

Elicitação de Requisitos
Julio Cesar Sampaio do Prado Leite

RESUMO: Apesar do crescente interesse sobre a análise de requisitos, observa-se que a maioria dos trabalhos e métodos apresentados dirigem sua atenção para o problema da modelagem dos requisitos, deixando sempre em aberto o problema fundamental que é exatamente o que modelar. Nosso artigo aborda justamente este problema. A absorção, pelos engenheiros de software, das necessidades reais dos futuros usuários do software a ser produzido, é o problema central da elicitação de requisitos.

PALAVRAS-CHAVE: Engenharia de Software, Análise de Requisitos, Elicitação de Requisitos, Análise de Sistemas, Aquisição de Conhecimento.

Introdução

Análise de requisitos é um termo estabelecido em engenharia de software, para identificar a primeira das fases na construção de software. Apesar do crescente interesse acerca da análise de requisitos, observa-se que a maioria dos trabalhos e métodos apresentados dirigem sua atenção para o problema da modelagem dos requisitos, deixando sempre em aberto o problema fundamental que é exatamente o que modelar. Em particular cresce o interesse pela tarefa de elicitação dos requisitos, ou seja de tornar explícito os desejos, as intenções e necessidades dos clientes em relação ao software a ser construído.

Os métodos de análise estruturada e seus correlatos já são de grande conhecimento do profissional de informática, apesar de ainda pouco praticados. A disponibilidade de ferramentas CASE, certamente darão um novo impulso a padronização desses métodos em uma série de empresas. Sem dúvida a automação dos gráficos e tabelas dos métodos de análise estruturada pelas ferramentas CASE, aliaadas a alguma possibilidade de verificação das representações utilizadas, aumentará a produtividade dos grupos de desenvolvimento de Software. No entanto, nenhuma dessas ferramentas baseadas nos métodos estruturados se ocupa do problema fundamental que é o de realmente identificar quais são os requisitos do software a ser produzido.

Durante algum tempo, a área de engenharia de software vem concentrando seus esforços na representação de requisitos, especificações e projetos. Várias propostas de formalização das mais formais a menos formais vêm sendo apresentadas pela comunidade de pesquisa em engenharia de software [Freeman 88]. Atualmente alguns grupos de pesquisa em várias universidades e centros de pesquisa [Finkelstein 89], vem estudando o problema de elicitação de requisitos. Isto é a absorção das reais necessidades dos futuros usuários do software a ser produzido. Estes estudos vêm sendo bastante influentes devido aos resultados alcançados pela Inteligência Artificial na área de aquisição de conhecimentos.

O termo Elicitação de Requisitos refere-se ao processo de elicitar. Este neologismo se justifica porque trata-se de um subprocesso da análise de

requisitos em que a preocupação fundamental não é a de modelar, mas a de coletar, validar e tornar explícito os requisitos do software a ser construído.

O grande problema que o engenheiro de software encontra nesta tarefa de elicitar requisitos, é que normalmente não há disponível quer seja na sua formação acadêmica, quer na sua formação profissional um tratamento mais preciso para esta árdua tarefa. Naturalmente vale lembrar que na maioria dos casos, a elicitação de requisitos é deixada a cargo do "boom analista de sistemas" que segundo as listas de qualidades na literatura [Pressman 88, Daniels 71] mais parece super-homens.

Apesar desse fato, existe, disperso na literatura, principalmente na literatura de sistemas de informação, uma série de métodos e pseudo-métodos para a tarefa de elicitação. Também cada vez mais se observa que os pesquisadores de Inteligência Artificial, sabidamente experimentalistas, estão publicando resultados de interesse para a tarefa de elicitação. Em Inteligência Artificial esta área é chamada de aquisição do conhecimento.

