Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

PROPOSTA DE UM PROCESSO PARA A AQUISIÇÃO DE CONHECIMENTO EXPLÍCITO

ANDREA PADOVAN JUBILEU
JAQUELINE BRIGLADORI PUGLIESI
SOLANGE OLIVEIRA REZENDE
ROSELY SANCHES

Nº 77

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Ago./1998

Proposta de um Processo para a Aquisição de Conhecimento Explícito [1]

Andrea Padovan Jubileu

Jaqueline Brigladori Pugliesi

Solange Oliveira Rezende

Rosely Sanches

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Caixa Postal 668, 13560-970 - São Carlos - SP
E-mail: {andrea, jbpuglie, solange, rsanches}@icmc.sc.usp.br

Versão 1.0

Julho de 1998

[1] Trabalho realizado com auxílio do CNPq e RECOPE-FINEP.

Sumário

1 Introdução **1**

2 Modelos de Ciclo de Vida para Desenvolvimento de Software **2**

2.1 Considerações Iniciais 2

2.2 Modelos de Ciclo de Vida para Desenvolvimento de Sistemas Tradicionais . 3

2.3 Modelos de Ciclo de Vida para Desenvolvimento de Sistema Baseado em Conhecimento 8

2.4 Principais Erros nas Etapas de Desenvolvimento de Sistemas Baseados em Conhecimento 13

2.5 Considerações Finais 14

3 Processo de Aquisição de Conhecimento **15**

3.1 Considerações Iniciais 15

3.2 A Aquisição de Conhecimento no Desenvolvimento de um Sistema Baseado em Conhecimento 15

3.3 Aquisição de Conhecimento Explícito 18

3.4 O Processo para Aquisição de Conhecimento Explícito - IPAIA 20

3.5 Considerações Finais 23

4 Conclusão **24**

Lista de Figuras

2.1 Modelo Cascata [Pressman 94]. 3

2.2 Modelo Espiral [Jones 90]. 4

2.3 Ciclo de Vida Evolucionário ou Incremental [Mayrhauser 90]. 5

2.4 Modelo de Prototipação Rápida [Jones 90]. 6

2.5 Ciclo de Desenvolvimento Eterno [Mayrhauser 90]. 6

2.6 Abordagem de Fase Embutida [Mayrhauser 90]. 7

2.7 Estrutura de um Sistema Baseado em Conhecimento. 8

2.8 Modelo Linear para Desenvolvimento de um Sistema Baseado em Conhecimento [Giarratano 94]. 10

2.9 Modelo Espiral para Desenvolvimento de um Sistema Baseado em Conhecimento [Giarratano 94]. 12

3.1 Formulário para a Administração da Aquisição de Conhecimento. 17

3.2 Tarefas e Fases da Aquisição de Conhecimento. 18

3.3 A Natureza Iterativa da Aquisição de Conhecimento [Tuthill 90]. 19

3.4 Proposta de um Processo para Aquisição de Conhecimento Explícito. . . . 21

..... BRANCO .....

Capítulo 1

Introdução

No desenvolvimento de software é importante que se tenha uma metodologia que auxilie a sua construção, de modo a se obter um sistema com qualidade. Para isso, existem diversos modelos de ciclo de vida de desenvolvimento de software, no qual cada um se adequa mais a um determinado tipo de software.

Os Sistemas Baseados em Conhecimento (SBCs) são um tipo específico de software. Para o desenvolvimento de SBCs existem propostas de modelos que foram elaboradas utilizando os modelos de ciclo de vida de desenvolvimento de software tradicionais como suporte. Os SBCs tem como ponto chave a aquisição e manipulação de conhecimento e portanto, a Aquisição de Conhecimento deve ser bem elaborada e realizada, para que se produza um sistema com qualidade e confiabilidade.

A Aquisição de Conhecimento é considerada o estágio do desenvolvimento de SBC mais complexo, dispendendo muito tempo e recursos. Para melhorar a viabilidade é apresentada neste trabalho uma proposta de um processo bem definido para Aquisição de Conhecimento Explícito considerando a natureza iterativa da obtenção do conhecimento, entre outras características.

Este relatório está organizado em quatro capítulos. No Capítulo 1 é apresentada uma introdução, considerando os aspectos que motivaram a elaboração da proposta do processo para Aquisição de Conhecimento Explícito. Alguns modelos de ciclo de vida para o desenvolvimento de software tradicional e para o desenvolvimento de Sistemas Baseados em Conhecimento são apresentados no Capítulo 2, além dos principais erros cometidos nas etapas de desenvolvimento de SBCs. O Capítulo 3 apresenta alguns aspectos da Aquisição de Conhecimento, e a proposta do processo IPAIA para a Aquisição de Conhecimento. No Capítulo 4 é apresentada a conclusão do trabalho, e por fim, as referências bibliográficas consultadas para a elaboração deste relatório.

Capítulo 2

Modelos de Ciclo de Vida para Desenvolvimento de Software

2.1 Considerações Iniciais

Um processo de software representa um conjunto sequencial de atividades, objetivos, transformações e eventos que encapsulam estratégias para o cumprimento da evolução do software [Kan 95].

Vários modelos de processo de desenvolvimento de software têm sido utilizados para sistemas tradicionais. Um tipo especial de software a ser considerado são os Sistemas Baseados em Conhecimento (SBC), que possuem natureza iterativa na obtenção do conhecimento, complexidade na validação e testes e complexidade na obtenção completa de seu comportamento no início do projeto. Devido essas características, os modelos existentes para o desenvolvimento de software tradicional não se adequam especificamente para o desenvolvimento de SBCs, mas foram utilizados como suporte para algumas propostas de modelos para o desenvolvimento de SBCs.

Neste capítulo são apresentados alguns modelos de ciclo de vida de desenvolvimento de software tradicionais, bem como alguns exemplos de modelos de ciclo de vida para desenvolvimento de SBC.

2.2 Modelos de Ciclo de Vida para Desenvolvimento de Sistemas Tradicionais

Para o desenvolvimento de software com qualidade é necessário a utilização de um modelo que direcione todas as atividades do processo de desenvolvimento desse software [Pressman 94].

- *Modelo Cascata*: este modelo é composto por seis fases sequenciais que são Engenharia de Sistemas, Análise de Requisitos, Projeto, Codificação, Teste e Manutenção (Figura 2.1). A etapa de Engenharia de Sistemas envolve a coleta dos requisitos em nível de sistema, com uma pequena parcela de projeto e análise de alto nível. Na Análise de Requisitos, o analista de software deve compreender o domínio da informação para o software, bem como a função, desempenho e interface exigidos. O Projeto de software é um processo que concentra-se na estrutura de dados, arquitetura do software, detalhes procedimentais e caracterização da interface. Durante a Codificação, o projeto deve ser traduzido para uma linguagem de programação. A etapa de Teste concentra-se nos aspectos lógicos internos do software, garantindo que todas as instruções tenham sido testadas e também concentra-se nos aspectos funcionais externos, realizando testes para descobrir erros e garantir que a entrada definida produza resultados reais condizentes aos resultados exigidos. Na Manutenção de software reaplica-se cada uma das etapas precedentes do ciclo de vida a um programa existente.

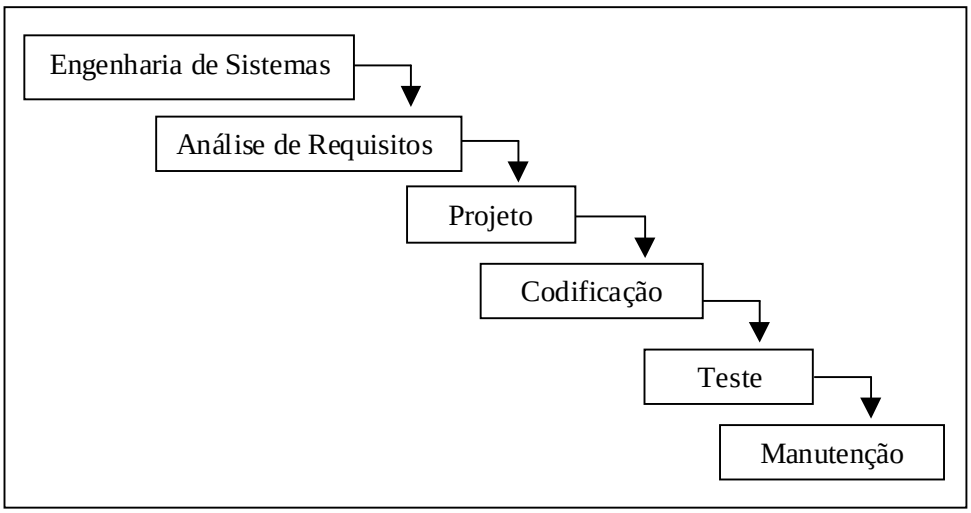


Figura 2.1: Modelo Cascata [Pressman 94].

