

Uma Proposta de Avaliação do Usuário para a Elicitação de Requisitos de Software

Júlio César Godoy Bertolin (julio@upf.tche.br)

Maristela Piva Guazelli (maris@annex.com.br)

Willingthon Pavan (pavan@upf.tche.br)

Universidade de Passo Fundo - UPF

Campus I - Bairro São José - Cx Postal 611/631 - CEP 99001-970 - Passo Fundo - RS - Brasil

Palavras-chave: requisito, resistência, elicitação, fontes de informação.

Resumo: O presente trabalho visa incrementar a definição de requisitos de software critérios para avaliação e tratamento da resistência do usuário ao processo de criação do software. Um usuário pode ser a principal fonte de informação e a pessoa menos interessada na implantação do novo sistema. A resistência ao processo de criação do software tem origem principalmente em aspectos relacionados à motivação. O trabalho propõe, para a identificação da resistência uma análise da motivação do usuário em relação ao processo de criação do software e através do uso de uma taxionomia para as causas de resistência e de heurísticas propostas, a classificação do usuário quanto à origem e ao potencial de resistência. Nele é trabalhado o tratamento da resistência do usuário. Estratégias e abordagens para superação da resistência ao processo de criação do software são apresentadas. Como resultado, obtém-se maior eficiência na etapa de identificação de fontes de informação.

1. Introdução

Dentre as atividades realizadas durante a definição de requisitos, foi, certamente, a modelagem que mais evoluiu nos últimos anos; a elicitação, ao contrário, pouco se desenvolveu. Considerações sobre as razões desse fato conduzem à natureza essencialmente humana da elicitação. Essa característica, centrada na comunicação interpessoal, fruto da interação entre usuário e engenheiro de software, pode ter afastado pesquisadores, mais afeitos às atividades estruturadas das demais fases do ciclo de vida de sistemas (Zirbes, 1993).

Um problema, ao ser tratado por pessoas, agrega sempre características inerentes ao fator humano do querer, do saber, do poder e, principalmente, da comunicação e do entendimento do requisito (Zanlorenzi, 1998). Como a fonte de problemas vem das pessoas, é importante identificar qual a necessidade de ter resolvido o problema e se existe, realmente, o desejo de solução. Assim o processo de criação do software tem tanto em comum as questões de técnicas e métodos como com o estudo das pessoas. Fatores humanos, tais como, motivação, linguagem, aspecto cognitivo e comportamento são elementos importantes que devem ser considerados no contexto da elicitação e na identificação de usuários como fonte de informação. Como um usuário pode, ao mesmo tempo, ser a principal fonte de informação e a pessoa menos interessada na implantação do novo sistema a identificação da resistência é fundamental para o processo de criação do software.

A parte 2 deste artigo destaca o aspecto humano da definição de requisitos. A parte 3 apresenta uma proposta de taxionomia para as causas de resistência ao processo de criação do software. A parte 4 apresenta subsídios para avaliação da resistência do usuário. A parte 5 apresenta propostas para tratamento da resistência no contexto da engenharia de software. A parte 6 finaliza com um resumo de considerações sobre o processo de avaliação do usuário e os resultados esperados com a aplicação das propostas apresentadas no trabalho.

2. A motivação do usuário: um problema de comunicação

Para que a elicitação tenha sucesso, é fundamental que o engenheiro de software possa se comunicar de forma eficaz com o usuário. Como principal fonte de informação, é com o usuário que serão coletados, através de entrevistas, questionários, reuniões, observações, etc., os fatos do UdI. No entanto, é notório que existem incríveis barreiras na comunicação entre engenheiros de software e usuários (Leite, 1994).

As diferenças de conhecimentos entre usuários e engenheiros de software são reflexos de culturas diferentes e não são fáceis de se resolver. Nesse mesmo campo de dificuldades de comunicação, encontra-se o que é conhecido como "conhecimento tácito", isto é, o conhecimento que é trivial para um, que dificilmente explicitado, mas não o é para o outro.

Porém, quando a comunicação é analisada pela perspectiva da pragmática, isto é, dos efeitos comportamentais da comunicação, temos que todo comportamento, não só a fala, é comunicação; e toda comunicação afeta o comportamento (Watzlawick, 1993). Então, no que poderíamos chamar de outra classe de dificuldades de comunicação na elicitação, estão as causadas pela resistência por parte do usuário em contribuir com dados sobre o UdI (universo de informação) para definição de requisitos. Na maioria dos casos, tal resistência está relacionada a fatores humanos (Hampton, 1990). É durante o desenvolvimento da tarefa de coleta de fatos que esses fatores podem dificultar a comunicação entre o engenheiro de software e o usuário. Dessa forma, durante a identificação de fontes de informação, esse aspecto deve ser considerado no intuito de se evitar que usuários dificultem a comunicação e, por conseguinte, a própria definição de requisitos. Essa possibilidade reforça a idéia de que uma exploração eficiente das fontes de informação antes de se iniciar a coleta de fatos pode ser de grande validade.