Nosso artigo apresentará uma caracterização da análise de requisitos orientada a processos, bem como apresentará as propostas que vem sendo estudadas para o problema da elicitação. Nosso objetivo é, através de uma caracterização do processo da análise de requisitos, propor uma classificação dos diferentes processos que ocorrem durante a análise de requisitos. Portanto a maior ênfase é um levantamento das várias tarefas que no nosso entender compõem a análise de requisitos. Não é nosso objetivo propor uma metodologia de elicitação de requisitos. Primeiro, porque acreditamos que uma metodologia de elicitação deve ser orientada ao problema em questão. Segundo, porque nosso artigo não se propõe a ser um manual, e sim apresentar conceitos e fundamentos para aprofundar os conhecimentos sobre a elicitação.

Análise de Requisitos

A análise de requisitos não se processa no vácuo. Nossa perspectiva de sistemas [Freeman 87], identifica a análise de requisitos como uma parte do sistema de construção de software, cujo o produto, o software, é um subsistema de um sistema maior. Este sistema maior, onde o software vai atuar define o que chamamos de universo de informações. "Universo de informações é o contexto geral no qual o software deverá ser desenvolvido. O universo de informações inclui todas as fontes de informação e todas as pessoas relacionadas ao software.

Estas pessoas são também conhecidas como- os atores desse universo de fatos. O universo de informações é a realidade circunstanciada pelo conjunto de objetivos definidos pelos que demandam o software."

O uso da teoria geral de sistemas, como arcabouço básico para o entendimento do contexto da análise de requisitos, se fundamenta na proposta dessa teoria de servir como um fator integrador de diversas áreas de conhecimento. O S

oftware é sempre um subsistema. Dependendo do caso, o software pode ser: um subsistema de um sistema de computação (hardware (+) software), um subsistema de uma determinada aplicação num sistema de informação numa organização, um subsistema de controle de frenagem em um automóvel, ou um subsistema do processo de produção de software. O universo de informações é exatamente o resultado da aplicação dessa visão sistêmica, como objetivo de facilitar o trabalho de compreensão e detalhamento do que deve ser feito.

O universo de informações (Udi), já delimitado pelo sistema mais abrangente, não implica que a informação necessária para o que deve ser feito pelo software já esteja disponível. No entanto, o Udi restringe as fontes de onde estas informações serão elicítadas. Esta visão da análise de requisitos, ocorrendo a posteriori da definição do universo de informações, restringe seu campo de atuação. Esta restrição é equivalente ao que Davis [Davis 82] chama de segundo nível na determinação de requisitos. Davis entende que a determinação de requisitos ocorre em dois níveis: requisitos organizacionais e requisitos da aplicação.

Outro ponto importante é o reconhecimento da necessidade de elicitação. Isto é baseado no senso comum de que antes de fazer algo, o primeiro passo é conhecer e entender o que deve ser feito. Como diz a Von Neumann: "There is no sense in being precise about something when you do not even know what you are talking about."

Afirmar o óbvio nem sempre é tarefa simples. Muito da confusão reinante, na literatura de engenharia de software, sobre os termos "requisito" e "esp ecificação" [Leite 87] provém justamente da falta de entendimento do óbvio frase citada acima. Sem entrar em maiores detalhes citaremos as definições de cada palavra segundo o dicionário:

- Requisito: Condição necessária para a obtenção de certo objetivo, ou para o preenchimento de certo objetivo.
- Especificação: Descrição minuciosa das características que um material, uma obra ou um serviço deverão apresentar.

Levando em consideração estes pontos, isto é, que requisitos pertencem a um sistema maior, e da necessidade de elicitação, propomos a seguinte caracterização da análise de requisitos.

"Análise de requisitos é um processo em o que deve ser feito é elicitado e modelado. Este processo deve lidar com diferentes pontos de vista, e usa uma combinação de métodos, ferramentas e atores. O produto desse processo é um modelo, do qual um documento, chamado requisitos, é produzido."

Em seguida detalharemos esta visão da análise de requisitos como um processo, bem como descreveremos cada um de seus subprocessos.

