O Impacto dos COTS no Processo de Engenharia de Requisitos

Fabrízia M. de Sousa[1], Fernanda M. R. de Alencar[2], Jaelson F. B. Castro[1]

[1]UFPE - Universidade Federal de Pernambuco
CCEN - Departamento de Informática
Av. Prof. Luiz Freire, s/n - Cidade Universitária
Recife - PE - CEP: 50.740-540
E-mail: {fms, jbc}@di.ufpe.br

[2] UFPE - Universidade Federal de Pernambuco
CT - Departamento de Eletrônica e Sistemas
Rua Acadêmico Hélio Ramos, s/n - Cidade Universitária
Recife - PE - CEP: 50.740-530
E-mail: fmra@npd.ufpe.br

Resumo Surge uma nova forma de desenvolvimento de software baseada no uso de software prontos, já existentes no mercado, chamados componentes COTS (Commercial Off-The-Shelf). O sucesso dessa nova metodologia de desenvolvimento tem sido demonstrado pelo uso crescente em importantes organizações, incluindo governo, indústria e comércio. Redução de custos e de prazos são os principais benefícios prometidos por esta nova tecnologia de desenvolvimento de software. Por outro lado, existem riscos associados ao novo processo. Nesse trabalho é apresentado o processo de desenvolvimento de software baseado em COTS e suas implicações na engenharia de requisitos. É feita uma análise das fases tradicionais do processo de engenharia de requisitos para adaptar as novas características de desenvolvimento introduzidas pelos componentes COTS.

Palavras chaves*: Metodologia de Desenvolvimento, Requisitos, COTS*

Abstract A new way for software development has been proposed. It is based on the use of existing software products, already available in the market, COTS (Commercial Off-The-Shelf) components. The success of this new development methodology has been demonstrated by the growing use of it in important organizations including government, industry and businesses. Reduction of costs and of time are the main benefits promised by this new software development technology. On the other hand, there are risks associated to this new process. In this work we present the software process based on COTS and its implication in the Requirement Engineering. It's done an analysis of the requirements engineering traditional phase for fit new features development insert by COTS components

Key Words: *Development Methodology, Requirements, COTS*

1. Introdução

Na literatura disponível, selecionamos as seguintes definições de *COTS (Commercial Off-The-Shelf)*: "Software de terceiros" [Voas 1998b], "Software comercial, desenvolvido sem nenhuma aplicação específica em mente" [McDermid 1998], "Software comercial, com código fonte não disponível e com versões periódicas" [Deifel 1999a] e "Software que existe, está disponível para o público e pode ser comprado ou licenciado" [Oberndorf 1997].

O desenvolvimento usando componentes COTS considera que a maior parte da funcionalidade necessária de um sistema deva ser adquirida de terceiros enquanto o paradigma tradicional assume que praticamente toda a funcionalidade deve ser desenvolvida. A filosofia COTS se concentra na produção de componentes competitivos e fáceis de integrar, muitas vezes usando componentes de outros fornecedores. A abordagem de produzir software baseado em componentes já é usada com maturidade em setores industriais como telefonia, automotores e hardware. As principais motivações para o incentivo a utilização de COTS tem sido a proposta de redução de custos de desenvolvimento dos produtos desejados e a proposta de redução e cumprimento dos prazos de elaboração. Outras características que também colaboram para o aceite dessa nova tendência de desenvolvimento de software são: (i) Dificuldade de desenvolvimento de software complexo e extenso, isto é com mais de 300.000 linhas de código; (ii) A disponibilidade de COTS alternativos de domínio específico ou genérico; (iii) A maturidade dos componentes COTS; (iv) A interoperabilidade de componentes COTS para padrões de mercado; e (v) O interesse conjunto da indústria, comércio e governo.

O aumento considerável no número de organizações que adotam uso de componentes COTS no desenvolvimento de software fortalece a credibilidade na obtenção de benefícios com essa nova forma de desenvolvimento. Governo, indústria, comércio e instituições de pesquisa têm demonstrado interesse pelo tema.

Na década de 90 importantes organizações, como NASA e DoD (Departament of Defence dos Estados Unidos), declaram ter obtido sucesso com redução de custos e prazos de desenvolvimento de software devido ao uso de componentes COTS [Oberndorf 1997]. Tanto que em 1994 foi estabelecido uma política de incentivo ao uso de COTS no DoD. Em 1998, McDermid chegou a declarar que dependendo da aplicação, COTS pode suportar 80% da funcionalidade com custo de apenas 2% do desenvolvimento tradicional [McDermid 1998].