No que diz respeito ao grau de motivação que um usuário pode possuir em relação à sua contribuição no processo de criação do software, apontam-se três possibilidades (Alencar, 1976). Ele tanto pode apresentar contrariedade, como também indiferença; em ambas as situações, estará dificultando o trabalho do engenheiro de software na tarefa de elicitação de requisitos, e claro que com maior ênfase na primeira alternativa. Pode, também, apresentar simpatia em relação ao processo de criação do software com o que, evidentemente, estará facilitando o desenvolvimento da definição de requisitos.

A indiferença pode ser superada pela conscientização do usuário. A contrariedade, no entanto, é uma questão mais complexa. A resistência de um usuário ao desenvolvimento do software tem origem principalmente em aspectos relacionados à motivação, algo que é inerente a todos os seres humanos.

Sendo o usuário a principal fonte de informação fica claro que a coleta de fatos com indivíduos contrariados com o processo de desenvolvimento do software dificultará ou, mesmo, comprometerá de forma significativa o resultado da elicitação e, por conseguinte, a qualidade do software a ser produzido.

3. Taxionomia para as causas de resistência

Motivar implica levar a agir ou a produzir uma resposta (Hampton, 1990). Talvez a característica fundamental de todas as formas de motivação seja uma mobilização de energia por parte do organismo, a qual pode ocorrer em uma variedade de condições e circunstâncias. Entre outras, pode estar ligada a condições que resultam em emoções, como medo ou raiva, que levam o organismo a mobilizar energias em direção ou em oposição ao objeto aliciador da emoção. Os motivos emocionais são, em larga escala, adquiridos no processo de interação do organismo com o ambiente (Parry, 1972). O medo, que é talvez o motivo emocional mais estudado, além de ter alta resistência à extinção, apresenta alto grau de generalização. Segundo Izard e Tomkins "o medo é um afeto de grande potência

na determinação do que o indivíduo perceberá, pensará e fará. E o mais constringente de todos os afetos. O medo reduz as alternativas do comportamento”.

Pode ainda, a motivação, estar ligada a condições externas sociais, como o desejo de promoção e *status*, que catalisam e dirigem as energias do indivíduo para a obtenção de certas metas, da mesma forma que os impulsos de fome, sede ou sexo. Esses motivos sociais foram classificados por Murray, que distinguiu diferentes grupos de necessidades, a que chamou de *psicogênicas*: algumas ligadas ao desejo de realização e prestígio; outras, a desejos e ações que envolvem a defesa do *status* e a esquiva de humilhação; outras, mais relacionadas ao desejo de influir, controlar e emitir ou resistir a influências de outros sujeitos, e, ainda, outras ligadas à manifestação de afeição entre pessoas (Hampton, 1990). Dessa forma, o comportamento de resistência de uma pessoa a determinada modificação em seu ambiente pode estar vinculado tanto a motivos emocionais, como, por exemplo, o medo, como a motivos sociais, como o desejo de promoção.

Com base nas análises da literatura e na análise lógica de ambientes de desenvolvimento de sistemas, na qual o cliente tem, enquanto usuário, papel fundamental no processo de criação do software, constata-se uma série de causas para possíveis comportamentos de resistência do usuário em contribuir para a construção do novo sistema e, conseqüentemente, em participar como fonte de informação na coleta de fatos. Dessa forma, os usuários podem ser classificados em razão das suas causas para apresentar resistência ao desenvolvimento do novo software. Entre outras, podem-se identificar as seguintes causas:

- Resistência natural do ser humano a mudanças (Hampton, 1990): as pessoas, muitas vezes, resistem à mudança porque ela ameaça os seus hábitos e relações estabelecidas. A perspectiva de mudança pode despertar temores sobre a falta de capacidade para satisfazer os novos requisitos e a perda de padrões estabilizados de relações pessoais. É natural, então, que as pessoas tenham medo de modificações futuras ou de coisas estranhas (Pressman, 1995). Nesse contexto, muitas pessoas, comprovadamente, apresentam aversão a aproximar-se de computadores. Além do mais, o que já é conhecido, no caso o sistema manual ou antigo, dificilmente apresentará surpresas desagradáveis; em oposição, um novo programa de computador cria muitas expectativas para o usuário.
- Fatores políticos dentro da organização (Zanlorenzi, 1998): nas organizações, as disputas pelo poder e relações de influências entre diferentes pessoas podem causar contrariedade à implantação de um novo sistema. Um usuário envolvido num complexo jogo de disputas dentro da organização pode vislumbrar no novo software uma ameaça às suas pretensões. O próprio descontentamento com o salário, cargo ou situação pessoal dentro da organização deve resultar numa certa resistência do usuário em colaborar, de forma plena, com informações sobre o UDI.
- A perda de importância no contexto da organização (Hampton, 1990): em muitas organizações, os usuários ainda possuem o monopólio sobre o conhecimento de determinadas tarefas ou processos executados rotineiramente e que são fundamentais para o funcionamento de subsistemas da organização. Um usuário nessa situação visualiza o novo software como uma ameaça ao seu domínio porque, enquanto somente ele souber e puder realizar determinados procedimentos, a sua importância e, conseqüentemente, segurança estarão mantidas dentro da organização.
- O medo do desemprego: o sentimento de medo, por parte do usuário, de que a informatização, ou seja, o software a ser desenvolvido, venha a causar o seu próprio desemprego é bem provável. Além da existência de crises pontuais que assombram funcionários com ameaças de desemprego, também começa a ocorrer o fenômeno em âmbito

global de forma constante. Invariavelmente tem-se atribuído a origem deste problema ao desenvolvimento científico e tecnológico. Dessa forma, o usuário tem encontrado cada vez mais motivos para ser contrário à informatização do seu trabalho ou setor.

A existência de um usuário altamente contrário ao processo de criação do software não é improvável. Esta possibilidade justifica, de forma cabal, a avaliação do usuário quanto à sua disposição como fonte de informação antes de se dar início à coleta de fatos.

De acordo com a teoria da hierarquia das necessidades de Maslow, as pessoas são movidas por cinco tipos distintos de necessidades conforme sua capacidade de motivar o comportamento: 1º necessidades fisiológicas; 2º necessidade de segurança; 3º necessidade de afeto; 4º necessidade de estima e 5º necessidade de auto-realização (Hampton, 1990). Assim, aplicando-se a teoria da hierarquia das necessidades às causas de resistência do usuário, observa-se que a intensidade da contrariedade pode ser avaliada em função da relação que as próprias causas possuem com as necessidades de Maslow (Tabela 1).

Logo, para toda e qualquer causa de resistência identificada, pode-se estipular a tendência de intensidade da contrariedade do usuário ao processo de criação do software. O nível de resistência do usuário depende das particularidades de cada caso e de cada pessoa envolvida. Logo, ressalta-se que a indicação do nível de intensidade de resistência para cada grupo de usuários, definidos em função das causas, pode apenas indicar uma tendência de intensidade ou de nível de contrariedade de cada usuário.

Tabela 1 – Intensidade de resistência vinculadas às necessidades de Maslow

Causa da resistência	Motivação do comportamento	Relação necessidades de Maslow	Provável nível de intensidade
Fatores políticos	Social	Auto-realização	Menor
Importância na organização	Social	Estima	Média
Medo do desemprego	Emocional	Segurança	Maior

4. Identificando a resistência do usuário

Atualmente, a Definição de requisitos implementada por métodos convencionais limita-se a identificar usuários ou pessoas envolvidas com o novo software e a identificar documentos ou sistemas existentes relacionados ao assunto em questão. A possibilidade de avaliar o usuário, seja do ponto de vista da motivação, seja do ponto de vista cognitivo ou de capacidade de comunicação, não é considerada em tais abordagens.

Demonstrada a existência de usuários com potencial diferenciado para participar da elicitação, em razão, primeiramente, da possibilidade de resistência ao processo de criação do software e, num segundo nível, das diferenças de intensidade de resistência dos mesmos, constata-se a necessidade da avaliação desses na perspectiva de exploração de fontes de informação.

Naturalmente, a resistência que provém de qualquer das causas apontadas neste trabalho é muitas vezes disfarçada de maneira criativa pelas pessoas. Até porque a maioria delas não teria como, consciente ou inconscientemente, argumentar abertamente a favor de um interesse próprio, limitado, em comparação a um interesse maior da organização. Ao invés disso, elas tendem a racionalizar ou buscar justificativas à sua posição em termos mais convincentes para os outros, do que para si mesmas (Hampton, 1990).

Assim, apesar da reconhecida dificuldade de predição de comportamento (Davidoff, 1983), ou seja, de se constatar antecipadamente a atitude de resistência do usuário, é importante que se procurem tendências ou indícios de contrariedade no grupo daqueles que participarão do desenvolvimento do software como fontes de informação. Para tanto, não existe um teste ou regras fixas porque cada indivíduo é um ser único; porém, algumas heurísticas, fundamentadas nas linhas deste trabalho, podem ser importantes para auxiliar os engenheiros de software na tarefa:

- Procurar, num primeiro momento, não deixar que os usuários obtenham a exata compreensão do objetivo da entrevista ou reunião.