- *Modelo Espiral*: neste modelo (1) o problema e as possíveis soluções são investigadas; (2) as soluções são avaliadas e os riscos identificados; (3) os riscos são resolvidos

através de experimentações com o protótipo; (4) o produto é desenvolvido e testado; (5) a próxima atividade é planejada, incluindo possíveis partições de esforços pessoais ou da função do software; e (6) os resultados são revisados e os planos para a próxima iteração são confirmados. O modelo espiral minimiza os riscos identificando incertezas no próximo estágio e usando protótipos para esclarecer a situação. Os estágios de desenvolvimento para um ciclo são reduzidos em quatro atividades como mostra a Figura 2.2. Assim, o desenvolvimento do software torna-se uma aplicação repetida das mesmas atividades para criar, incrementalmente, as versões preliminares do produto final.

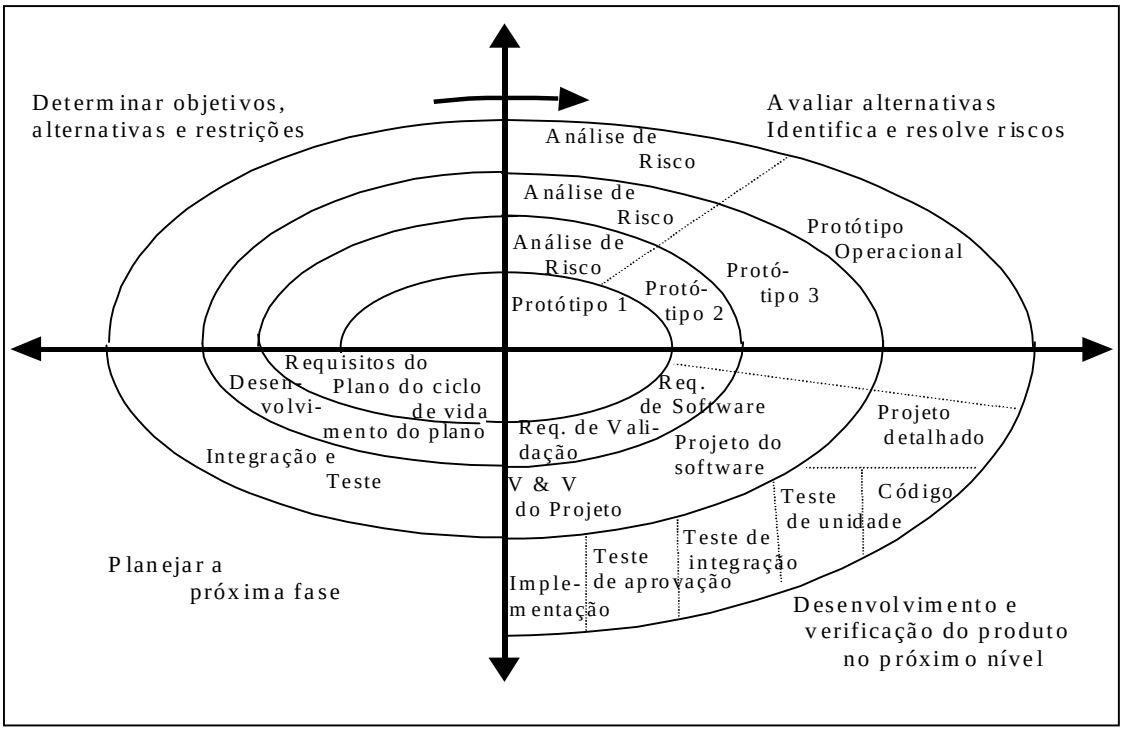


Figura 2.2: Modelo Espiral [Jones 90].

- *Ciclo de Vida Evolucionário ou Incremental*: os passos iterativos deste modelo abrangem Especificação, Projeto, Codificação, Teste e Monitoramento de Desempenho Contínuo. Esses passos diferenciam-se dos passos do modelo cascata pela existência de *feedback* e pela oportunidade de inserir alguma nova funcionalidade a cada iteração. O desenvolvimento evolucionário se orienta de acordo com os objetivos globais. A identificação de incrementos sucessivos requer um plano evolucionário preliminar e uma abordagem estrutural básica. Sintetizando, o Ciclo de Vida Evolucionário é um processo repetitivo que se inicia com um conjunto de soluções minimal e que vai implementando, sucessivamente, mais soluções. Assim, o software produto desenvolve-se através das versões incrementadas. A Figura 2.3 mostra o Ciclo de Vida Evolucionário.

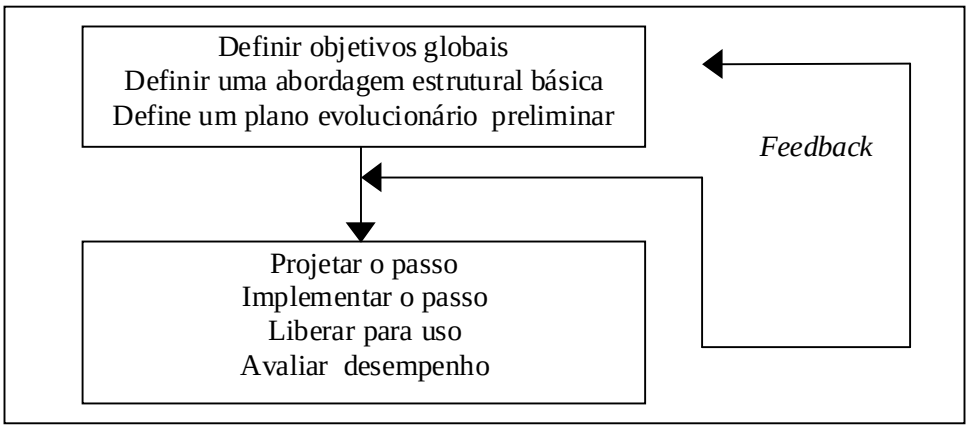


Figura 2.3: Ciclo de Vida Evolucionário ou Incremental [Mayrhauser 90].

- *Prototipação Rápida*: neste modelo, uma versão preliminar do software é desenvolvida reusando partes de outros produtos. Essa versão é avaliada pelos desenvolvedores e usuários. Se o produto for inviável ele é cancelado, senão os desenvolvedores obtêm, introspectivamente, a natureza do problema experimentando as soluções alternativas e investigando outras interfaces. Os usuários podem refinar seus requisitos e avaliar as soluções alternativas interagindo com os desenvolvedores. Cada versão posterior tem a probabilidade de ser mais eficiente e completa até ser obtido o produto final. Devido ao efeito do *feedback* em cada protótipo sucessivo, ao invés de linhas progressivas do modelo tradicional, tem-se algo parecido com o modelo espiral. Assim, os protótipos adicionais são refletidos após os estágios do ciclo de vida: Análise, Especificação, Projeto, Implementação e Teste, como mostra a Figura 2.4.

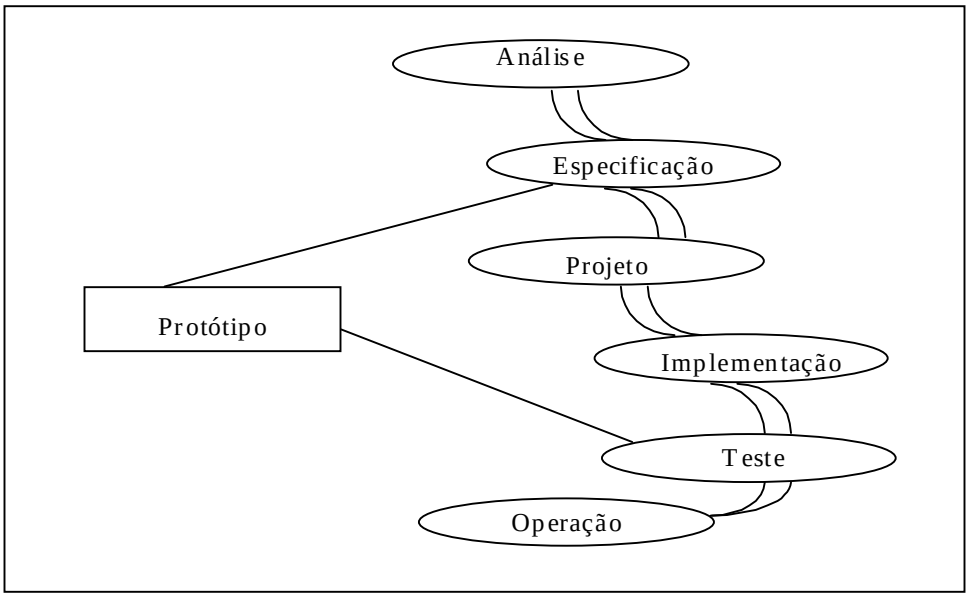


Figura 2.4: Modelo de Prototipação Rápida [Jones 90].