A comunidade de pesquisa mostra seu envolvimento com componentes COTS através de eventos internacionais que destacam COTS como área de interesse. Como exemplos de eventos com contribuições para a área de COTS destacamos : *Annual Reliability and Maintainability Symposium* (1996), *Fifth International Symposium on Assessment of Software Tools and Technologie*s (SAST'97), *International Conference on Software Engineering* (ICSE'97), *COTS and Safety Critical Systems* (1997), *NRC-CNRC Software Engineer Seminar* (1998), *International Workshop on Requirements Engineering: Foundation for Software Quality* (REFSQ'98), REFSQ'99 e ICSE'99. Além desse eventos, existem projetos relativos a COTS em universidades e centros de pesquisas conceituados.

Existem dois tipos de sistemas baseados em COTS [Wallnau 1998]: (i) soluções COTS que estão prontas para uso; e (ii) COTS integrados que são componentes que precisam ser adaptados e integrados em uma sistema maior. Os

desafios se encontram exatamente nesse segundo tipo de sistemas, ou seja em COTS integrados.

O processo de desenvolvimento de software, considerando COTS integrados, possui dois enfoques com características e objetivos distintos : (i) desenvolvimento de sistemas baseados em componentes COTS, que produz um sistema maior a partir de componentes de mercado; e (ii) desenvolvimento de componentes COTS que gera para o mercado componentes confiáveis e fáceis de integrar. O processo de desenvolvimento nos dois casos são distintos e mudam também, com relação ao processo de desenvolvimento de software tradicional. Para o desenvolvimento baseado em COTS foi proposto um novo ciclo de vida definido pelas fases de avaliação, seleção, adaptação, junção e atualização. Para o desenvolvimento de componentes COTS não existe um novo modelo proposto, o processo é feito baseado no ciclo de vida tradicional e no novo ciclo de vida proposto para desenvolvimento de sistemas baseado em COTS.

Nesse trabalho iremos investigar quais as implicações do advento da tecnologia COTS nas etapas do processo da engenharia de requisitos, ressaltando as principais diferenças entre o processo de engenharia de requisitos tradicional e o novo processo utilizando COTS. Investigaremos requisitos tanto para o caso de desenvolvimento de software baseado em COTS, como requisitos para desenvolvimento de componentes a serem disponibilizados no mercado como COTS. Requisitos para sistemas que se baseiam no uso de componentes COTS necessitam se integrar com as fases propostas pelo novo ciclo de desenvolvimento (avaliação, seleção, adaptação, junção e atualização). Requisitos para a construção de componentes COTS devem ser identificados a partir de diferentes contextos, como mercado, sistema e desenvolvedor.

Na seção 2 apresentamos o processo tradicional de engenharia de software. Na seção 3 é descrito o desenvolvimento baseado em COTS, incluindo características, vantagens, desvantagens e modelos propostos. Na seção 4 é feita uma avaliação quanto às possíveis alterações no processo da engenharia de requisitos devido à tecnologia COTS, sendo apresentadas na seção 5 algumas conclusões.

2. Processo Tradicional da Engenharia de Requisitos

No processo tradicional de desenvolvimento de sistemas e, principalmente, de software, a engenharia de requisitos ocupa destacada posição. Representa a base para o desenvolvimento de qualquer tipo de produto, dela dependendo fundamentalmente a satisfação do produto às necessidades e desejos dos clientes contratantes, bem como sua qualidade. Sabe-se não haver um processo ideal para requisitos, contribuindo para tanto, a maturidade técnica, a cultura organizacional e o domínio da aplicação, mas algumas atividades básicas, bem definidas, do processo da engenharia de requisitos sempre existirão. Desta forma é que a fase de definição dos requisitos e, como conseqüência, o documento de requisitos, representam o primeiro estágio do desenvolvimento, enquanto, que o sistema de software produto, propriamente, é resultante dos estágios posteriores do desenvolvimento.