Essa primeira heurística busca evitar que o usuário conheça antecipadamente o objetivo do engenheiro de software de identificar possibilidades de resistência na sua pessoa.

- Questionar, inicialmente, de forma dissimulada, os usuários sobre as possibilidades de resistência.

Essa heurística visa que o usuário não perceba, no transcorrer do primeiro conjunto de perguntas para a identificação de possibilidade de resistência, o objetivo do engenheiro de software. Dessa forma, procura-se preservar a sinceridade nas respostas do usuário.

Questões para a execução da segunda heurística:

- 1- Você acha que, no momento, deve-se manter a forma e o método de execução de suas tarefas ou trabalho na organização?
- 2- As disputas entre colegas ou questões políticas podem prejudicar o seu desempenho na organização?
- 3- Você tem encontrado dificuldade de relação com seus colegas?
- 4- Você acredita ser de importância fundamental para a organização?
- 5- Existe algum procedimento que só você domina dentro da organização?
- 6- Você tem alguma prevenção em repassar conhecimentos para colegas de serviço?
- 7- Você atribui alguma parcela de culpa ao desenvolvimento tecnológico pelo desemprego?

- Questionar, numa segunda rodada de entrevistas, de forma direta e objetiva, os usuários sobre as possibilidades de resistência.

A terceira heurística busca reafirmar e validar as respostas obtidas nas questões realizadas sem o conhecimento pelo usuário do objetivo da entrevista: identificar a disposição do usuário em relação ao processo de criação do software.

Questões para a execução da terceira heurística:

- 8- Mudanças na forma de trabalho no momento atual seriam inadequadas?
- 9- Na sua opinião, computadores mais prejudicam do que auxiliam no trabalho?
- 10- Você tem receio da possibilidade de mudança do seu sistema atual de trabalho por um sistema de computador novo?
- 11- A implantação de um sistema de computador no seu setor pode lhe prejudicar ou atrapalhar em relação às suas pretensões na organização?
- 12- Você tem receio de que a implantação de um sistema de computador no seu setor reduza sua importância e relevância dentro da organização?
- 13- Você tem medo de que a implantação de um sistema de computador no seu setor ocasione demissões?

- Procurar identificar tendência de resistência ou não no usuário através da observação de coerência entre as respostas dadas às questões dissimuladas com as questões realizadas de forma direta e objetiva.

Considerando-se que apenas respostas afirmativas para ambas as formas de questões (dissimuladas e diretas) formuladas para uma determinada causa de resistência apontam para uma maior tendência de o usuário apresentar contrariedade ao processo de criação do software, essa heurística visa definir a avaliação do usuário.

Por exemplo, se um usuário responder a todas as perguntas sobre *questões políticas* afirmativamente (questões de números 2 e 3 –dissimuladas- e questão 11 –diretas-), concluir-se-á então, pela existência da possibilidade de esse apresentar resistência ao processo de criação do software em razão das disputas políticas existentes dentro da organização. Por outro lado, se as respostas forem todas negativas, concluir-se-á sobre a tendência de o usuário não apresentar resistência, no aspecto em questão, ao processo de criação do software. Dessa forma, aplicando-se todas as questões dentro dessa lógica para as outras causas de resistência (Tabela 2), identificar-se-á a tendência de resistência ou não do usuário.

Tabela 2 – Questões relacionadas as causas de resistência

Causa de resistência	Questões dissimuladas	Questões diretas
Receio a mudanças	1	8, 9 e 10
Fatores políticos	2 e 3	11
Importância na organização	4, 5 e 6	12
Medo do desemprego	7	13

- Procurar identificar, inclusive com os donos do sistema, usuários com histórico de apresentação de idéias inovadoras, disposição para novas tarefas e relações solidárias.

Essa última heurística busca reafirmar a identificação de usuários sem tendências de resistência ao processo de criação do software observadas pelas heurísticas anteriores.