- *Ciclo de Desenvolvimento Eterno*: neste modelo, diferentemente do Ciclo de Vida Evolucionário, o *feedback* acontece no final do ciclo. Do princípio ao fim de todo o ciclo de vida deve-se avaliar a qualidade e analisar o que está certo ou errado. Baseado nisso, novos objetivos são definidos, e retorna-se à primeira fase do ciclo de vida. A Figura 2.5 mostra o Ciclo de Desenvolvimento Eterno.

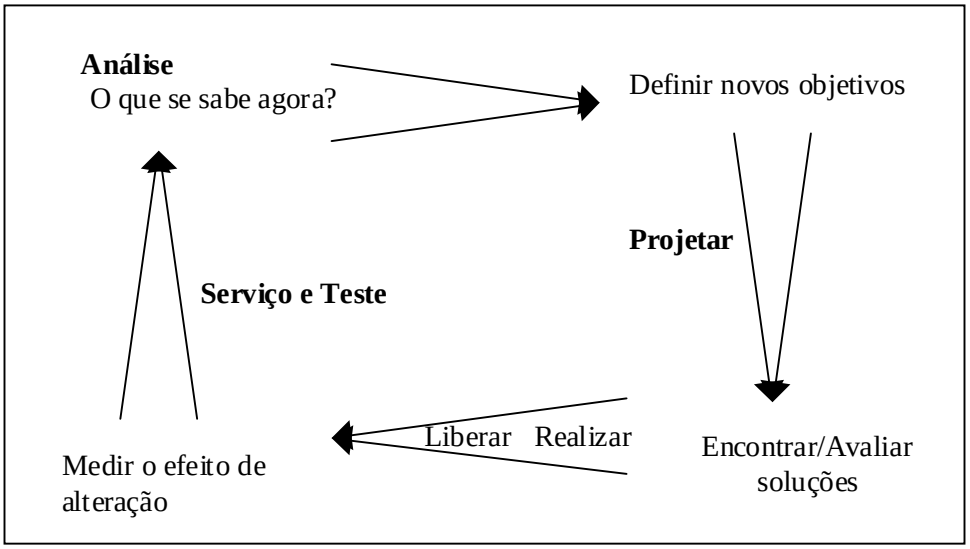


Figura 2.5: Ciclo de Desenvolvimento Eterno [Mayrhauser 90].

- *Abordagem de Fase Embutida*: este modelo combina a idéia de evolução (iteração) com os passos de desenvolvimento sequenciais, bem definidos, similares aos passos do Modelo Cascata. O que o torna diferente do modelo de Ciclo de Vida Evolucionário e do Ciclo de Desenvolvimento Eterno é a existência de um *feedback* mais freqüente embutido em todas as fases de desenvolvimento do Modelo Cascata. Este modelo é utilizado para projetos complexos a fim de forçar o particionamento do desenvolvimento em pequenos passos gerenciáveis.

  O projeto é dividido em pequenos pedaços similarmente às versões do Ciclo de Vida Evolucionário, como mostrado na Figura 2.3. O *loop* entre a manutenção e a redefinição do problema é fechado, similarmente ao Ciclo de Desenvolvimento Eterno. Há também um "pequeno ciclo de vida" inserido nesse *loop*. O conjunto de objetivos corrente direciona a definição do (sub)problema a ser resolvido no próximo ciclo de desenvolvimento. O *feedback* das versões existentes para as fases de especificações, de projeto, de código e de teste também é considerado quando o próximo ciclo de desenvolvimento é preparado. A Figura 2.6 indica esse *feedback* por linhas com setas em todas as fases.

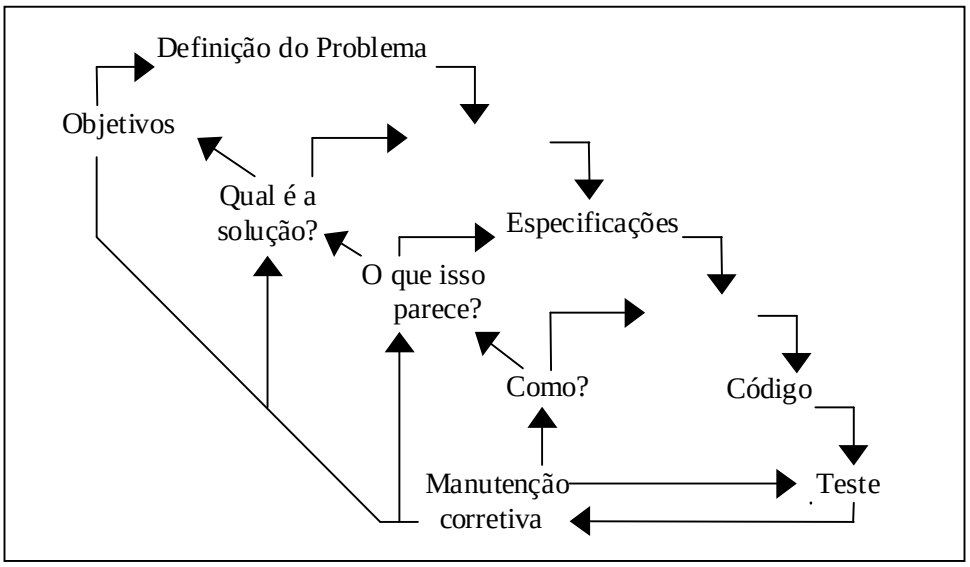


Figura 2.6: Abordagem de Fase Embutida [Mayrhauser 90].

2.3 Modelos de Ciclo de Vida para Desenvolvimento de Sistema Baseado em Conhecimento

Os Sistemas Baseados em Conhecimento (SBCs) são programas de computador que usam conhecimento representado explicitamente para resolver problemas. Eles manipulam conhecimento e informação de forma inteligente e são desenvolvidos para serem usados em problemas que requerem uma quantidade considerável de conhecimento humano e de "esperteza".

Para que um SBC chegue perto do desempenho de um especialista humano, o sistema deve ter grande quantidade de conhecimento especializado disponível, conseguir ter acesso rápido a esse conhecimento e ser capaz de raciocinar adequadamente com este conhecimento.

Um SBC possui dois componentes essenciais: uma base de conhecimento que contém o conhecimento específico do domínio e um motor de inferência consistindo de algoritmos para manipular o conhecimento representado na base de conhecimento.

A Figura 2.7 apresenta a estrutura básica geral de um SBC. O Núcleo do Sistema Baseado em Conhecimento (NSBC) desempenha as principais funções do SBC; a Base de Conhecimento (BC) contém todo o conhecimento do especialista do domínio selecionado. A forma de representação deve ser compatível com a forma de manipulação do NSBC ou SHELL, usando algum formalismo de Representação de Conhecimento; a Memória de Trabalho (MT) consiste na área de trabalho do sistema.

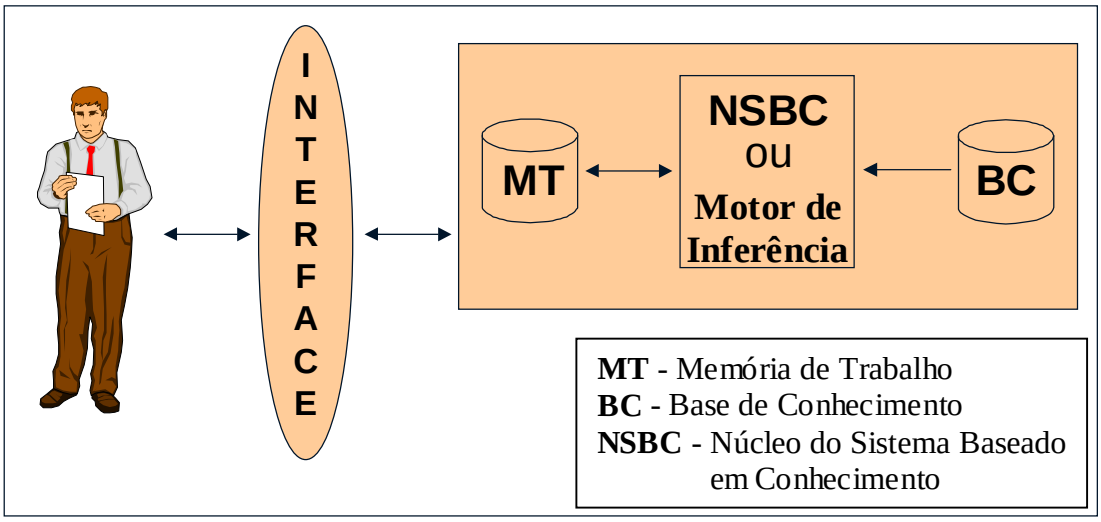


Figura 2.7: Estrutura de um Sistema Baseado em Conhecimento.