No desenvolvimento tradicional [Castro 1995], como pode ser visto na figura 1, são distinguidas quatro atividades básicas para o processo da engenharia de requisitos:

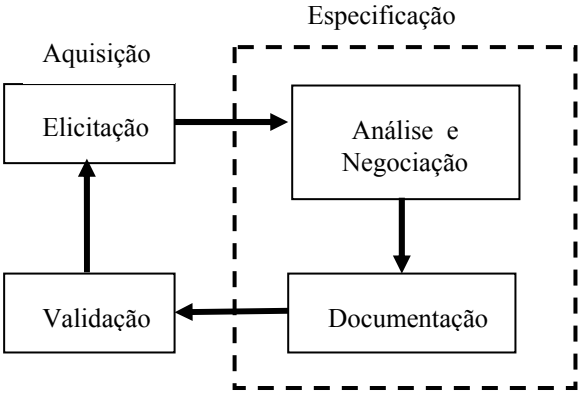

Fig. 1 - Atividades do Processo Tradicional da Engenharia de Requisitos.

(i) *Elicitação*, onde procura-se levantar os requisitos existentes e pretendidos pela organização, seja eles funcionais e não-funcionais; (ii) *Análise e Negociação de Requisitos*, onde faz-se análise sobre os requisitos levantados na fase anterior, descobrindo-se possíveis conflitos e necessidades de elicitar outros requisitos, fatos estes negociados com todas as partes envolvidas; (iii) *Documentação dos Requisitos*, através de um documento que sirva como base para o desenvolvimento do sistema requerido; e (iv) *Validação dos Requisitos*, onde serão levantadas as possíveis inconsistências e determinada a completude ou não dos requisitos levantados. Nessas atividades, durante todo o ciclo de desenvolvimento a interação entre as partes, contratante e contratados, deverá ser sempre ativa. Um bom processo de comunicação é pré-requisito para que o desenvolvimento do produto desejado seja adequado às reais necessidades e anseios do contratante.

Sabe-se que a depender do tamanho, da complexidade e da criticalidade do sistema a ser desenvolvido, os desafios a serem vencidos serão muitos. Todavia, da qualidade e da maturidade do processo da engenharia de requisitos dependerá o sucesso ou insucesso do sistema sob desenvolvimento.

3. Desenvolvimento de Software com uso de COTS

Um princípio básico do desenvolvimento de software com uso de componentes COTS é adquirir componentes para a maior parte da funcionalidade desejada, ao invés de desenvolver. O processo de desenvolvimento, de uma forma

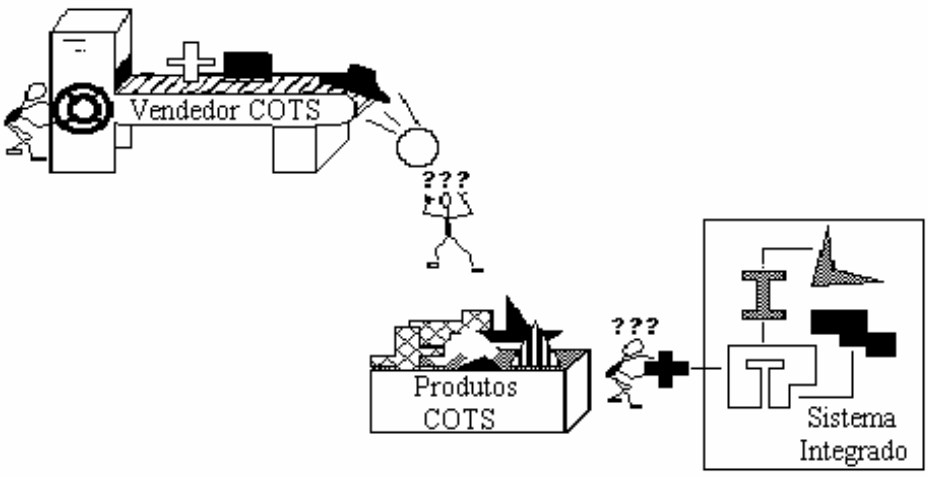

Fig. 2 : Desenvolvimento baseado em COTS.

Softwares desenvolvidos com o uso de componentes COTS são ricos em funcionalidades, utilizam tecnologia moderna, estão disponíveis para uso e por custos de aquisição previsíveis. Entretanto, existem desvantagens como taxa de licença de uso, dependência do desenvolvedor, falta de controle sobre novas versões (upgrades), integração não trivial e consumo de recursos computacionais (como memória e disco) por funcionalidades disponíveis, mas desnecessárias para alguns usuários. Software desenvolvido de forma tradicional, com o desenvolvimento de toda a funcionalidade desejada, apesar de não ter dependência de terceiros, permitindo controle de desenvolvimento e controle de segurança, são historicamente caros, de desenvolvimento imprevisível e de difícil portabilidade.