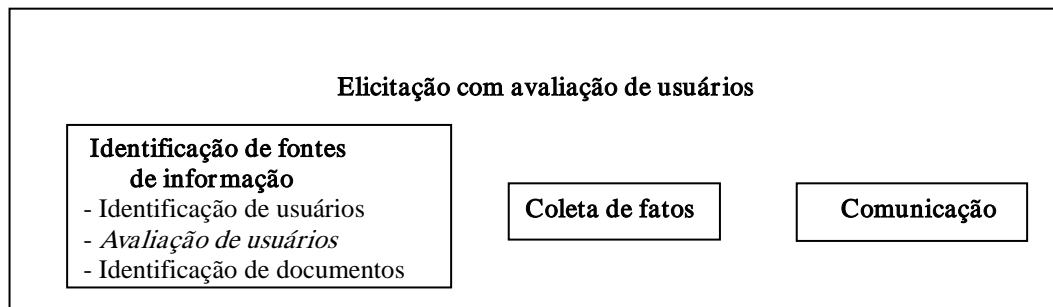


Figura 1 – Elicitação com avaliação do usuário

Com essas heurísticas, os engenheiros de software podem, por meio de entrevistas, identificar os usuários com possibilidades de maior ou menor potencial de contribuição para o processo de criação do software. Elas servem também para reduzir o problema de comunicação entre os engenheiros de software e os usuários em razão da resistência dos últimos ao novo sistema. Porém, elas propõem, fundamentalmente, uma abordagem de avaliação do usuário para a tarefa de identificação de fontes de informação (Figura 1).

5. Tratando a resistência do usuário

Dentro de uma organização, o que pode ser mudado são os métodos e equipamentos de trabalhos, as normas, a própria estrutura, as práticas de controle, os padrões de comunicação entre as pessoas, o projeto do trabalho, as técnicas de planejamento e muitos outros aspectos (Hampton, 1990). Apesar de o processo de desenvolvimento de software implicar, imperativamente, algumas das mudanças citadas, tal como no caso da implantação de novos equipamentos ou métodos de trabalho, a resistência - fruto dessas mudanças - à visão da engenharia de software requer análise e tratamento diferenciados dos desenvolvidos pela perspectiva da área da administração de empresas. Isso se deve principalmente à diferença existente na relação hierárquica entre o agente responsável pela implementação da mudança e as pessoas que serão atingidas por ele dentro dos dois contextos. Geralmente, o funcionário envolvido num processo de mudança implementado por chefes ou superiores não encontra muitas alternativas para impor sua resistência a tal, porque, em última instância, ele depende muito mais da organização, ou seja, do superior que coordena o processo, do que a organização depende dele para a efetivação, com sucesso, da própria modificação. Essa dependência do funcionário em relação ao superior está, na maioria das vezes, ligada à necessidade de segurança e estabilidade daquele no emprego.

Já, durante um processo de criação de software, que, naturalmente, ocasiona uma mudança significativa dentro da organização e no qual o engenheiro de software representa a figura do responsável pela implementação das modificações, o funcionário e usuário, além de não possuir o *status* de subordinado, geralmente se descobre numa situação privilegiada por ser de fundamental importância para o sucesso do processo, já que essa depende do seu conhecimento. Esse fator, por si só, pode aumentar a possibilidade de problemas devido à constatação por parte do usuário de que é uma boa oportunidade para se autovalorizar. Dessa forma, no contexto da engenharia de software, a resistência ao processo de mudança ocorre numa perspectiva diferenciada da estudada prioritariamente pela área de administração de empresas, porque nela o funcionário assume maior poder de influência no desenvolvimento do próprio processo.

Diante de um quadro em que um funcionário, como usuário, apresenta resistência a um processo de mudança no qual métodos ou equipamentos serão substituídos pela implantação de um novo sistema informatizado, uma primeira decisão se faz necessária. Ela consiste em desistir do usuário ou decidir-se por aplicar uma atividade ou abordagem com o objetivo de superar a sua resistência. A opção pela desistência de envolver ou por utilizar o conhecimento e experiência de determinada pessoa deve ser considerada em contextos nos quais se apresentam quadros de alta resistência, viabilidade de substituição ou requisito de tempo. Uma pessoa que demonstra explicitamente sua contrariedade a um determinado processo de mudança dificilmente poderá ser convencida em curto espaço de tempo da adequação de sua colaboração à implementação da própria mudança. Na maioria dos processos de criação de software, o cumprimento de cronograma e, conseqüentemente, a conclusão das etapas nos prazos previstos são tomados como critério para a avaliação da evolução e viabilidade do próprio projeto, ou seja, atrasos podem comprometer a confiança de todos os envolvidos, sejam os donos do sistema, seja a equipe ou o engenheiro de software responsável pela definição de requisitos. Nesse sentido, não parece adequado que todo um projeto maior fique paralisado no seu desenvolvimento apenas em razão da busca de superação da resistência de um determinado usuário.

A desistência de obter colaboração de um usuário é mais premente num contexto em que existe a possibilidade de se envolver outra pessoa que tenha conhecimento sobre as atividades e tarefas desenvolvidas, substituindo aquele que apresenta alta resistência ao processo de criação do novo software. Entretanto, a própria exacerbação de apenas um dos três fatores citados pode recomendar a desistência do engenheiro de software pelo envolvimento de determinado usuário. Mesmo que se possua prazos longos para conclusão da elicitação a comprovada intenção do usuário de dificultar o

processo de criação do software compromete o seu envolvimento no desenvolvimento do sistema. Essa decisão deve ser tomada, então, a partir da constatação de alta resistência do usuário, inclusive com a aplicação das heurísticas propostas neste trabalho, e da devida análise de contexto em relação a requisitos de cumprimento de prazos e impossibilidade de substituição do usuário.