As principais etapas no desenvolvimento de um SBC são:

- **Especificação dos Requisitos**: envolve estudo de viabilidade, desenvolvimento de

metas, refinamento do domínio, escolha da equipe de Aquisição de Conhecimento e de projeto, e identificação de fontes de conhecimento;

- **Aquisição de Conhecimento**: refere-se ao processo de adquirir, organizar e estruturar o conhecimento, bem como a documentação do sistema a qual envolve, entre outros, a definição de um dicionário de conhecimento;

- **Projeto**: envolve a definição da estrutura e organização do conhecimento, a definição de métodos para processamento, a seleção de ferramentas adequadas e a documentação;

- **Implementação**: refere-se à codificação, documentação do sistema e geração de manuais;

- **Teste**: envolve a validação e verificação do sistema e é considerado um processo contínuo, pois há uma necessidade de assegurar que o sistema funcione corretamente, forneça resultados verdadeiros (corretos) e satisfaça os requisitos do cliente;

- **Manutenção**: realiza eventuais mudanças nos requisitos do sistema, enfatizando a Aquisição de Conhecimento contínua e a avaliação do sistema em andamento.

Várias pesquisas relatadas na literatura se referem ao desenvolvimento de Sistemas Baseados em Conhecimento [Giarratano 94, Rocha 97, Liebowitz 97]. O desenvolvimento desses sistemas normalmente usam métodos empíricos, e não tem suporte de metodologias que sejam amplamente aceitas. Um dos motivos é a falta de várias características desejáveis de um processo de Engenharia de Software, por exemplo, confiabilidade, estimativa de custos, garantia de qualidade, etc. [Liebowitz 89].

Algumas das propostas de modelos para desenvolvimento de SBCs incluem o modelo linear, espiral e incremental.

- *Modelo Linear*: este modelo foi utilizado em um número considerável de projetos de SBC (Figura 2.8), [Bochsler 88, Giarratano 94].

  Este modelo consiste de estágios que descrevem o desenvolvimento do sistema. Embora não explicitamente, a verificação e validação do sistema ocorrem em paralelo com os estágios. A mesma seqüência de passos se repete até que o sistema seja disponibilizado para uso. Os mesmos estágios são usados para manutenção e evoluções subseqüentes do sistema.

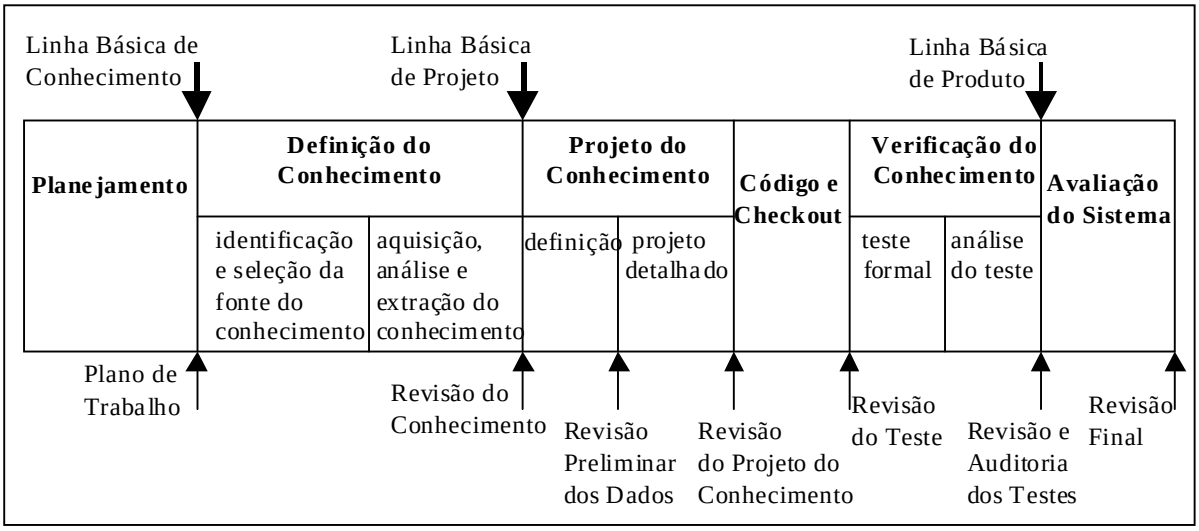


Figura 2.8: Modelo Linear para Desenvolvimento de um Sistema Baseado em Conhecimento [Giarratano 94].

- Planejamento

  O propósito deste estágio é especificar um plano formal de trabalho constituído de um conjunto de documentos que será utilizado para guiar e avaliar o desenvolvimento do SBC. O cálculo de possibilidades é uma das tarefas mais importantes nesse processo pois deve responder questões referentes à vantagem do projeto e ao paradigma apropriado para o sistema. As respostas para estas questões determinam a abordagem pela qual o projeto deve prosseguir, considerando fatores como: seleção de domínio apropriado, custos e viabilidade. Várias outras tarefas são realizadas neste estágio como: gerenciamento de recursos, divisão das tarefas, análise funcional preliminar e definição requisitos de alto nível.

- Definição do conhecimento

  O objetivo deste estágio é definir os requisitos de conhecimento do SBC, através dos seguintes sub-estágios com suas respectivas tarefas:

  1. identificação e seleção da fonte do conhecimento: identificação, importância, disponibilidade e seleção de uma ou mais fontes de conhecimento;
  2. a Aquisição de Conhecimento: estratégia de aquisição, identificação do elemento conhecimento, sistema de classificação do conhecimento, análise funcional detalhada, fluxo de controle preliminar, manual preliminar do usuário, especificação de requisitos, e linha básica do conhecimento.

- Projeto do conhecimento

  O objetivo deste estágio é produzir o projeto detalhado para o SBC e envolve as seguintes tarefas:

1. definição do conhecimento: representação, estrutura de controle detalhada, estrutura interna dos fatos, interface preliminar do usuário, plano inicial de teste;
2. projeto detalhado: estrutura, estratégia de implementação, interface do usuário, especificações do projeto, relatórios e plano de teste detalhado. O resultado deste estágio é a documentação produzida que deve ser revisada antes de iniciar o estágio de codificação.

– Codificação e *checkout*

O propósito deste estágio é a implementação e documentação do sistema, através das seguintes tarefas: representação do conhecimento, codificação, testes, listagem do código fonte, manual do usuário, guia de instalação e operação, e documento de descrição do sistema. Este estágio termina com uma revisão do sistema, que determina se o SBC está pronto para iniciar o próximo estágio.

– Verificação do conhecimento

Neste estágio o objetivo é determinar a corretitude, completitude e consistência do sistema através das tarefas:

1. teste formal: procedimentos de teste e relatórios de teste;
2. análise de testes: avaliação dos resultados e recomendações. Esta etapa procura olhar para os problemas maiores, como: respostas incorretas, incompletas e inconsistentes. Isso determina se o problema está concentrado no conhecimento, cadeias de inferência, incerteza ou alguma combinação desses três fatores. Se o problema não pode ser detectado, o sistema deve ser analisado com alguma ferramenta que possibilite encontrar e identificar as falhas.

– Avaliação do sistema

Este é o estágio final no processo de desenvolvimento tendo como propósito avaliar o sistema considerando as recomendações para melhorias e correções através das seguintes tarefas: avaliação dos resultados, recomendações, validação, produção do relatório final/provisório.

Considerando que o SBC seja construído por um processo contínuo, o relatório do estágio de avaliação do sistema normalmente é provisório descrevendo o acréscimo de funcionalidade, quando novos conhecimentos são adicionados. A verificação do sistema deve ser executada em conjunto com todo o conhecimento e não apenas o novo conhecimento. O SBC deve também ser validado neste estágio antes de esperar pela iteração final. Sistemas automatizados para validação de bases de conhecimento podem ser úteis neste estágio.

- *Modelo Espiral*: este modelo é representado por uma espiral que pode ser refinada em quatro quadrantes, que são Planejamento, Aquisição de Conhecimento, Codificação e Avaliação do SBC.

Cada iteração ao redor da espiral (do centro para fora) adiciona alguma nova capacidade funcional ao sistema. O ponto final da espiral não determina de fato o término do sistema, ao invés disso, uma nova espiral é iniciada com a manutenção e a evolução do sistema. Na Figura 2.9 é mostrada uma adaptação do modelo espiral proposta por Giarratano para desenvolvimento de SBC [Giarratano 94].