3.1. Modelos Propostos

Alguns modelos de ciclo de vida de sistemas desenvolvidos com uso de componentes COTS foram propostos. Entre eles, apresentados em 1997, temos o modelo *COTS Intensive System* (CIS) e o *COTS Integreted System Development* (CISD).

O modelo CIS, apresentado na figura 3, descreve cinco fases no ciclo de vida de desenvolvimento [Wallnau 1998]: (i) *Mercado de COTS* que corresponde à construção de componentes independentes por diversos fabricantes do mercado; (ii) *Qualificação* que representa a seleção de componentes com funcionalidade e capacidade de integração apropriadas para as necessidades da organização; (iii) A*daptação* que corresponde ao ajuste de componentes baseados em parâmetros da organização; (iv) *Integração* que representa a junção de componentes formando o sistema desejado pela organização; e (v) A*tualização* que trata da substituição de um componentes do sistema integrado ou mesmo substituição de versão.

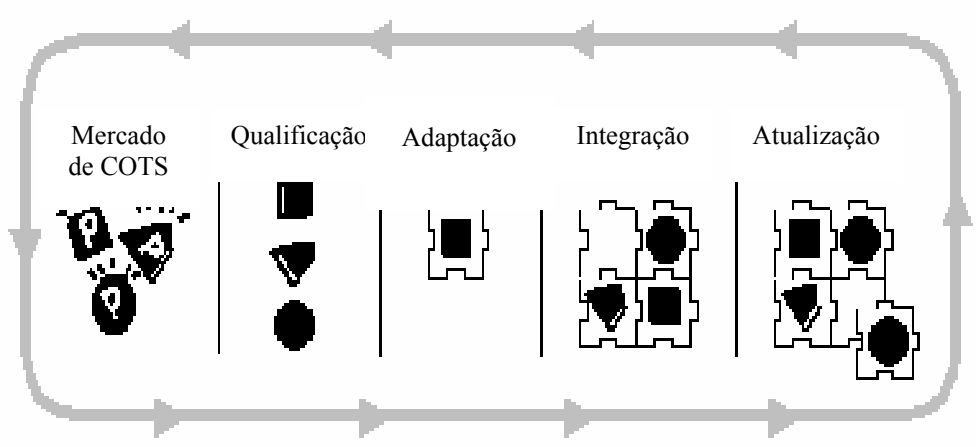

Fig. 3 : Modelo CIS (COTS Intensive System).

O modelo CISD, apresenta na figura 4.a sua interface externa constituída por COTS disponíveis no mercado e definição de requisitos. Na figura 4.b são apresentadas as fases do ciclo de vida de desenvolvimento [Tran 1997b]: (i) *Identificação* que enumera componentes disponíveis no mercado como candidatos a serem selecionados; (ii) *Avaliação* que faz um estudo relativo da funcionalidade e capacidade de integração dos componentes, identificando e selecionando aqueles que sejam apropriados para as necessidades da organização; e (iii) *Integração* que faz a integração de vários componentes, incluindo ajustes necessários.