5.1 Estratégias

A partir da constatação pelo engenheiro de software de que a superação da resistência de um determinado usuário mostra-se adequada, a busca por estratégias, através de abordagens ou métodos, para a solução de tal problema dentro de um contexto complexo, como o é o de desenvolvimento de software, faz-se necessária. Num nível superior de análise, dispõe-se de três estratégias a serem aplicadas com o intuito de superar a resistência do usuário, ou seja, o engenheiro de software pode optar entre as estratégias de convencer, manipular ou coagir. Cada uma apresenta vantagens e desvantagens e se faz adequada diante de uma especificidade de quadro que a resistência possui e da forma como se apresenta (Tabela 3).

É lógico que a possibilidade de convencimento do usuário para sua colaboração com o processo de criação do software se apresenta como a alternativa preferencial. Dentro dessa linha de tratamento, podem-se aplicar métodos na perspectiva de participação ou negociação com o usuário, os quais apresentam uma grande vantagem por viabilizarem a conquista da colaboração do usuário, envolvê-lo num ambiente de sinceridade e possibilitar a melhora na qualidade da própria atividade. Numa situação em que o engenheiro de software encontra-se em condições adversas para obter, tal confiança, essa estratégia se faz ainda mais adequada; por outro lado, o convencimento pode exigir muito tempo, ser dispendioso e, ainda, falhar.

A estratégia de manipulação pode ser aplicada quando o convencimento se demonstra ineficiente ou dispendioso. Ela pode ser rápida, porém, seja através da alternativa de omissão seja através de inverdade, o método pode ocasionar futuros problemas quando da conscientização por parte do usuário de sua manipulação.

Por sua vez a estratégia de coerção, tanto explícita como implícita, faz-se adequada principalmente quando a rapidez é essencial. A sua viabilidade só é possível pelo uso do poder, o que requer, de alguma forma, a intervenção de superiores do usuário. Essa alternativa para superação da resistência pode ser rápida e até vencer alguns tipos de contrariedade, entretanto é muito arriscada por ocasionar, invariavelmente, problemas futuros de comportamento e relacionamento com o usuário. E mais, como o superior não estará sempre presente no trabalho desenvolvido pelo engenheiro de software e o usuário, a situação de conflito e constrangimento pode, até, inviabilizar a tarefa de elicitación.

Tabela 3 - Estratégias para superação da resistência do usuário

Estratégia	Vantagem	Desvantagem	Indicação
Convencer	Comprometimento do usuário e qualidade da atividade	Pode exigir tempo e falhar	Na maioria dos casos
Manipular	Pode ser mais rápida que o convencimento	Problemas futuros se o usuário sentir-se manipulado	Quando o convencimento parecer ineficiente
Coagir	Rapidez	Risco quanto à eficiência, problemas de	Requisito de tempo

		comportamento e conseqüências futuras	
--	--	--	--

A decisão do engenheiro de software por uma das estratégias de superação deve ser pautada não só pelas vantagens e desvantagens de cada alternativa, mas, também, pela especificidade de cada caso/situação, sobretudo levando em conta a importância do usuário no contexto do sistema.

Com base nas considerações expostas, podem-se dispor as estratégias numa ordem preferencial para aplicação. A opção pelo convencimento, além de satisfazer todo e qualquer requisito ético, apresenta a grande vantagem de possibilitar um aumento de qualidade no resultado final da tarefa de elicitação; por isso, apresenta-se como a alternativa preferencial de implementação para o engenheiro de software. Já as outras duas estratégias, de manipulação e coerção, em virtude de seus riscos em relação às conseqüências futuras e métodos comprometedores, do ponto de vista ético, devem ser consideradas apenas em situações extremas. Além disso, ainda que se faça necessária a aplicação de uma delas, pelo alto risco e, principalmente, pela relação conflitante que possivelmente desencadeará a coerção deve ser preterida.

5.2 Abordagens

Dentro de cada uma das estratégias consideradas, abordagens diferenciadas, com métodos e características próprias, podem ser consideradas (Tabela 4). Com o intuito de convencimento do usuário, duas abordagens surgem como as mais adequadas: através da participação ou através da negociação e acordo. A abordagem de participação pode ser implementada sobre as perspectivas de representatividade ou de comprometimento das pessoas suscetíveis à resistência.