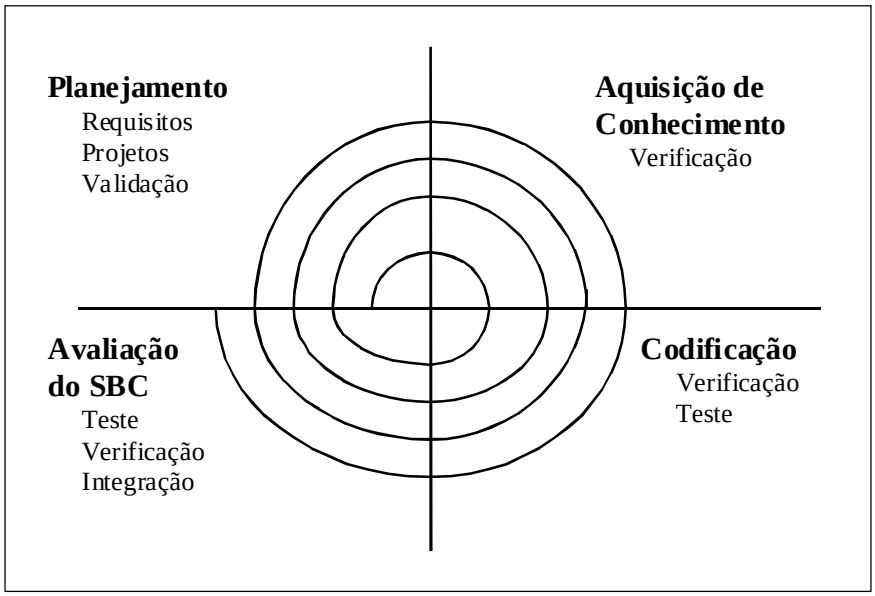


Figura 2.9: Modelo Espiral para Desenvolvimento de um Sistema Baseado em Conhecimento [Giarratano 94].

- *Modelo Incremental*: este modelo é um refinamento do modelo cascata e da abordagem *top-down* padrão. A idéia básica do desenvolvimento incremental é desenvolver software baseado nos incrementos da capacidade funcional. O Modelo Incremental tem sido utilizado com muito sucesso em grandes projetos de software convencional. O Modelo Incremental é também útil para o desenvolvimento de Sistemas Baseados em Conhecimento, no qual a adição de regras aumentam as capacidades do sistema de nível básico para nível especialista. Os menores incrementos correspondem aos incrementos de conhecimento que ofereça algum aumento significativo. O micro incremento é a mudança no conhecimento pela adição ou refinamento de regras [Giarratano 94].

  A vantagem principal do Modelo Incremental é que os aumentos na capacidade funcional são mais fáceis de testar, verificar e validar do que os produtos dos estágios individuais do Modelo Cascata. Cada incremento funcional pode ser testado, verificado e validado imediatamente com o especialista, ao invés de tentar fazer toda a validação no final. Isso diminui o custo de incorporar correções no sistema. Em essência, o Modelo Incremental é similar a uma prototipação rápida contínua que se

estende sobre todo o desenvolvimento. Ao invés de apenas um protótipo rápido dos estágios iniciais para determinar os requisitos na abordagem "faça isso duas vezes", o protótipo desenvolvido é o próprio sistema.

2.4 Principais Erros nas Etapas de Desenvolvimento de Sistemas Baseados em Conhecimento

Os potenciais erros de desenvolvimento de SBC podem ser classificados pelas etapas mais prováveis nas quais eles ocorrem [Giarratano 94]. Esses erros incluem:

- **erros de conhecimento do especialista:** quando o especialista é a fonte de conhecimento do SBC, se seu conhecimento é errôneo, os resultados podem ser propagados através de todo o processo de desenvolvimento. Um ponto positivo de se construir um SBC é a potencial detecção de conhecimento errôneo quando o conhecimento do especialista é feito explícito;

- **erros semânticos:** um erro de semântica ocorre quando o significado do conhecimento não é propriamente comunicado. Como por exemplo, suponha que um especialista diga "Você pode extinguir um fogo com água" e o engenheiro de conhecimento interprete isso como "Todo fogo pode ser extinguido com água". Erros semânticos ocorrem quando o engenheiro de conhecimento interpreta mal as respostas do especialista, ou o especialista interpreta mal a pergunta do engenheiro de conhecimento, ou ambos;

- **erros sintáticos:** esses são erros simples que ocorrem quando é entrada a forma incorreta de uma regra ou fato. Caso se utilize uma ferramenta de SBC, ela deve comunicar esses erros e uma mensagem apropriada deve ser mostrada. Outros erros que ocorrem na fase de construção da Base de Conhecimento são devido a erros de origem do conhecimento que não foram detectados em etapas anteriores;

- **erros do motor de inferência:** assim como qualquer espécie de software, o motor de inferência pode possuir erros. Na medida que uma ferramenta de SBC é licenciada para uso geral, todos os erros comuns devem ser revisados e solucionados. Entretanto, podem existir erros que apenas aparecem em circunstâncias raras, tal como ter 159 regras na memória de trabalho. Alguns erros podem ser bastante sutis e aparecer apenas em certos padrões de operações;

- **erros da cadeia de inferência:** esses erros podem ser causados por conhecimento errôneo, erros semânticos, erros no motor de inferência, especificação incorreta das prioridades das regras, e interações inesperadas entre regras. Erros de cadeia de inferência mais complexos podem ser devidos à incerteza de regras e evidências, à propagação de incerteza na cadeia de inferência.

A escolha de um método para lidar com incerteza não resolve automaticamente todas as questões relacionadas à incerteza. Por exemplo, antes de escolher inferência Bayesiana simples, deve-se checar se a premissa de independência condicional é garantida;

- **erros dos limites ignorantes:** um problema que é comum a todos os estágios de desenvolvimento é a especificação dos limites de ignorância do sistema. Os especialistas humanos conhecem a extensão de seu conhecimento e seu desempenho diminui gradativamente às barreiras de suas ignorâncias. Especialistas humanos devem ser honestos o suficiente para admitir mais incerteza em suas conclusões próximas à essas barreiras. Entretanto, a não ser que um SBC seja especificamente programado para admitir incerteza, ele pode continuar a suprir respostas mesmo que a cadeia de inferência e evidência seja muito fraca.

2.5 Considerações Finais

Neste capítulo foram citados alguns dos principais modelos de desenvolvimento de software tradicional. Alguns desses modelos, com modificações, são adequados para o desenvolvimento de Sistemas Baseados em Conhecimento, como o Modelo Linear, Espiral e Incremental.

A Aquisição de Conhecimento é um dos estágios mais difíceis e determinantes no desenvolvimento de um Sistema Baseado em Conhecimento com qualidade. Dessa forma, é necessário uma atenção especial nessa etapa, sendo importante uma formalização e detalhamento de um processo para auxiliar a etapa de Aquisição de Conhecimento.

Considerando a maneira como os modelos mostrados estão definidos, nenhum deles se adequam para a execução do estágio de Aquisição de Conhecimento. Assim, será proposto, no próximo capítulo, um modelo de processo para Aquisição de Conhecimento considerando suas características particulares e suas principais tarefas.

Capítulo 3

Processo de Aquisição de Conhecimento

3.1 Considerações Iniciais

A expressão Aquisição de Conhecimento (AC) corresponde a todos os mecanismos para adquirir conhecimento a ser utilizado na modelagem de um domínio de conhecimento. Diversas fontes – especialistas humanos, livros, exemplos, etc. – podem ser utilizadas na AC. Pode-se dizer também que a AC é o processo que visa obter toda a informação necessária à construção de uma Base de Conhecimento (BC) para Sistemas Baseados em Conehcimento. Já a expressão Elicitação do Conhecimento designa a tarefa de apreensão do conhecimento de algum especialista através de sessões de AC. Alternativamente, pode ser chamada de "aquisição de conhecimento explícito ou manual" ou "eliciação de conhecimento".

Nas seções a seguir será apresentado o estágio de AC de um SBC mais detalhadamente e proposto um processo para Aquisição de Conhecimento Explícito.

3.2 A Aquisição de Conhecimento no Desenvolvimento de um Sistema Baseado em Conhecimento

A Aquisição de Conhecimento é o estágio mais crítico no desenvolvimento de SBCs, consumindo muito tempo e pessoas. É uma área que inclui principalmente Psicologia Cognitiva e Inteligência Artificial. As metodologias de desenvolvimento atuais enfatizam a abordagem de prototipação rápida que provê algumas diretrizes e procedimentos para ajudar a equipe de AC a organizar o processo de Aquisição de Conhecimento. Como um processo, a AC envolve a extração, interpretação e representação do conhecimento de um dado domínio.

