co; 1st edition, October 1999.

BEZERRA, Eduardo. *Desenvolvimento incremental e iterativo*. MundoOO. Disponível em <http://www.mundooo.com.br/php/modules.php?name=MOOArtigos&pa=showpage&pid=20>. Acessado em agosto de 2007.

COSTA FILHO, Bento A.; PIRES, Péricles J. *Avaliação dos Fatores Relacionados na Formação do Índice de Prontidão À Tecnologia - TRI (Technology Readiness Index) como Antecedentes do Modelo TAM (Technology Acceptance Model)*. In Anais: XXIX Encontro da Associação Nacional de Pós-Graduação e Pesquisa em Administração, Brasília, 2005.

ESPINDOLA, Rodrigo S.; MAJDENBAUM, Azriel; AUDY, Jorge L. N. *Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software*. Anais do WER04 - Workshop em Engenharia de Requisitos, Tandil, Argentina, Dezembro 9-10, 2004, pp 226-238.

FERREIRA, Renata B.; LIMA, Francisco P. *Definição de Requisitos na Concepção de Sistemas Informatizados: da elicitação à cooperação*. Anais do 1º Workshop um Olhar Sociotécnico sobre a Engenharia de Software – WOSSES. Rio de Janeiro, 2005.

FURNIVAL, Ariadne C. *A participação dos usuários no desenvolvimento de sistemas de informação*. In: Ciência da Informação - Vol 25, Número 2, 1995.

HOPPEN, Norberto. *Sistemas de Informação no Brasil: uma Análise dos Artigos Científicos dos Anos 90*. RAC, v.2, n.3, Set./Dez. 1998: 151-177.

KUHN, Thomas S. *A estrutura das revoluções científicas*. 6 ed. São Paulo. Editora Perspectiva, 2001.

O'BRIEN, James A. *Sistemas de Informação e as decisões gerenciais na era da internet*. São Paulo: Saraiva, 2 ed., 2006.

POZZEBON M.; FREITAS H.. *Por um conjunto de princípios que possibilitem a construção de novos modelos de sistemas de informação*. São Paulo: RAP, v. 31, nº 5, Set/Out. de 1997, p. 87-104.

RAMOS, Isabel M. P. *Aplicações das Tecnologias de Informação que suportam as Dimensões Estrutural, Social, Política, Simbólica do Trabalho*. Tese de Doutorado em Tecnologias e Sistemas de Informação - Universidade do Minho, Departamento de Sistemas de Informação. Guimarães - Portugal, Dezembro de 2000.

REINHARD, Nicolau. *Implementação de Sistemas de Informação*. In: Tecnologia da Informação. Organizado por Albertin, Alberto L. e Albertin, Rosa M. M. São Paulo: Atlas, 2005.

REZENDE, Denis A. *Sistemas de Informações Organizacionais: guia prático para projetos em cursos de administração, contabilidade e informática*. São Paulo: Atlas, 2005.

ROCHA, Álvaro; VASCONCELOS, José. *PERSI – Modelo para Avaliação do Pendor da Engenharia de Requisitos de Sistemas de Informação*. Disponível em: [http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER03/alvaro\\_rocha.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER03/alvaro_rocha.pdf). Acesso em 11 de julho de 2007.

RODRIGUES FILHO, J.; Borges, C. F.; Ferreira, R. C. F.. *O paradigma interpretativo na pesquisa e desenvolvimento de sistemas de informação*. Conferência da Business Association of Latin American Studies, New Orleans, 1999.

SOMMERVILLE, Ian. *Engenharia de Software*. São Paulo: Addison Wesley, 2003

VASCONCELOS, Isabella F. F. G.; MASCARENHAS, André. *Paradoxos Organizacionais e Tecnologia da Informação: Uma Análise Crítica da Implantação de Sistemas de Auto-Atendimento na Área de Gestão de Pessoas da Souza Cruz*. In Anais: XXVII Encontro da Associação Nacional de Pós-Graduação e Pesquisa em Administração, Atibaia, Setembro de 2003.

VASCONCELOS, Isabella F. F. G.; MOTTA, Fernando C. P.; PINOCHET, Luis H. C.; SEGALLA, Denise R. *O Lado Humano da Tecnologia: um Estudo Exploratório sobre os Paradoxos Organizacionais dos Sistemas de Informação*. In Anais: XXV Encontro da Associação Nacional de Pós-Graduação e Pesquisa em Administração, Campinas 2001.

VIANNA, Elaine Cristina C. M. *Estudo e Proposta de Práticas Participativas na Gestão de Requisitos*. Trabalho Final de Mestrado Profissional - Universidade Estadual de Campinas, Instituto de Computação. Campinas, 2004.

WEBER, Ron. *The Rhetoric of Positivism Versus Interpretivism: A Personal View*. In: MIS Quarterly, Vol. 28, No. 1, March 2004.