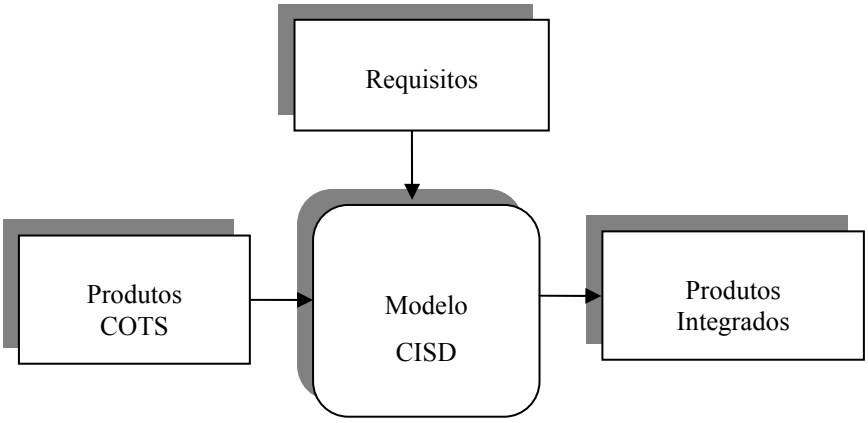

Fig. 4.a) Interfaces externas do modelo CISD

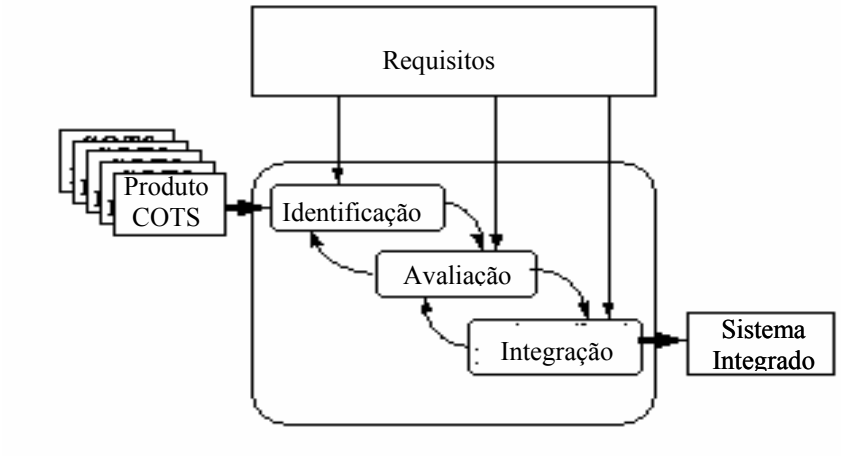

Fig. 4.b) Fases do modelo CISD.

3.2. Aspectos Relevantes ao Desenvolvimento COTS

Dentre os principais aspectos associados ao desenvolvimento baseado em COTS, destacam-se: (i) os riscos do processo de avaliação e integração; (ii) a garantia de confiabilidade de todo o sistema; e (iii) o custo final do sistema.

A avaliação e seleção de componentes verifica a funcionalidade e a capacidade de integração desejada. Precisa ser feita de forma rápida, usando as informações disponíveis, que podem ser insatisfatórias, e com grau de rigor apropriado para a aplicação em questão. É um processo de escolha subjetivo e sujeito a falhas porque é difícil examinar, em um prazo curto, componentes concorrentes e aparentemente semelhantes.

Para eliminar os risco do processo de avaliação e seleção, pode ser usado o método de certificação [Voas 1997a][Voas 1998c], que guia o processo de avaliação através de testes de caixa-preta, injeção de falhas e testes feitos em ambientes operacionais.

Os riscos relativos ao processo de integração são devido a falhas que podem ocorrer com as entradas ou saídas dos componentes, muitas vezes devido a defeitos (bugs) dos próprios componentes, comprometendo a ligação entre dois componentes e quebrando, consequentemente, a integração geral do sistema.

Para eliminar risco do processo de integração foram propostas estratégias como, capas de proteção (wraper) [Tran 1997a][Tran 1997b][Voas 1998c], análise atenuação de riscos [Voas 1997b][Zhong 1998][Kropp 1998] e localização de falhas [Hissam 1997][Voas 1998a].

O problema com a confiabilidade de todo o sistema é conseqüência direta de defeitos (bugs) e limitações dos componentes. Um sistema que usa componentes COTS herda seus defeitos e limitações, diminuindo sua confiabilidade. Pode-se dizer também que a confiabilidade e segurança de um sistema depende da confiabilidade e segurança de cada componente que use.

Uma falsa idéia que se tem do desenvolvimento com COTS é que o custo do processo é apenas o custo de aquisição do componente. Uma avaliação de custos realística para esse tipo de desenvolvimento é descrita no modelo de custos COCOTS (*Constructive COTS*) [COCOTS] uma variação do modelo de custos COCOMO [Boehm 1981].

O COCOTS classifica a utilização de componentes pela organização em *ferramenta*, *infra-estrutura* e *componentes*. Para cada categoria de uso apresenta fórmulas de cálculo e tabelas de referência para serem calculados os seguintes custos: *avaliação*, *ajuste*, *integração* e *incorporação* de novas versões.

4. Alterações no Processo de Requisitos

Podemos dizer que no desenvolvimento tradicional a fase de definição dos requisitos é crítica e essencial a construção de qualquer produto. Pelo que analisamos com a tecnologia COTS de desenvolvimento, esta fase também é necessária e importante. O que acontece é que muitos pontos que antes precisavam ser bem definidos numa fase inicial, não são necessários nos primeiros estágios do desenvolvimento, uma vez que já dispomos de componentes acabados. Além da não existência de um relacionamento mais aprofundado entre as organizações

desenvolvedoras, nem tão pouco de uma única autoridade para o projeto. Essa é a visão que se tem quando se está trabalhando com desenvolvimento de sistemas baseado em COTS. Os requisitos e a arquitetura são definidas de forma concorrente [Vigder 1998]. Quando estamos desenvolvendo os componentes COTS é diferente.