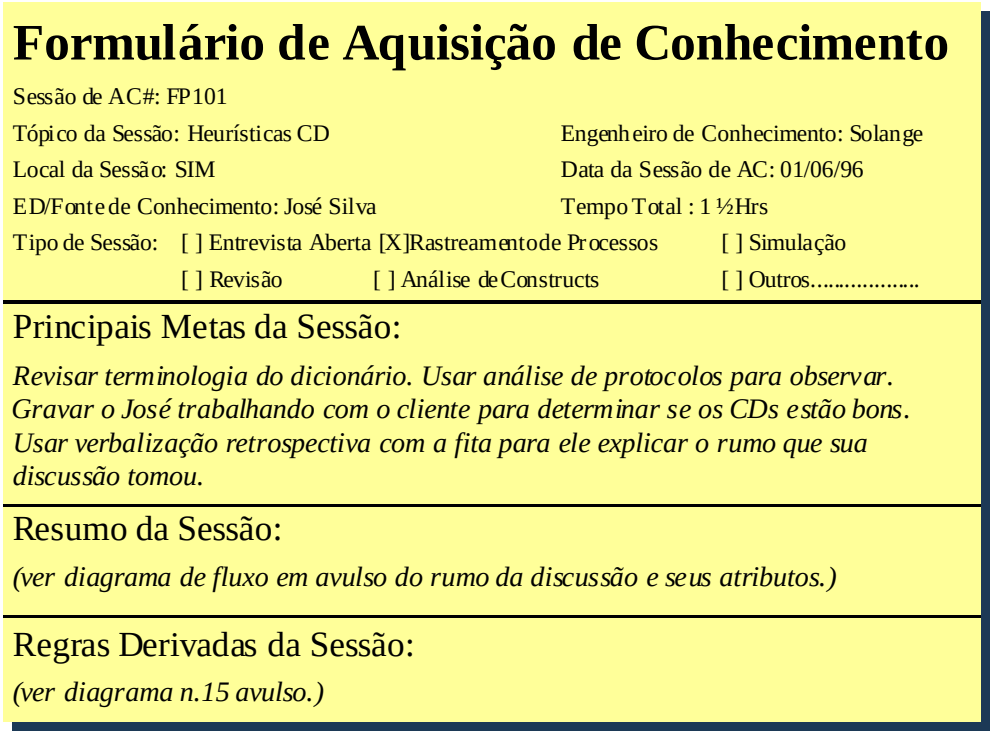
Adquirir conhecimento especialista envolve obter informações dos especialistas e/ou de fontes documentadas, classificação dessa informação em declarativa (factual) ou procedural, codificação dessa informação num formato utilizado pelo sistema, e checagem de consistência do conhecimento codificado com o conhecimento existente no sistema.

Existem técnicas para apoiar o estágio de AC. Essas técnicas podem ser divididas em dois grandes grupos. O primeiro deles, foco deste relatório, é a AC explícito ou manual, onde é necessária a presença de um ou mais especialistas do domínio, a partir dos quais a equipe de AC fará toda a extração do conhecimento necessário para o desenvolvimento do sistema. Entre as técnicas de Aquisição de Conhecimento Explícito pode-se destacar a Entrevista (estruturada e desestruturada), Análise de Protocolos, *Brainstorming*, Estudo de Casos, Rastreamento de Processos, Repertório *Grid*, Estudo de Casos, Introspecção, etc. Maiores detalhes sobre essas e outras técnicas podem ser obtidos em [Rezende 98]. O segundo tipo é a AC implícito ou automática, que visa extrair conhecimento de dados e/ou informações referentes ao domínio, através de sistemas de Aprendizado de Máquina, *Data Mining* e Extração de Conhecimento de Base de Dados.

Vários fatores influenciam a escolha de uma técnica de AC, que depende, entre outros, da escala do problema e do tipo de metas. Por exemplo, as entrevistas desestruturadas e análise de conceitos são mais úteis nos estágios iniciais, enquanto entrevistas estruturadas, rastreamento do problema e simulações tendem a ser mais úteis depois que o domínio estiver claramente definido. Outro ponto que vale destacar na determinação das técnicas a serem utilizadas refere-se às fontes de conhecimento, levando em consideração a disponibilidade de especialistas, o número e experiência desses especialistas, bem como a disponibilidade de exemplos e Bases de Dados.

Por tratar-se de um estágio bastante complexo, é viável e de grande importância a administração do processo de Aquisição de Conhecimento. Um ponto chave nessa administração é possuir um formulário que documente os planos ou os resultados das sessões de AC [McGraw 89]. Inicialmente, esse formulário pode ser usado para planejar a sessão de AC e alertar o especialista do domínio para os tópicos que serão discutidos. Posteriormente, ele pode ser expandido para documentar as principais conclusões da sessão, planos futuros, e/ou uma representação intermediária de possíveis regras de conhecimento.

Os formulários para documentar a AC devem possuir no mínimo o número, a data e a hora em que foi realizada a sessão; o nome do engenheiro de conhecimento, do especialista ou de materiais consultados; fatos, regras ou conceitos principais que foram derivados; e pontos levantados e áreas que requerem maior refinamento em sessões futuras. A Figura 3.1 apresenta um exemplo de um formulário para a administração da AC.



Formulário de Aquisição de Conhecimento

Sessão de AC#: FP101

Tópico da Sessão: Heurísticas CD — Engenheiro de Conhecimento: Solange

Local da Sessão: SIM — Data da Sessão de AC: 01/06/96

ED/Fonte de Conhecimento: José Silva — Tempo Total : 1 ½Hrs

Tipo de Sessão: [ ] Entrevista Aberta [X]Rastreamento de Processos [ ] Simulação

[ ] Revisão [ ] Análise de Constructs [ ] Outros.....

Principais Metas da Sessão:

Revisar terminologia do dicionário. Usar análise de protocolos para observar. Gravar o José trabalhando com o cliente para determinar se os CDs estão bons. Usar verbalização retrospectiva com a fita para ele explicar o rumo que sua discussão tomou.

Resumo da Sessão:

(ver diagrama de fluxo em avulso do rumo da discussão e seus atributos.)

Regras Derivadas da Sessão:

(ver diagrama n.15 avulso.)

Figura 3.1: Formulário para a Administração da Aquisição de Conhecimento.

Outro ponto que merece ser considerado refere-se aos problemas que normalmente ocorrem durante a Aquisição de Conhecimento. Independentemente das habilidades da equipe de AC, a natureza interpessoal da sessão de AC, juntamente à dificuldade da tarefa, assegura que problemas devam surgir [McGraw 89]. As seguintes dificuldades serão, muito provavelmente, evidentes por elas mesmas:

- negativismo e apatia;
- falta de compromisso;
- blocos de comunicações verbais e não verbais;
- hostilidade e reações defensivas;
- conflitos entre expectativas e realidades;
- conhecimento irrelevante, incorreto, incompleto ou inconsistente;
- incapacidade do especialista de verbalizar o seu conhecimento e/ou raciocínio;
- especialista não é capaz de especificar seu raciocínio;
- discordância entre múltiplos especialistas.

Uma maneira de formalizar a Aquisição de Conhecimento é através de um processo que organize as diretrizes para tal.

3.3 Aquisição de Conhecimento Explícito

Alguns dos modelos existentes para direcionar a Aquisição de Conhecimento Explícito são:

- Modelo de McGraw

  Para cada ciclo de desenvolvimento, existem algumas fases discerníveis. Na maioria dos casos, essas fases seguem o nível de especialidade da informação desejada [McGraw 89]. Segundo McGraw, a Aquisição de Conhecimento pode ser dividida em três grandes fases. A Figura 3.2 mostra uma visão geral dessas fases, quem são os maiores responsáveis por cada fase, tarefas específicas a serem conduzidas durante cada fase e estimativa da porcentagem total de esforço humano necessário. Essa porcentagem representa a divisão de tempo exigida para cada grupo (equipe de Aquisição de Conhecimento e especialista do domínio) completar suas tarefas.

Fase 1 80% Equipe de AC 20% Especialista do Domínio	Definir o Problema Desenvolver um Mapa Mental Fazer uma Análise Funcional Fazer uma Análise de Tarefas
Fase 2 50% Equipe de AC 50% Especialista do Domínio	Realizar Sessões Iniciais de Aquisição de Conhecimento *Feedback* e Refinamento Realizar Sessões Aprofundadas de Aquisição de Conhecimento
Fase 3 80% Equipe de AC 20% Especialista do Domínio	Implementar, testar e refinar a Base de Conhecimento

Figura 3.2: Tarefas e Fases da Aquisição de Conhecimento.

Os esforços de AC na Fase 1, em um dado ciclo de desenvolvimento, são mais gerais, e quando finalizados, proporcionam um alicerce para o conhecimento que será extraído. Tarefas importantes a serem completadas durante essa fase incluem desenvolver um dicionário de conhecimento do vocabulário e conceitos do domínio, e identificar relações entre esses conceitos, bem como suas principais funções, entradas e saídas atuais. Nessa fase, técnicas como entrevistas desestruturadas podem ser apropriadas, sendo que o maior esforço será por parte da equipe de AC. À medida

que as informações se tornam mais específicas, o desenvolvimento passa para a Fase 2, e técnicas como entrevista estruturada e análise de protocolos se tornam mais apropriadas. Nessa fase, ocorre a elaboração de exemplos e grafos de conhecimento, o desenvolvimento de um pseudocódigo e a revisão de notas de sessões anteriores. A Fase 3 inclui a representação do conhecimento em pseudocódigo, seu refinamento, bem como teste de cenários ou testes através de painéis.

