

5

Processo de Reificação e de Desenvolvimento com ACCA

Resumo

Este capítulo visa esclarecer e descrever atividades existentes em um processo de desenvolvimento de soluções a partir de componentes de software já existentes. Primeiramente, descreve-se o processo de reificação de ACCA. Finalmente, utilizando ACCA são descritas as atividades de um processo de desenvolvimento de soluções a partir de componentes de software.

5.1.

Processos de Desenvolvimento de Software

Um processo de desenvolvimento de software define um conjunto de atividades e resultados associados que levam a produção de um produto de software (Sommerville, 2001). Segundo Jacobson et al. (1999), um processo de desenvolvimento define quem está fazendo o quê, quando e como para construir ou evoluir um software. Um processo eficaz deve prover um roteiro para um desenvolvimento eficiente de um software com qualidade.

O objetivo deste capítulo é apresentar uma seqüência de atividades capazes de organizar e definir pontos relevantes para a utilização de ACCA. Para isto, são definidas duas seções: o processo de reificação de ACCA e o processo de desenvolvimento a partir de componentes de software. O processo de reificação de ACCA corresponde às etapas de criação, elaboração, implementação, teste e avaliação existentes para a utilização da arquitetura ACCA em domínios específicos de aplicação. O outro processo corresponde ao conjunto de atividades relevantes para o desenvolvimento de aplicações utilizando componentes de software.

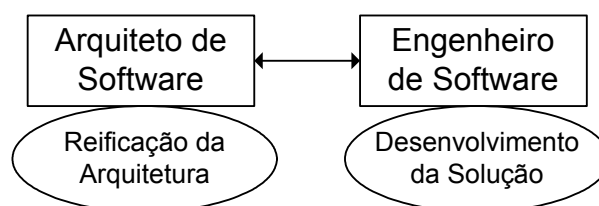


Figura 47 – Papéis no processo de desenvolvimento associado a ACCA

Recomenda-se que o processo de reificação de ACCA seja executado por um arquiteto de software, enquanto que o processo de desenvolvimento de aplicações seja executado por um engenheiro de software (Figura 47).

5.2.

Processo de Reificação de ACCA

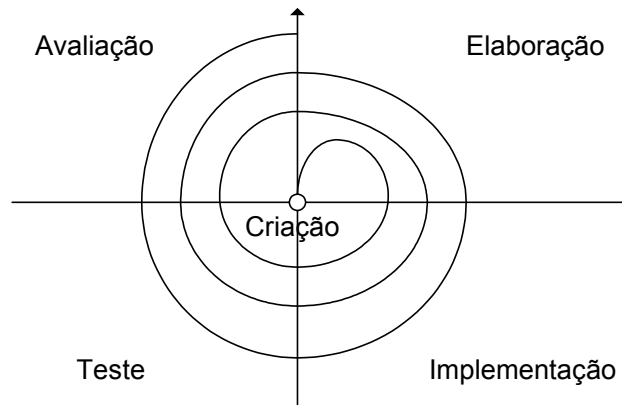


Figura 48 – Ciclo de vida de ACCA

O processo de reificação (Figura 48) é baseado na descrição da implementação do padrão *Layer* proposto em Buschman (1996). Ele é composto pelas etapas de criação, elaboração, implementação, teste e avaliação. É importante observar que as etapas deste processo de reificação não são obrigatórias, isto é, sua execução depende do problema em questão e de decisões de projeto tomadas por um arquiteto de software.

Etapa Criação da Abstração de ACCA (Figura 48): a princípio, esta primeira etapa já foi executada e corresponde a escolha do paradigma coordenação / composição / artefatos de software como agrupamento de abstrações básicas a serem utilizadas em camadas hierárquicas da arquitetura de software. A partir deste paradigma, três níveis de abstração foram criados para a organização da solução, cada nível corresponde a uma camada da arquitetura de software proposta. A cada camada foi atribuído um nome: questões de coordenação estão presentes na camada mais abstrata chamada de Coordenação, questões de composição estão presentes na camada Composição, e por fim, componentes de software estão presentes na camada mais concreta chamada de Artefatos de Software.

Etapa Elaboração de ACCA (Figura 48): nesta etapa, questões específicas do domínio do problema são identificadas. Em cada camada, as abstrações criadas são projetadas em função dos recursos disponíveis, das demandas apresentadas e

de decisões de projeto. Além das abstrações, os serviços oferecidos por cada camada, juntamente com a sua forma de uso, devem ser especificados. É importante observar que, ao final desta etapa, todas as interfaces entre as camadas estejam devidamente estabelecidas. Abaixo, para cada camada, é descrito sucintamente um resumo da realização das abstrações.