Tomando-se por base a nova tecnologia COTS, apresentamos a seguir uma análise a respeito de como ficam as atividades do processo da engenharia de requisitos, considerando o desenvolvimento de sistemas baseados em COTS e o desenvolvimento de componentes COTS.

4.1. Requisitos para Desenvolvimento de Softwares Baseados em COTS

Com base no modelo CIS, apresentado anteriormente na figura 3, podemos inicialmente dizer que a determinação dos requisitos é feita de maneira exógena ao processo de desenvolvimento. Para o início do desenvolvimento, parte-se de uma descrição abstrata dos requisitos a fim que se tenha a almejada funcionalidade. Explicitamente, não existe no modelo CIS, qualquer fase do desenvolvimento voltada diretamente à definição dos requisitos do sistema requerido. Entretanto, constatamos neste modelo, atividades do processo da engenharia de requisitos tradicional: (i) na fase *qualificação* ocorre a atividade de *análise dos requisitos,* pois é preciso garantir que a funcionalidade dos componentes COTS selecionados no mercado, satisfaçam aos requisitos mínimos elicitados antes do início do processo CIS; (ii) na fase de *adaptação* temos a atividade de *elicitação,* porque é necessário levantar requisitos que não foram enquadrados nos componentes qualificados; e por fim, (iii) na fase *integração,* existe a atividade de *validação,* considerando que os ajustes feitos na fase *adaptação*, precisam ser consolidadas.

O modelo CISD, entretanto, é mais detalhado no tocante a requisitos, mesmo considerando requisitos como processo externo. Esse modelo apresentado anteriormente nas figuras 4.a e 4.b mostra que a definição de requisitos participa de forma diferente nas fases do ciclo de vida do modelo. *Identificação*, *avaliação* e *integração* são as fases que compõem esse modelo e para cada uma dela a definição de requisitos é vista de forma diferente.

A fase de *identificação* seleciona componentes COTS para serem posteriormente avaliados. Nessa fase é necessário se fazer uma análise do documento de requisitos visando particionar os requisitos funcionais em domínios de serviços, facilitando assim a identificação de componentes, que passa a ser feita por domínio.

A fase de *avaliação* compara, em ambiente operacional (por prototipação), a funcionalidade e capacidade de integração dos componentes previamente selecionados. Nessa fase é feito um particionamento dos requisitos, gerando-se a partir dessa divisão estágios de avaliação, como : (i) avaliação da funcionalidade que verifica, por testes, se o componente satisfaz à funcionalidade exigida; (ii) avaliação da interoperabilidade para se certificar da capacidade de integração dos componentes; e (iii) avaliação de desempenho para avaliar se os requisitos não funcionais de desempenho foram atendidos depois da integração (por protótipo) de diferentes componentes.

A fase de *integração* corresponde ao desenvolvimento e teste de "adaptadores" para formar o sistema integrado final. Nessa fase a participação dos requisitos diz respeito à certificação de atendimento dos requisitos não funcionais.

Comparando esse processo de desenvolvimento com o processo tradicional de requisitos podemos dizer que tarefas típicas de requisitos estão distribuídas no novo ciclo de desenvolvimento. Na fase de *identificação* do modelo CISD, a classificação dos requisitos por domínio é um meio de atribuir prioridades, tarefa típica da fase de negociação de requisitos. A fase de *avaliação* do CISD faz uma validação parcial de requisitos ao testar requisitos funcionais e não funcionais do sistema. A fase de *integração* do CISD é uma validação final de requisitos, com destaque para os requisitos não funcionais. Outro aspecto relevante de requisitos para sistemas baseados em COTS é o nível de detalhamento das fases de *elicitação*, *especificação* e *validação* de requisitos. Não há necessidade de elicitar e especificar com muitos detalhes, já que os componentes fornecem funcionalidades prontas, entretanto validação de requisitos é mais detalhada, devido à importância dos testes de integração.

4.2 Requisitos para Desenvolvimento de Componentes COTS

Uma importante característica dos componentes COTS é que eles são desenvolvidos para um mercado inteiro, ao invés de um único cliente. Isso afeta de forma significativa o processo de desenvolvimento, em todas as fases onde ocorra interação entre desenvolvedor e cliente. Outros fatores importantes relativos a componentes COTS é que eles são distribuídos em versões e, normalmente, fazem parte de um sistema produto maior.