0 Processo da Análise de Requisitos

Conforme a caracterização orientada a processo, o processo de elicitação é diferenciado do processo de modelagem dos requisitos. O processo de elicitação compreende os processos de: coleta de fatos, validação dos fatos e comunicação. O processo de modelagem compreende os processos de: organização e representação. Apesar da dificuldade de se separar estes processos, já que a análise de requisitos é um emaranhado desses processos, esta classificação nos auxilia na tentativa de melhor compreender esta tão difícil tarefa. A Figura 1 nos dá a classificação utilizada.

É durante a elicitação que o conhecimento do universo de informação de uma determinada aplicação é adquirido, para que mais tarde possa ser modelado. Os subprocessos coleta de fatos, validação de fatos e comunicação estão bastante interligados. A coleta de fatos procura enumerar fatos que fielmente traduzam o mundo real, evitando ao máximo interpretações. A validação é a maneira a com que se confirma ou não a acuidade do processo de coleta. O processo de comunicação permitia a coleta e a validação, já que é fundamental o diálogo entre vários atores presentes na elicitação.

Modelagem é o processo responsável pela caracterização dos resultados obtidos pela elicitação de requisitos. Modelagem, não só produz o modelo final, mas também sua forma formatada, isto é os requisitos.

O processo de modelagem é composto de duas partes: representação e organização. Para que seja produzido um modelo dos fatos elicitados, os atores usam métodos e ferramentas. Um modelo é composto de um esquema de representação, que são organizados segundo as políticas organizacionais do modelo. Um modelo deve ser capaz de capturar pontos de vista diferentes.

Pontos de vista refletem diferentes "backgrounds": quer dos atores responsáveis pela análise de requisitos, como também por aqueles atores participantes ao Udi e que estão envolvidos de alguma forma com a aplicação. Método é um conjunto de técnicas e procedimentos. Ferramentas fornecem apoio no uso de um determinado método.

O modelo SAOI [Ross 77] apresentado na Figura 2, reflete esta caracterização e procura clarificar os diferentes relacionamentos entre as partes e os subprocessos da análise de requisitos. O ponto de vista usado neste modelo SAOI é o de identificar relacionamentos que podem existir no processo de análise de requisitos, ele não aborda a operação desse processo.

Devemos ressaltar, mais uma vez, que o processo de produção de requisitos é bastante entrelaçado. Isto é, é difícil realmente separar cada um desses subprocessos, que interagem continuamente. Por causa disso, várias interações ocorrem antes que a versão final do modelo e os requisitos sejam produzidos. A taxonomia apresentada tem por objetivo explicitar conceitos e identificar componentes, que apesar do interrelacionamento muito estreito, tem características únicas.

Nas seções que se seguem usaremos esta caracterização de processo, para abordar algumas heurísticas relativas ao processo de elicitação. Neste artigo não examinaremos a parte referente a modelagem apesar de sua íntima relação com todo o processo. Conviém observar que a análise de requisitos, por justamente ser a fase do desenvolvimento de software que procura traduzir o mundo informal para notações formais, tem uma dificuldade inerente a este processo, e portanto depende fortemente de comportamentos heurísticos. Não é nosso objetivo neste artigo, definir um método, entendido como um conjunto de métodos e ferramentas, mas sim apontar pontos fundamentais que devem ser observados no processo de elicitação.

Coleta de Fatos

A coleta de fatos se realiza no ambiente do universo de informações, onde atuam vários grupos de atores sujeitos a uma série de regras e padrões. Estes atores podem ter diferentes pontos de vista, alguns deles até conflitantes. Este meio social é a fonte que os engenheiros de software têm para conhecer e entender a necessidade funcionalidade do software, bem como as restrições que serão impostas ao produto software.