A camada Artefatos de Software deve ser vista como um grande catálogo ou repositório de componentes de software. As abstrações desta camada são os componentes de software disponíveis e o catálogo utilizado para organizar e disponibilizar os componentes de software. Além da escolha de repositórios, as diferentes formas de catalogação de artefatos devem ser determinadas e configuradas. Também deve ser avaliada a viabilidade da utilização de uma linguagem específica para a realização de consultas ao catálogo de componentes de software. Diferentes tecnologias podem conviver, sendo que a integração entre as unidades é feita pela camada superior de composição de software. É importante lembrar que a decisão por uma convivência entre um maior número de tecnologias distintas pode gerar um maior esforço de desenvolvimento na camada de Composição.

A camada Composição pode ser vista como um *middleware* de integração. Este *middleware* viabiliza a unificação das diferentes tecnologias de componentes de software utilizadas. Para isto, esta camada deve determinar a estrutura que realizará os padrões de composição, projetando as conexões que tornem possível a convivência entre as tecnologias específicas de componentes. Estas conexões correspondem ao projeto de integração dos diferentes tipos de artefatos de software, e geralmente podem ser reutilizadas.

Um exemplo de realização desta camada é o projeto de *frameworks* orientados a objeto que implementem uma estrutura básica de composição. Uma instância deste *framework* deve levar em consideração as tecnologias de componentes de software utilizados nas composições e os padrões de composição escolhidos ou necessários para a integração dos artefatos de software. Esta camada pode estender o serviço de busca de artefatos oferecido pela camada inferior e pode prever um serviço de catalogação de composições básicas a serem reutilizadas. Ainda neste ponto, é necessário determinar uma interface para o acesso a um serviço de uma composição. Esta interface é o meio pelo qual as

instruções de coordenação interagem com as construções da camada de composição.

Na camada de Coordenação é necessário determinar as diferentes formas de relacionar e estabelecer as interdependências entre as composições existentes. O conjunto de interdependências reflete o modelo de coordenação a ser utilizado pelo cliente da arquitetura. Neste ponto, deve ser avaliada a viabilidade da criação ou utilização de uma linguagem específica para configurar estas interdependências entre composições de software. Um exemplo de linguagem que pode ser utilizada para esta finalidade é a proposta de contratos de coordenação (Lano et al., 2002).

Apesar de não entrar em detalhes é preciso que, ao final desta etapa, seja proposta uma política de tratamento de erros e exceções entre as camadas.

Etapas Implementação, Teste e Avaliação de ACCA (Figura 48): a partir das especificações geradas pela etapa de elaboração, é possível implementar o projeto e avaliar os resultados obtidos a partir da proposta de arquitetura gerada.

Os testes correspondem ao processo de coleta de informações sobre o funcionamento da arquitetura. Com relação à avaliação, é importante verificar se as interfaces definidas entre as camadas são suficientes para que o fluxo de serviços desempenhe seu papel previsto. Após a avaliação das informações colhidas e do funcionamento geral da arquitetura, correções ou evoluções podem ser propostas para que, em uma nova iteração, elas sejam aplicadas.

Apesar da arquitetura não impor maiores restrições sobre a estrutura interna de cada camada, é importante que o seu desenvolvimento seja satisfatório, pois uma boa realização das abstrações propostas em uma única camada pode refletir em boas conseqüências para a solução de maneira geral.

Outro ponto interessante que pode contribuir para o controle da evolução de soluções é minimizar o acoplamento entre as camadas adjacentes. Ao isolar as camadas, espera-se que alterações específicas a uma questão fiquem restritas ao máximo possível a uma única camada.

Após a aprovação de uma versão de arquitetura, são catalogados todos os componentes disponíveis para composição, sendo que novos componentes podem ser incluídos conforme o aparecimento de novas necessidades ou o surgimento de alternativas úteis para o desenvolvimento de soluções.

Como descrito, o processo de reificação de ACCA corresponde a uma iteração ou ciclo completo de elaboração, implementação, testes e avaliação da

arquitetura proposta. Este processo segue a linha de evolução ou desenvolvimento iterativo. Ao final de cada iteração, dispõe-se de mais funcionalidades de interesse para o engenheiro de software. Este ciclo prevê que após a aprovação de todas as definições, pode ser necessário rever alguma decisão tomada nas etapas anteriores, quer seja por inadequação da resolução de alguns dos problemas, quer seja pelo surgimento de novos requisitos, ou quer seja pelo aperfeiçoamento da arquitetura já existente. Estas questões podem implicar em novos serviços sendo criados em cada camada e no estabelecimento de novos relacionamentos entre as abstrações.