Diante dessas novas características, foi apresentado no REFSQ'98 uma visão diferente para as fases tradicionais do processo de Engenharia de Requisitos [Deifel 1998]. Considerando a Engenharia de Requisitos composta pelas fases *elicitação*, *negociação*, *documentação* e *validação*, temos as seguintes mudanças relativa a requisitos para desenvolvimento de componentes COTS :

No processo tradicional de requisitos, a fase de *elicitação* é feita por técnicas que se baseiam na premissa de existência de comunicação entre desenvolvedor e cliente, tais como, entrevistas, questionários, análise de protocolos, participação ativa dos usuários, etc. Entretanto, para componentes COTS, como não existe comunicação entre desenvolvedor e Cliente, a *elicitação* passa a ser feita dividida por *contexto* e buscando informações em *canais de comunicação.* Contextos são divisões de requisitos, como por exemplo, requisitos do mercado, do sistema e do desenvolvedor ou ainda requisitos do domínio da aplicação, do contexto de utilização e do sistema básico [Deifel 1999b]. Canais de comunicação são locais possíveis de conter informações relativas ao componente que será construído, tais como, workshops, vendedores, hotlines, suporte, manutenção, outros softwares e versões anteriores [Deifel 1998].

A fase de *negociação*, que consolida os requisitos, diferencia-se pela inexistência de um usuário que colabore no processo de checagem da consistência dos requisitos e pelo diferente papel da priorização. As fontes de informações para essa fase passam a ser os canais de comunicação. A priorização, antes feita sob todo o conjunto de requisitos, passa a ser feita pela distribuição de requisitos sobre versões, liberando-se para as primeiras versões os requisitos mais importantes. A distribuição dos requisitos por versão adiciona ao processo um fator de dificuldade relativo à

compatibilidade de versões, que precisa ser cuidadosamente elaborada. Em [Deifel 1999a] é apresentado um modelo de requisitos para planejamento de versões.

A fase de *documentação,* que traduz requisitos do cliente em especificações técnicas, antes feita para todos os requisitos do sistema, passa a ser feita apenas para os requisitos de uma versão. A distribuição de requisitos por versão é feita na fase de *negociação* e coloca os requisitos mais importantes nas primeiras versões. A documentação por versão, apesar de ser menor que a documentação para todo o sistema, precisa garantir compatibilidade com versões passadas e futuras, constituindo-se assim, em uma nova obrigação dessa fase.

A fase de *validação* têm como propósito garantir que o software desenvolvido esteja correto, verificando se a funcionalidade solicitada foi fornecida e se está implementada corretamente. No processo tradicional de requisitos a checagem entre a funcionalidade desejada e a funcionalidade fornecida é feita pela apresentação do produto final ao cliente, entretanto para COTS, como não existem clientes, essa confirmação só é obtida pelo sucesso de vendas do componente. A validação referente à implementação correta das funcionalidades é feita por testes, tanto no processo tradicional de requisitos como nos requisitos para COTS.

5. Conclusão

A tecnologia COTS fornece novos direcionamentos ao processo de desenvolvimento de software. Desenvolver software baseado no uso e integração de COTS e desenvolver componentes COTS confiáveis e fáceis de integrar são enfoques diferentes de desenvolvimento de software utilizando a tecnologia COTS.

Para representar o ciclo de vida do desenvolvimento baseado em COTS forma propostos os modelos CIS e CISD. O CIS é definido pelas fases *produção de mercado COTS, qualificação, adaptação,* i*ntegração* e a*tualização* e CISD definido pelas fases de *identificação*, *avaliação* e *integração*. Apesar de dividir e nomear o ciclo de vida em fases diferentes, ambos os modelos são comuns ao definir as tarefas de seleção e integração de componentes no processo de desenvolvimento. Para o desenvolvimento de componentes COTS entretanto, não temos um modelo de ciclo de vida específico.