Coletar fatos, pressupõe que o engenheiro de software se atenha aos fatos evitando interpretações. Cabe ao engenheiro reportar no melhor estilo jornalístico de objetividade os resultados de seu trabalho de coleta. Várias são as técnicas disponíveis para este trabalho, como por exemplo:

- Observação,
- Leitura de documentos do Udi,
- Entrevistas,
- Questionários,
- Análise de protocolos,
- Participação ativa dos atores do Udi,
- Enfoque antropológico dos engenheiros de software.

A combinação destas técnicas e seus procedimentos associados

em um conjunto harmonioso, é fundamental para o sucesso da tarefa de coleta de fatos. Portanto é importante frisar que não existe um único método para a coleta de fatos. Existem sim, uma série de heurísticas e de ferramentas aliadas a estas técnicas. Também existe, na literatura, uma série de heurísticas de como combinar estas várias técnicas.

Por exemplo, no caso da técnica de observação pode-se usar a observação participativa, ou observação transparente, ou seja uma observação onde há uma interação sem interação. No caso de entrevistas, várias outras técnicas podem ser usadas, desde da entrevista camuflada, até uma entrevista formal com perguntas já

previamente divulgadas. Diferentes heurísticas de preparação e aplicação de questionários também estão disponíveis. No caso de análise de protocolos, uma técnica desenvolvida no âmbito de Inteligência Artificial, há também uma série de estratégias que podem ser adotadas [Ericsson 84].

A combinação de procedimentos e técnicas vai definir o método a ser utilizado na coleta de fatos. Este método poderá ser apoiado por ferramentas,

como por exemplo: no caso de um método fortemente baseado em questionários e entrevistas formais, o "check-list" pode ser uma ferramenta bastante útil. Outro exemplo de ferramenta para suportar este método poderia ser um editor interactivo que permitisse que os questionários fossem respondidos num terminal de computador.

Em nossa trabalho de pesquisa [Leite 89] temos investigado uma variação

da técnica de entrevistas baseada no conceito de "vocabulário de aplicação", no qual procura-se respeitar a forma sintática das palavras chave usadas no jargão do universo de informação já previamente definido. Nossos resultados [Leite 90] vêm comprovando a eficácia deste conceito.

Novamente faz-se necessário observar que a combinação dessas técnicas para formação de um método de coleta de fatos, não só é dependente do universo de informações, como também tem uma forte interação com as técnicas usadas tanto para Comunicação com o usuário quanto para Comunicação com o usuário. Na próxima seção falaremos sobre técnicas disponíveis para a comunicação entre o engenheiro de software e os atores do Udi.

Comunicação

A comunicação é o sub-processo no qual ocorrem as interações entre os engenheiros de software e os outros atores do universo de informações. Apesar de sua grande interação e integração como a coleta de fatos e a validação é possível identificar técnicas e procedimentos com os quais o engenheiro de software pode contar para um melhor desempenho na tarefa de elicitação.

Convmém observar que o processo de comunicação é um canal em que há duplo tráfego: isto é os atores do Udi se comunicam com os engenheiros de software e estes com os atores do Udi. Apesar dessa via dupla, acreditamos ser do engenheiro de software a responsabilidade maior, ou pelo menos a liderança do processo de comunicação. Com isto queremos ressaltar que cabe ao engenheiro de software os encargos de tornar a comunicação engenheiro de software -> atores do Udi mais clara e limpa possível. Dentre as técnicas disponíveis destacamos:

- Apresentação,
- Negociação,
- Linguagem da Aplicação,
- Participação,
- Treinamento.

A comunicação entre engenheiro de software e outros atores, torna-se mais importante na medida em que a validação dos fatos coletados depende, fundamentalmente, dessa comunicação. Em função do processo de validação, que produz uma série de observações sobre os fatos coletados, o processo de comunicação, usando uma combinação de técnicas procede a comunicação Delta. A comunicação Delta é um processo em que os engenheiros de software procuram dirimir, junto aos atores do Udi, dúvidas levantadas pelas observações geradas no processo de validação.