Propostas desenvolvidas por estudiosos e pesquisadores da própria área da engenharia de software enquadram-se perfeitamente dentro da abordagem de participação. A metodologia JAD, por exemplo, é uma metodologia de desenvolvimento de software que envolve o usuário numa perspectiva de participação representativa, responsabilizando-o por parte do resultado final do trabalho desenvolvido. Já as metodologias, classificadas como PD, que buscam também aumentar a satisfação das pessoas envolvidas e a eficiência das tarefas implementadas, envolvem o usuário numa perspectiva de maior comprometimento, já que o usuário sente-se inteiramente responsável pelo resultado do trabalho desenvolvido.

Entretanto, essas abordagens exigem o envolvimento de um número mínimo de pessoas; requerem habilidades pessoais, como do líder do JAD, por exemplo; muitas vezes, demandam tempo e, também, exigem o comprometimento de todos os envolvidos, o que, no caso de alta resistência do usuário, fica difícil de se obter.

Tabela 4 – Abordagens das estratégias para superação da resistência

Estratégia	Abordagem	Exemplo
Convencer	Participação	Representatividade (JAD) ou Com comprometimento (PD)
	Negociação e acordo	Reuniões esclarecedoras e cursos amigáveis de informática
Manipular	Omissão	Etnografia com omissão
	Inverdade	Cooptação com justificativas inverídicas
Coerção	Poder	Coerção explícita do superior e do engenheiro de software sobre o usuário

A abordagem de negociação e acordo, com a intenção de convencimento do usuário, faz-se adequada sobretudo para situações nas quais a resistência tem origem numa causa específica e o usuário pode ser conscientizado da sua improcedência ou de que os benefícios resultantes do desenvolvimento do novo software serão bem maiores que os custos, tanto para a empresa como para ele próprio. Uma reunião na qual o usuário discuta com seus superiores, e até mesmo com o engenheiro de software, todas as implicações que a implantação do novo software acarretará pode ser eficiente se desenvolvida num ambiente de sinceridade e franqueza.

O esclarecimento dos reais planos da organização para o aproveitamento futuro do usuário, ou uma proposta de qualificação profissional, através de cursos amigáveis de operação de computador para a aplicação no novo software, podem ser alternativas adequadas para a superação da resistência de um usuário inseguro em relação a sua estabilidade na organização. Por outro lado, essas abordagens ficam restritas a situações de real interesse de aproveitamento a longo prazo, pela organização, dos serviços do usuário e da existência de confiança por parte dele nos seus superiores e, por conseguinte, na organização.

Dentro da estratégia entendida como de manipulação, surgem duas abordagens aplicáveis para a superação da resistência do usuário, ambas questionáveis do ponto de vista ético. A utilização da etnografia, que busca descobrir, por meio de observações e participação do engenheiro de software no trabalho e na rotina do usuário, pode ser entendida como uma abordagem de manipulação se o seu real objetivo for omitido. O engenheiro pode acompanhar por um bom tempo a rotina do usuário e descobrir muitas facetas e detalhes que nem mesmo numa entrevista sem resistência seriam detectadas sem revelar o seu real objetivo. O problema é que tal abordagem possivelmente provoca desconfiança no usuário durante a implementação do método e pode comprometer a atividade.

Outra abordagem considerada como de manipulação pode ser aplicada através de colocações inverídicas em reuniões sobre o real objetivo da organização em relação ao aproveitamento futuro do usuário e em relação ao objetivo da etnografia. Essas abordagens, entretanto, como já se referiu, podem comprometer de forma significativa o ambiente futuro do usuário dentro da organização e, ainda, dependem da aceitação tanto do engenheiro de software como dos superiores na organização da implementação de métodos questionáveis do ponto de vista ético.

Uma abordagem que implemente a estratégia de coerção é altamente questionável, não só sob o aspecto ético mas, também, do da eficiência, principalmente no contexto da engenharia de software, por envolver, obrigatoriamente, superiores do usuário resistente à sua implementação. A concessão de autoridade ao engenheiro de software em relação ao usuário por deliberação da organização pode dificultar a comunicação entre aquele e este, que não aceita a subordinação. A implementação dessa forma de abordagem não é nada simples, uma vez que pressupõe a aceitação da própria organização da concessão de considerável autonomia a um estranho e, ainda, desconsidera o funcionário, que tem uma relação bem mais próxima e antiga com a empresa. Nesse contexto, a possibilidade de insucesso na tarefa de elicitação aumenta consideravelmente, e o próprio futuro do usuário dentro da organização pode ficar comprometido.