É importante que depois de rever as suas realizações, verificar se as abstrações continuam preservadas em cada camada e se o paradigma proposto continua válido. Revisões sucessivas são feitas ao longo do ciclo de vida da arquitetura e das aplicações geradas a partir dela. Pela forma de trabalho proposta, estas alterações devem ser bem estudadas e bem validadas teórica e experimentalmente.

5.3.

Processo de Desenvolvimento a partir de Componentes de Software

A seguir, é proposta uma seqüência de atividades (Figura 49) que pode ser utilizada, em conjunto com o processo de reificação de ACCA, para o desenvolvimento de software baseado em componentes.

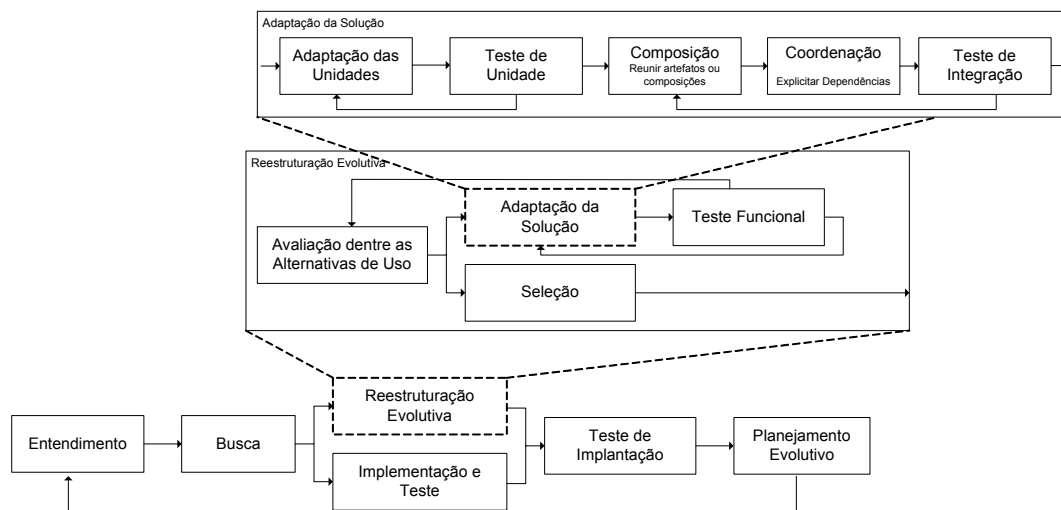


Figura 49 – Visão geral do processo de desenvolvimento de software

Cada atividade deve considerar e descrever, quando possível, o tratamento adequado para a evolução de uma solução. Estas atividades não precisam necessariamente ser executadas seqüencialmente e na ordem descrita na Figura

50. No entanto, esta representação é a forma proposta neste trabalho para o desenvolvimento de soluções a partir de componentes de software, e é considerada como referência neste trabalho.

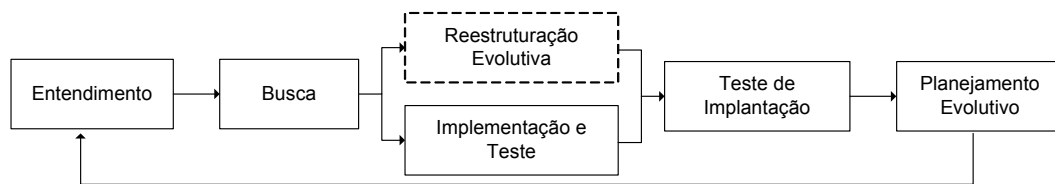


Figura 50 – Seqüência principal de atividades do processo de desenvolvimento

A atividade Entendimento (Figura 50) corresponde ao processo de reflexão e compreensão de um problema. Ela é seguida por um conjunto de fases que correspondem à proposição de uma abordagem para a solução do problema. Geralmente, é nesta atividade que uma solução complexa é conceitualmente decomposta em subunidades de menor complexidade. Estas subunidades, quando reunidas, representam a solução completa.

A Busca representa a atividade do processo de desenvolvimento que prevê o levantamento e a localização de componentes já existentes e passíveis de reutilização na solução. É neste momento que são consultados os catálogos ou repositórios, internos ou externos, disponíveis ao engenheiro de software de ACCA. Caso não existam componentes com o perfil adequado para a solução do problema, novos componentes devem ser implementados e testados, correspondendo às atividades de Implementação e Testes (Figura 50), respectivamente. Estas atividades, apesar de muito importantes, não fazem parte do escopo deste trabalho.