As fases deste modelo apresentado são de natureza linear, e estão sendo substituídas por modelos com ciclos entre seus estágios (protótipos são projetados, construídos e testados e então versões revisadas são novamente projetadas, construídas e testadas).

- Modelo de Martin e Oxman

  Martin e Oxman (1988) consideram que o processo iterativo depende do tamanho do sistema a ser construído, profundidade e amplitude da tarefa a ser realizada, e a qualidade do conhecimento adquirido [Tuthill 90] (Figura 3.3).

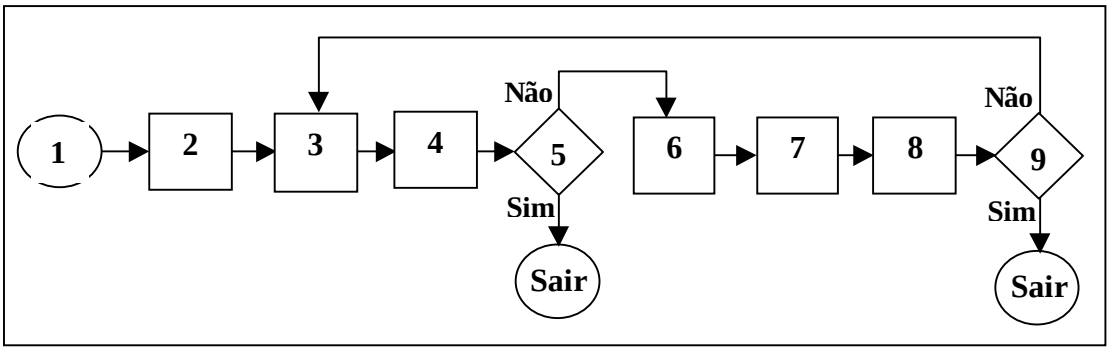


Figura 3.3: A Natureza Iterativa da Aquisição de Conhecimento [Tuthill 90].

Os passos que compõem o processo iterativo de Martin e Oxman são os seguintes:

1. adquirir conhecimento inicial do domínio;
2. prototipar o conhecimento e colocá-lo em um SHELL para testes;
3. dar ao protótipo um exemplo de tarefa;
4. deixar o especialista observar o sistema;
5. se a Base de Conhecimento está razoavelmente completa, sair;
6. deixar o especialista inferir o que é necessário para a Base de Conhecimento;
7. adquirir conhecimento necessário;
8. adicionar o conhecimento necessário a Base de Conhecimento;
9. retornar ao passo 3.

Vale ressaltar que esses modelos não satisfazem por completo as necessidades de organização de um processo de Aquisição de Conhecimento Explícito.

Propõe-se a seguir, um processo de AC baseado na abordagem IDEAL desenvolvida para melhoria de qualidade do processo de software [Paulk 95].

3.4 O Processo para Aquisição de Conhecimento Explícito - IPAIA

O processo de Aquisição de Conhecimento descrito nesta seção denomina-se IPAIA (acrônimo para as iniciais das fases propostas) e apresenta orientações que proporcionam uma abordagem de Aquisição de Conhecimento Explícito de uma maneira organizada. Esse processo pode ser direcionado tanto para grandes quanto para pequenos programas de AC e pode ser desenvolvido em cinco fases: (1) **I**nício do Processo de Aquisição de Conhecimento, (2) **P**reparação da Sessão de Aquisição de Conhecimento, (3) **A**quisição do Conhecimento, (4) **I**mplementação do Conhecimento e (5) **A**valiação do Conhecimento. A fase 1 é realizada apenas uma vez, e as fases 2, 3, 4 e 5 compõem uma etapa contínua de melhoramento do conhecimento, como mostrado na Figura 3.4.

Fase 1 - Início do Processo de Aquisição de Conhecimento: esta fase descreve o domínio de conhecimento, termos chaves e referências. Também identifica um resumo simplificado dos conceitos relacionados ao domínio de conhecimento, para que as pessoas que tenham que interagir com o processo de Aquisição de Conhecimento possam compreendê-lo melhor. É desenvolvido um mapa mental do domínio de conhecimento, realizada a análise funcional na qual módulos distintos e entradas e saídas atuais são identificados, selecionada a equipe de AC e escolhida a estrutura de pré-representação do conhecimento dentro do domínio específico.

- *Selecionar o Domínio de Conhecimento*: a seleção de um domínio apropriado é crucial para o êxito no desenvolvimento de um SBC. Os limites da tarefa devem ser claramente delineados;
- *Analisar o Problema*: os termos chaves são importantes para se obter uma terminologia única entre os membros da equipe de Aquisição de Conhecimento; devem ser identificadas todas as referências que o processo de Aquisição de Conhecimento fizer a padrões, procedimentos, terminologias ou outros documentos; os conceitos iniciais referentes ao domínio de conhecimento devem ser especificados; um mapa mental do domínio de conhecimento deve identificar relações entre os conceitos e identificar as principais funções permitindo, assim, uma maior familiarização com o domínio do conhecimento; uma análise funcional pode ser realizada com o propósito de refletir a principal funcionalidade que o usuário deseja do sistema. A análise funcional é uma

técnica que o engenheiro de conhecimento pode usar para estruturar os requisitos do sistema e delinear a sua funcionalidade. Na análise funcional devem ser identificados módulos distintos, entradas e saídas atuais. A ênfase nesta tarefa é traduzir os requisitos para um diagrama funcional inicial e descrever o relacionamento básico entre as funções. Usando diagramas de fluxo funcional, o engenheiro de conhecimento pode descrever sistemas maiores, subsistemas, requisitos de sistemas e interações entre os componentes do sistema;

- *Selecionar a Equipe de Aquisição de Conhecimento*: uma característica fundamental para a formação da equipe de AC é uma boa interação, que é indispensável entre os componentes de qualquer grupo de trabalho, principalmente na situação aqui tratada, que envolve um processo dependente da motivação de todos os participantes. Uma equipe de AC normalmente é composta pelos seguintes membros: Gerente, Engenheiro de Conhecimento, Anotador/Assistente, Observador e Psicólogo;

- *Selecionar a Estrutura de Pré-Representação do Conhecimento*: a estrutura interna que representará o conhecimento deve ser selecionada de acordo com a necessidade da aplicação e/ou das definições do projeto do SBC. Dentre estas estruturas estão regras, *scripts* e *frames*.

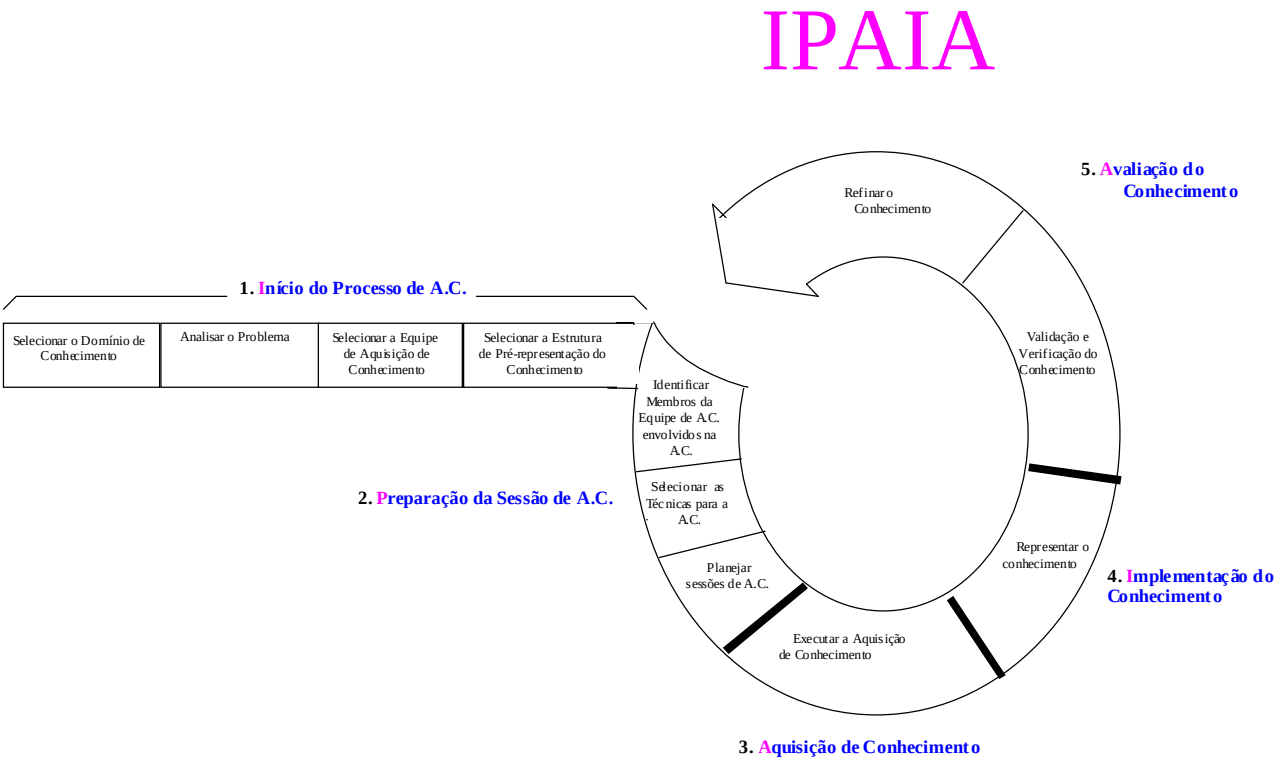


Figura 3.4: Proposta de um Processo para Aquisição de Conhecimento Explícito.