5.3 Causas x abordagens

Antes de um engenheiro de software decidir-se por uma ou outra abordagem para superação de resistência do usuário ao processo de criação do novo software, muitos fatores devem ser considerados para que a sua opção seja a mais adequada dentro do contexto em questão. Características pessoais do usuário resistente, como, por exemplo, a sua personalidade; necessidade de cumprimento de prazos para a definição de requisitos; importância do usuário para a organização e no contexto do sistema e a necessidade de aceitação por parte dele do novo software são questões básicas que sempre devem ser

consideradas pelo engenheiro antes da definição por uma ou outra abordagem. Outro fator que deve ser considerado é o da análise da causa da resistência do usuário. Ressaltando-se as abordagens ligadas à estratégia de convencimento, por ser a mais indicada, há algumas indicações de tratamento específicas para cada causa de resistência (Tabela 5).

Tabela 5 – Indicações de abordagens para as causas da resistência

Causa	Tratamento	Exemplo
Medo de mudança	Negociação e acordo	Cursos amigáveis de informática
Fatores políticos	Participação	JAD ou PD
Receio de perda de importância	Participação	JAD ou PD
Medo de desemprego	Negociação e acordo	Reunião esclarecedora

Como bom exemplo dessa consideração, pode-se analisar o caso de um usuário que resiste ao processo de criação de novo software pelo receio de que a implantação do sistema informatizado venha a ocasionar o seu desemprego. Se a organização realmente possui projetos futuros para a pessoa resistente, a indicação de uma simples reunião com a explanação dos objetivos da organização para com o usuário é muito mais adequada, ou seja, nesse contexto, uma abordagem de negociação e acordo provavelmente seja mais eficiente do que uma abordagem de participação com suas demandas de pessoal e tempo.

6. Conclusão

A expectativa das vantagens a serem obtidas com a aplicação prática das heurísticas e abordagens propostas é principalmente demonstrar a resistência de um usuário e a indicação para superação dessa no contexto do desenvolvimento de software em organizações. Em razão de ser interdisciplinar, o trabalho não abordou de maneira completa, conforme os padrões da psicologia, a validação das heurísticas propostas.

Para trabalhos futuros surge a proposta de avaliação do usuário do ponto de vista da motivação, da capacidade cognitiva e da comunicação isto porque não é adequado que se inicie a elicitação se não se sabe exatamente como se deve proceder para se obter o máximo do conhecimento do usuário.

Entre as conclusões que se obteve com o desenvolvimento do trabalho estão que: a) num quadro de resistência ligada, principalmente, a necessidade de segurança a qualidade da comunicação e das informações do usuário ficam profundamente comprometidas. b) as heurísticas propostas podem auxiliar o engenheiro de software aproximar-se o máximo possível da verdade em relação a motivação do usuário com o processo de produção do software. c) as abordagens cooperativas contribuem mas não resolvem o problema da resistência principalmente porque requerem comprometimento o que a própria resistência inviabiliza. d) fatores como personalidade do usuário, demanda de tempo, importância do usuário para o sistema e as próprias causas da resistência são os principais elementos para indicar uma abordagem de superação da resistência e e) as estratégias de convencimento devem sempre ser priorizadas visto que satisfazem a ética e ainda aumentam a qualidade nos resultados da elicitação e evita problemas futuros decorrentes do tratamento da resistência.

Espera-se que com a divulgação deste trabalho a avaliação do usuário torne-se objeto de atenção e atividade por parte dos engenheiros de software na definição de requisitos para uma efetiva melhoria da qualidade do software desenvolvido.

Referências bibliográficas

ALENCAR, Eunice M. L. S. **Psicologia – introdução aos princípios básicos do comportamento.** Editora Vozes Ltda, Petrópolis, 1976.

DAVIDOFF, Linda L. **Introdução à psicologia.** McGraw-Hill do Brasil Editora Ltda, São Paulo, 1983.

HAMPTON, David R. **Administração – comportamento organizacional.** McGraw-Hill do Brasil Editora Ltda, São Paulo, 1990.

LEITE, Júlio C. S. P. **Engenharia de requisitos: notas de aula.** Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 1994.

PARRY, John. **Psicologia da comunicação humana.** Editora Cultrix Ltda, São Paulo, 1972.

PRESSMAN, Roger S. **Engenharia de software.** McGraw-Hill do Brasil Editora Ltda, São Paulo, 1995.

WATZLAWICK, Paul; BEAVIN, Janet H.; JACKSON, Don D. **Pragmática da comunicação humana.** Editora Cultrix Ltda, São Paulo, 1993.

ZANLORENCI, Edna P.; BURNETT, Robert C. **Modelo para qualificação da fonte de informação do cliente e de requisito funcional.** Workshop de engenharia de requisitos 1998 (WER'98). Disponível por www em <http://www.inf.puc-rio/~wer98> (12 outubro 1998).

ZIRBES, Sérgio Felipe. **Análise da contribuição que sistemas baseados em conhecimento e sistemas de apoio a decisão em grupo oferecem durante a elicitação dos requisitos de sistemas.** Porto Alegre : cpgcc da ufrgs, 1993.