Após obter as unidades adequadas, caso elas ainda não estejam catalogadas na camada Artefatos de Software de ACCA, o engenheiro de software deve fazê-lo.

A menos que o componente tenha seguido algum padrão previamente estipulado para promover a sua reutilização, e a menos que o componente esteja plenamente adequado para resolver o problema em questão, é provável que seja necessário fazer alguma adaptação em Reestruturação Evolutiva (Figura 50, Figura 51). Esta adaptação é prevista e ocorre após a avaliação das diferentes características e alternativas possíveis para a reutilização das unidades.

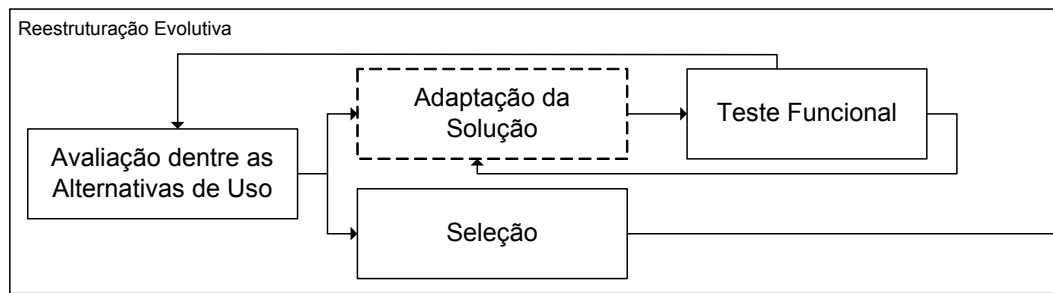


Figura 51 – Fase de reestruturação evolutiva da solução

Após a Adaptação da Solução (Figura 51), são feitos testes funcionais que têm como o objetivo verificar se realmente a solução atende às necessidades do engenheiro de software. Caso isto não ocorra, o componente volta à atividade de Adaptação da Solução. Caso a unidade atenda às necessidades previstas, a proposta de solução gerada é retornada para a atividade Avaliação dentre as Alternativas de Uso. A partir da existência de soluções adequadas para atender aos requisitos do problema, é por meio da atividade Seleção que será escolhida a melhor alternativa possível. Esta alternativa será recomendada e seguirá para as próximas fases deste processo.

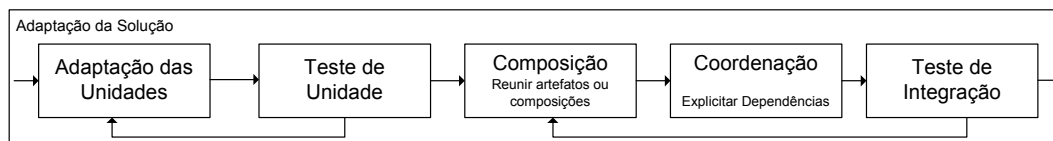


Figura 52 – Fase de adaptação da solução

Ainda na fase de adaptação da solução e após a aplicação de todas as adaptações necessárias, e da aprovação dos testes efetuados, é necessário reunir a unidade ao todo, por meio da atividade chamada de Composição (Figura 52). Esta atividade corresponde à utilização da camada de Composição de ACCA.

Após compor os artefatos necessários tem-se início a atividade de elaboração dos cenários de uso dos componentes de software. Esta atividade é chamada de Coordenação (Figura 52) e corresponde ao processo de configuração e explicitação de interdependências entre as composições de software disponíveis para a utilização. Esta atividade corresponde ao uso da camada de Coordenação de ACCA.

Após a composição estar integrada à solução já existente e as interdependências estarem explicitamente determinadas e configuradas, é necessário fazer testes de integração para verificar se o comportamento da reunião de unidades atende às necessidades iniciais (Figura 52). Caso o composto

coordenado esteja inadequado para a resolução do problema, na tentativa de resolver as diferenças, o processo retornará à atividade de Composição.

Após a Reestruturação Evolutiva (Figura 50), testes de implantação são executados para avaliar o funcionamento da solução em desenvolvimento no ambiente de execução. Esta avaliação é feita em conjunto com as demais aplicações já desenvolvidas e em produção, sendo útil para avaliar impactos e consequências desta nova aplicação.

Ao final do processo de desenvolvimento da solução baseada em componentes de software (Figura 50), é prevista uma última atividade que é chamada de Planejamento Evolutivo. Esta atividade corresponde à reflexão sobre a necessidade de novas alterações e quaisquer atividades de planejamento de configurações e mudanças relativas à solução. Geralmente, ao final desta atividade, o processo de desenvolvimento tem um novo ciclo iniciado.