No que se refere a técnica de apresentação há uma série de heurísticas na literatura sobre melhores formas de apresentar fatos já coletados. É consenso que gráficos aumentam a comunicabilidade, e portanto o engenheiro de software encarrega

do do processo de elicitação deve conhecer diferentes estratégias de comunicação visual. Sobre apresentação convém lembrar o antigo provérbio chinês: "Uma boa figura vale 1000 palavras". O importante neste provérbio é justamente o adjetivo "boa", ou seja de nada adianta uma figura, esta tem que ser a figura certa para que possa ter a carga semântica de muitas palavras.

A negociação é a maneira pela qual, no universo de informações, se procurará resolver conflitos resultantes de pontos de vista contraditórios. Cabe ressaltar que ao engenheiro de software na fase de elicitação, cabe uma posição imparcial no processo de negociação entre os vários atores. Os conhecimentos de estratégia de negociação facilitam a compreensão de certos fatores em discussão. Em alguns casos pode ter o engenheiro de software o papel de facilitador entre as partes. Várias são as heurísticas e procedimentos relativos a problemas de negociação, sendo a literatura de administração e negócios é a melhor fonte de informação.

A técnica do uso da linguagem da aplicação [Leite 89] já foi nominada na seção anterior. O objetivo maior é de que ao se comunicar com o Udi, o engenheiro de software se utilize, ao máximo, do vocabulário desse Udi. É claro que para utilização desse vocabulário a coleta de fatos tenha se efetuado com base nesta técnica. Estudos recentes [Leite 90] têm demonstrado a eficácia desta técnica, que se utiliza entre seus procedimentos e estratégias, conceitos provenientes da área da semiótica.

A participação, conforme também já mencionamos em coleta de fatos, pode ser efetuada tanto quando os atores do Udi fazem parte, explicitamente, da equipe de elicitação conjuntamente com os engenheiros de software, ou como com uma visão antropológica, isto é quando os engenheiros de software "vivem" por algum tempo no universo de informações.

A participação ativa de atores do Udi na equipe de elicitação, diminui em muito a carga de trabalho dos engenheiros de software no que se refere a comunicação, isto porque a responsabilidade da elicitação passa também a ser dos atores não técnicos. Apesar de todas as dificuldades de integração de equipes multidisciplinares, esta técnica muitas vezes tem sucesso. No entanto esta técnica só obtém sucesso quando os atores sem a formação de engenharia de software, tiveram algum treinamento e quando estes atores estiverem efetivamente empenhados na tarefa. Como sabemos, a diferença cultural ent

re os membros de um time de elicitação participativo muitas vezes leva com que esta técnica não apresente os resultados desejados.

A participação antropológica parte do pressuposto que os atores do Udi, não tem como prioridade a tarefa de elicitação de requisitos, e que caberá aos engenheiros de software aprender o que realmente se passa no Udi. Nesta técnica os engenheiros de software passam a ser atores do Udi, convivendo com as tarefas e regras ali existentes. Esta participação antropológica, apesar de mais onerosa é bastante importante em grandes projetos. O acultramento do engenheiro de software se dá através do "training in the job". Utilizando-se esta técnica, o engenheiro de software perde a sua imparcialidade quanto ao Udi, já que como aculturado passa a ter seu próprio ponto de vista.

A técnica de treinamento parte do pressuposto que os vários atores do Udi devem ser aculturados do jargão e dos métodos próprios da engenharia de software. Esta técnica foi muito usada na implementação dos primeiros sistemas de informação e geralmente, só era empregada, já nas fases finais do processo de construção de software. Enquanto a informática era um "otem", principalmente antes do advento dos microcomputadores, esta estratégia não era refutada. Com a massificação da informática, buscou-se como paliativo trazer este treinamento para as primeiras fases do processo de construção do software.