Fase 2 - Preparação da Sessão de Aquisição de Conhecimento: nesta fase são planejadas as sessões de Aquisição de Conhecimento, selecionadas as técnicas mais adequadas à aplicação e identificados os membros da equipe de AC envolvidos nas sessões.

- *Selecionar as Técnicas para a Aquisição de Conhecimento*: muitas variáveis influenciam na seleção de técnicas de Aquisição de Conhecimento e o grau de estruturação com o qual elas são aplicadas. Entre essas variáveis estão: Tamanho e Meta do Programa de Aquisição de Conhecimento, Disponibilidade de Especialista(s) do Domínio, Quantidade e Experiência da Equipe de AC, e Tipo de Conhecimento.

  - Tamanho e Meta do Programa de Aquisição de Conhecimento: programas de grande escala têm maior necessidade de estruturas e técnicas pré-determinadas do que programas menores. Por exemplo, numa fase inicial de Aquisição de Conhecimento pode-se utilizar as técnicas de Entrevista e Análise de Protocolos. Já nas sessões de Feedback e Refinamento, pode ser utilizado o Rastreamento de Processo seguido pela Entrevista Estruturada. Nas sessões aprofundadas de Aquisição de Conhecimento podem ser utilizadas Entrevista Estruturada, Simulações e Protótipos de desenvolvimento. A meta do programa de Aquisição de Conhecimento e a funcionalidade do Sistema Baseado em Conhecimento têm impacto sobre o tipo de técnicas de Aquisição de Conhecimento que serão selecionadas;
  - Disponibilidade de Especialista(s) do Domínio: as técnicas também são selecionadas de acordo com o tempo que um especialista dispõe para um programa de AC. Isso não desconsidera o fato de que em alguns casos, a técnica mais apropriada requerirá um tempo maior que aquele que o especialista tem disponível. Nesses casos deve haver uma negociação. Deve-se considerar, também, o fato de que o tempo disponibilizado a um programa de AC depende da complexidade do domínio do conhecimento em questão;
  - Quantidade e Experiência da Equipe de Aquisição de Conhecimento: além da quantidade de membros da equipe de AC disponíveis e da experiência dos mesmos com Aquisição de Conhecimento, um outro aspecto considerado no esforço dessa aquisição é o grau de compatibilidade entre as técnicas de Aquisição de Conhecimento e a experiência ou capacidade atual do engenheiro de conhecimento com as mesmas;
  - Tipo de Conhecimento: é importante caracterizar e combinar os tipos de conhecimento com as técnicas de Aquisição de Conhecimento apropriadas. Os tipos de conhecimento são declarativo, procedural, semântico e episódico. No conhecimento declarativo, identificam-se heurísticas gerais que são avaliáveis num nível consciente ("sabendo que"), geralmente em memória de curto prazo, e são expressas de forma verbal. No conhecimento procedural, identificam-se tarefas/procedimentos de rotina. Este conhecimento inclui a habilidade de um indivíduo de "como" realizar algo, envolve uma resposta automática para

um estímulo e pode ser de natureza reacionária. No conhecimento semântico, identificam-se principais conceitos, ou seja, o vocabulário, procedimentos de decisão e heurísticas avaliáveis num nível inconsciente. No conhecimento episódico, identificam-se heurísticas de resolução de problemas análogos. Este conhecimento reside em memória de longo prazo e contém informações sobre "episódios ou eventos temporariamente datados e relações temporais entre estes eventos";

- *Identificar Membros da Equipe de A.C. Envolvidos na Aquisição de Conhecimento*: inclui indicar quem deve acompanhar as atividades. Cada uma das tarefas de Aquisição de Conhecimento deve ter um membro responsável. É recomendável que se registrem os atributos necessários e desejáveis para quem ocupa cada cargo.
- *Planejar Sessões de Aquisição de Conhecimento*: esta tarefa tem por finalidade criar, efetivamente, um instrumento de auxílio à Aquisição de Conhecimento e planejar como as sessões de Aquisição de Conhecimento serão conduzidas;

Fase 3 - Aquisição do Conhecimento: esta fase tem como objetivo adquirir os conhecimentos que serão armazenados na Base de Conhecimento, ou seja, é a fase de execução do planejamento realizado na etapa anterior.

Fase 4 - Implementação do Conhecimento: nesta fase, o conhecimento adquirido deve ser representado formalmente. Para isso utiliza-se a estrutura de representação do conhecimento selecionada na Fase 1 deste Processo de Aquisição de Conhecimento.

Fase 5 - Avaliação do Conhecimento: nesta fase é realizada a verificação e validação do conhecimento que será refinado posteriormente, averiguando se o conhecimento existente é suficiente.

3.5 Considerações Finais

Neste capítulo foi apresentado o problema da Aquisição de Conhecimento durante o desenvolvimento de um Sistema Baseado em Conhecimento, bem como alguns processos de AC existentes na literatura.

Destaca-se a proposta de um processo de Aquisição de Conhecimento Explícito denominado IPAIA, que levou em consideração as características peculiares de um SBC, além das principais tarefas envolvidas na AC.

Capítulo 4

Conclusão

O foco deste trabalho foi a elaboração de uma proposta de um processo para Aquisição de Conhecimento Explícito a fim de auxiliar engenheiros de conhecimento na preparação e execução de uma AC estruturada e eficiente.

Essa necessidade originou-se devido aos vários problemas existentes durante a AC, e por ser essa uma etapa difícil e de extrema importância para o desenvolvimento de um Sistema Baseado em Conhecimento confiável e de qualidade.

Para instanciar o trabalho dentro de um contexto, foram estudados alguns modelos de ciclos de desenvolvimento para software, tanto tradicionais quanto para SBCs.

Referências Bibliográficas

[Bochsler 88] Bochsler, D. C.. *A Project Management Approach to Expert System Applications.* ISA 88, 1988.

[Giarratano 94] Giarratano, J. C.; Riley, G. *Expert Systems: Principles and Programming.* PWS Publishing Company, 2nd edition, 1994.

[Jones 90] Jones, G. W. *Software Engineering.* John Wiley & Sons Inc., Canadá, 1990.

[Kan 95] Kan, S. H. *Metrics and Models in Software Quality Engineering.* Addison-Wesley, EUA, 1995.

[Liebowitz 89] Liebowitz, J.; De Salvo, D. A. *Struturing Expert Systems: Domain, Design and Development.* Yourdon Press, 1989.

[Liebowitz 97] Liebowitz, J.; Wilcox, L. C. *Knowledge Management and Its Integrative Elements.* CRC Press, 1997.

[Mayrhauser 90] Mayrhauser, A. *Software Engineering: Methods and Management.* Academic Press Inc., 1990.

[McGraw 89] McGraw, K. L.; Briggs, K. H. *Knowledge Acquisition: Principles and Guidelines.* Prentice Hall, 1989.

[Paulk 95] Paulk, M. C.; Weber, C. V.; Curtis, B.; Chrissis, M. B.; et al. *The Capability Maturity Model: Guidelines for Improving Software Process.* Carnegie Mellon University/Software Engineering Institute, 1995.

[Pereira 97] Pereira, L. F. S. M.; Aragon, D. F. *A Aquisição de Conhecimento e o Processo de Modelagem.* Relatório Técnico, ILTC, Niterói-RJ, n. 02, 46p., Dezembro, 1997.

[Pressman 94] Pressman, R. S. *Software Engineering: A Practitioner's Approach.* Edição Européia, McGraw-Hill, 1994.

[Rezende 98] Rezende, S. O.; Pugliesi, J. B. *Aquisição de Conhecimento Explícito ou Manual.* Notas do ICMC-USP - Série Computação, n. 37, Março, 1998.

[Rocha 97] Rocha, C. A. J.; Padilha, T. P. P.; Horst, P. S.; Rezende, S. O. *Processo de Desenvolvimento de Sistemas Baseados em Conhecimento Considerando Aspectos de Qualidade.* Anais da SEPAI - Semana Paraense de Informática, Belém, Dezembro, 1997.

[Tuthill 90] Tuthill, G. S. *Knowledge Engineering: Concepts and Practices for Knowledge-Based Systems.* TAB Books Inc., 1990.