O impacto dos componentes COTS no engenharia de requisitos é relevante. Nesse novo tipo de desenvolvimento o processo de requisitos deve ser feito considerando os novos objetivo do desenvolvimento, tendo assim um nova visão das fases tradicionais de requisitos. Quando se elabora requisitos para integrar COTS, diminui-se o nível de detalhamento necessário nas fases tradicionais de elicitação e especificação, entretanto aumenta-se o detalhamento da fase de validação, gerado devido à necessidade de integração de componentes. Quando se produz requisitos para desenvolver COTS, devido a inexistência de interação com o cliente, as fases de requisitos são elaboradas a partir de contexto e canais de comunicação. Outra mudança na definição de requisitos para COTS é a determinação de prioridades, que coloca os requisitos mais importantes nas primeiras versões e deve suportar compatibilidade de versões.

A contribuição da Engenharia de Requisitos para o desenvolvimento de softwares que utilizam COTS ou para o desenvolvimento de componentes COTS poderia ser maior se o documento de requisitos fosse elaborado com o propósito de

facilitar os principais desafios desse tipo de desenvolvimento, que são a avaliação e a integração. Isso representa a nossa sugestão de trabalhos futuros.

Referências

Boehm, Barry W. *Software Engineering Economics*. Prentice Hall. 1981.

Castro, J. F. "Introdução a Engenharia de Requisitos". XV Congresso da Sociedade Brasileira de Computação, JAI 95 – XIV Jornada de Atualização em Informática. Canela, RS – Brasil. 1995.

COCOTS – Constructive COTS Model. Center for Software Engineering, University of Southers California, USA. (http://sunset.usc.edu/COCOTS/cocots.html)

Deifel, Bernhard. "*Requirements Engineering for Complex COTS*. Positional Paper. REFSQ'98.

Deifel, Bernhard. "*A Model for version Planning of COTS*. Extended Abstract. SCE'99.

Deifel, Bernhard. "*Supporting Reuse and Flexibility in COTS Variation Development*. REFSQ'99.

Hissam, Scott. "*Case Study: Correcting System Failure in a COTS Information Systems*". Software Engineering Institute, Carnegie Mellon University, USA. September, 1997.

Kropp, Nathan P.; Koopman, Philip J. and Seiwiorek, Daniel P. "*Automated Robustness Testing of Off-the-Shelf Software Components*". In Proceeding of the Fault Tolerant Computing Symposium. Germany. 1998.

McDermid, John (Interview). "*The Cost of COTS*". IEEE Computer Society. June, 1998. Vol 31. Number 6. Pages 46–52.

Oberndorf, Tricia. "*COTS and Open Systems – an Overview*". Software Engineering Institute, Carnegie Mellon University, USA. January, 1997.

Tran, Vu and Liu, Dar-Biau. "*A Procurement-centric Model for Engineering Component-based Software Systems*". Proceedings of the Fifth International Symposium on Assessment of Software Tools - SAST'97. Jun, 1997.

Tran, Vu and Liu, Dar-Biau. "*A Risk-Mitigating Model for The Development of Reliable and Maintainable Large-Scale COTS Integrated Software Systems* ". Proceedings of the 1997 Annual Reliability and Maintainability Symposium – The International Symposium on Product Quality and Integrity. Jan, 1997.

Vigder, Mark. "*An Architecture for COTS Based Software Systems*", National Research Council Canada, Institute for Information Technology, Ottawa, Ontario, Canada, Prepared for Defense Research Development Branch. Nov., 1998.

Voas, Jeffrey M. "*Certifying Off-the-Shelfs Software Componentes*". IEEE Computer Society. June, 1998. Vol 31. Number 6. Pages 53–59.

Voas, Jeffrey M. "*The Challenges of Using COTS Software in Component-Based Development*". IEEE Computer Society. June, 1998. Vol 31. Number 6. Pages 44-45.

Voas, Jeffrey. "*A Defensive Approach to Certifying COTS Software*". Technical Report.

Reliable Software Technologies Corporation, Sterling VA USA. August, 1997.

Voas, Jeffrey. "*Defensive Approaches to Testing Systems that Contain COTS and Third-Party Functionality*". Reliable Software Technologies Corporation, Sterling VA USA. June, 1998.

Voas, Jeffrey. "*Mitigating the Potential for Damage Caused by COTS and Third-party Software Failures*". Reliable Software Technologies Corporation, Sterling VA USA. August, 1997.

Wallnau, Kurt C.; Carney, David and Pollak, Bill. *"How COTS Software Affects the Design of COTS-Intensive Systems*". Software Engineering Institute, Carnegie Mellon University, USA. June, 1998.

Zhong, Qun and Nigel, Edward. "*Security Control for COTS Components.* IEEE Computer Society. June, 1998. Vol 31. Number 6. Pages 44 – 73.