Conforme vimos acima, a técnica de treinamento também pode ser aplicada na ordem atores do Udi -> engenheiros de software apesar de que este não é normalmente o caso. Apesar da utilidade do uso de técnicas de treinamento quando da elicitação, deve-se procurar processos onde possa haver um maior diálogo entre as partes envolvidas. A técnica de treinamento deve ser usada logo no começo do processo de elicitação para que haja por parte dos atores do Udi, no mínimo, um conhecimento dos métodos a serem empregados pelos engenheiros de software no processo de elicitação.

Conforme já mencionado em relação a coleta de fatos, caberá aos engenheiros de software a escolha da combinação dessas várias técnicas e suas estratégias com procedimentos adequados de forma a compor um método para comunicação. Dependendo do Udi, uma ou outra combinação será a mais eficaz.

O uso de ferramentas na comunicação é importante na medida em que pode tornar mais eficiente o emprego de determinado método. O uso de editores de texto com facilidades gráficas, o uso de hipertexto, uso de sistemas multi-mídia integrando áudio, vídeo e computador são exemplos de ferramentas que podem ajudar no processo de comunicação. Na seção seguinte tratamos de técnicas de validação no contexto da elicitação.

Validação

A validação de software, ou seja a conformação de que o produto é aquele desejado pelo usuário, ocorre, normalmente, no fim do ciclo de vida. O teste do sistema, como é comumente conhecida esta validação, é o teste integrado dos programas do sistema pelo usuário. No nosso caso a validação é feita no próprio processo de elicitação de requisitos, uma seja uma validação anterior a própria elicitação. Várias técnicas de validação de software tem sido propostas pela literatura. Segue alguns que algumas dessas técnicas também podem ser aplicadas a elicitação de requisitos, temos um conjunto do qual se sobressaem as seguintes técnicas:

- Comprovação Informal;
- Prototipagem;
- Uso de formalismos;
- Reusando Domínios;
- Resolução de Pontos de Vista.

Todas estas técnicas são capazes de achar alguma diferença (Delta (Deita)) entre os fatos coletados e o universo de informações. A identificação destes Delta, chamada de computação Delta, é feita durante o processo de comunicação. Portanto, no processo de comunicação entre atores do Udi e engenheiros de software há oportunidade para se encontrar diferenças entre os fatos coletados e o universo de informações. A coleta de fatos fornece as entradas para a computação Delta. As diferenças entre as técnicas de validação são determinadas pelo tipo e qualidade das entradas fornecidas para a computação Delta.

Na comprovação informal [Duid 86] a validação ou a revisão dos requisitos é basicamente uma tarefa de leitura de descrições em linguagem natural e do uso de "check-lists" para detectar problemas na expressão dos requisitos. As estratégias para a comprovação informal são várias, mas elas tem em comum a falta de um apoio automatizado e na excessiva dependência das habilidades analíticas de um "Don" engenheiro de software. A computação Delta é controlada por um lista de problemas/(informação, fatos)

Delta = lista de problemas/(informação, fatos)

Em prototipação vários tipos de protótipos tem sido propostos como um meio de obter uma retroalimentação do universo de informações. Alguns deles usam linguagens de alto nível (linguagens do tipo geradores de aplicação) [Horowitz 85], enquanto outros usam linguagens de especificação executáveis [Sissoft 82]. A idé² básica é que pela prototipação é possível validar os requisitos/especificação contra as expectativas do usuário. A computação Delta, neste caso, é controlada pelo comportamento dos fatos:

Delta = comportamento dos fatos/(expectativas do usuário, fatos)

No uso de formalismos [Jones 86], onde o engenheiro de software faz o papel de um provador de teoremas, a validação ocorre dada a possibilidade de se identificar inconsistências, as quais consequentemente controlam a computação Delta:

Delta = inconsistências/(fatos, informação)

Usando-se a técnica de reutilização de domínios [Reubstein 89], [Fickas 87], que é uma técnica na qual estratégias e heurísticas de Inteligência Artificial são usadas, pretendendo-se criticar os requisitos coletados contra um domínio previamente codificado. Portanto a formação desse domínio será baseado em fatos de sistemas similares que já tenham sido elicitados. O uso dessa técnica permitiria que ferramentas do tipo assistentes inteligentes fornecessem críticas sobre os requisitos preparados pelo engenheiro de software. Dada a disponibilidade de um domínio, é possível determinar "fatos errados" e "fatos faltantes". No caso do uso da técnica de reutilização de domínios, a computação Delta é controlada por fatos errados e fatos faltantes:

Delta = (fatos errados(fatos do domínio, fatos):fatos faltantes(fatos do domínio, fatos))

O uso de domínios para a validação possibilita que seja feita uma diferenciação entre problemas de correteza e completeteza, facilidade não presente nas atuais técnicas para validação. O problema com o uso de domínios é, não só o seu alto custo, como também a complexidade de sua construção [Arango 89].

A resolução de pontos de é uma técnica desenvolvida por nus [Leite 89a] [Leite 89b] com o objetivo de ser uma alternativa para a validação de pontos de vista é que esta oferece uma alternativa ao uso de domínios, já que pode diferenciar problemas de correteza e completeteza e não depende da construção de um domínio. Esta capacidade de diferenciar entre correteza e completeteza possibilita atacar um ponto crucial do desenvolvimento de software, isto é, será que estamos construindo o software desejado, ou não? Atacando o problema da validação logo no início do processo de construção estamos evitando ocorrer em possíveis erros de projeto, que mais tarde serão, não só de difícil correção, como de um alto custo. A técnica de resolução de pontos de vista é capaz de distinguir entre inconsistências, fatos errados e fatos faltantes.

Delta = (Inconsistencia(fatosa, fatosb):fatos errados(fatosa, fatosb):fatos faltantes(fatosa, fatosb))

De maneira semelhante aos processos de coleta de fatos e comunicação, a validação pode ser levada adiante usando-se uma combinação dessas técnicas, heurísticas a elas pertencentes e procedimentos. A combinação de um método vai depender do universo de informações. As ferramentas disponíveis dependem não só da técnica escolhida como também das estratégias escolhidas dentro de cada técnica. Na próxima seção concluiremos, resumindo o papel de cada s.t.c.-processo da elicitação e sua forma de interação.

Conclusão

Entendemos a tarefa de elicitação de requisitos como uma tarefa sob a responsabilidade do engenheiro de software, apresentamos uma caracterização e uma classificação dos importantes sub-processos envolvidos na elicitação. Nosso propósito é o de prover uma melhor estrutura para o entendimento do processo de elicitação, bem como para descrever algumas das técnicas que poderiam ser usadas.

Acreditamos que não é razoável a proposição de uma metodologia para a elicitação de requisitos, já que esta deve ser dependente do problema em questão. No entanto acreditamos, que nossa caracterização e classificação é um embasamento para a montagem de uma meta-metodologia, que servirá de base para a montagem de metodologias específicas segundo cada problema.

No que diz respeito a integração entre coleta de fatos, validação e comunicação, apresentamos na Figura 3, um diagrama SADT que trata da integração entre estas tarefas. Neste diagrama observamos o papel de cada processo. COLIAR usa as informações disponíveis para produzir os fatos segundo o método escolhido e as ferramentas de apoio, e é retroalimentado por novas informações resultante do processo de "COMUNICAR", no qual sob o controle de métodos e o uso de ferramentas, engenheiros de software com a participação opcional de atores do Udi descobrem a necessidade de analisar novas informações. O processo de "VALIDAR", também sob o controle de métodos e usando as ferramentas apropriadas, identifica problemas com os fatos. Observa-se que o processo de comunicação pode estar processando uma computação Delta, ou então processando uma das técnicas acima descritas com o objetivo de facilitar o entendimento pelos atores do Udi daquilo que o engenheiro de software entendeu.