Evolução de software baseada em avaliação de Arquitetura de Software.

Danielle P. Noronha Pontes[1], Reginaldo Arakaki [2]

[1]Escola Superior de Tecnologia – Universidade Estadual do Amazonas (UEA) / MINTER(UEA/USP), Manaus – AM - Brasil

[2] Engenharia da Computação e Sistemas Digitais - Universidade de São Paulo, São Paulo – SP – Brasil

dani@doctortech.com.br,reginaldo.arakaki@poli.usp.br

Resumo. *Este trabalho discorre sobre o estudo da utilização do método de avaliação ATAM como referência para evolução arquitetural de um sistema legado. O estudo apresentado está dividido em duas partes: a elaboração de um roteiro para evolução de software e a aplicação do roteiro em um ambiente real de um sistema para automação de linhas aéreas. Um dos objetivos é aplicar a correlação entre a evolução arquitetural e os requisitos não-funcionais. As decisões arquiteturais para a evolução do software são tomadas com base nas evidências obtidas na avaliação arquitetural realizada a partir dos atributos de qualidade estabelecidos como meta.*

1. Introdução

O processo de envelhecimento de um software é natural e inevitável, o que gera uma necessidade constante de evolução dos sistemas que precisam ser mantidos ativos por um período longo de tempo. Com o decorrer do tempo os sistemas corporativos instanciados no mercado podem apresentar problemas estruturais que afetam requisitos fundamentais de qualidade tais como: integração, flexibilidade, portabilidade, qualidade do serviço e segurança demandando uma evolução estrutural para atender as novas exigências e aumentar o tempo de vida do sistema. [Svahnberg, 2003]. Entretanto a alta complexidade dos sistemas dificulta as tentativas de substituição de plataforma, sistema ou fornecedor.

A evolução do software é um fato contínuo no ciclo de vida de um software, mas mudar sistemas sem técnica pode prejudicar alguns aspectos de qualidade da arquitetura. Para decidir como implementar as mudanças, é preciso usar um método que permita o controle da qualidade da arquitetura. O uso de um método estruturado ajuda a garantir que questões importantes serão tratadas antecipadamente. O método de avaliação *Architecture Tradeoff Analysis Method* (ATAM) tem como fundamento da avaliação estabelecer os atributos de qualidade desejados a partir dos pontos críticos do processo de negócio (referenciados como *business drivers*) [Clements et all 2009] e tratar os conflitos de escolha (*trade-off*) que surgem na tentativa de atender as necessidades de mercado.

Para tratar a evolução de forma estruturada são apresentados, neste trabalho, um roteiro de avaliação e os resultados de um exemplo prático. O roteiro para avaliação de arquitetura de software terá como referência o método de avaliação de arquitetura de software ATAM e como objetivo, gerar um plano de evolução a partir de sua

arquitetura, assegurando a longevidade do sistema no mercado. O exemplo prático será em um ambiente real de uma empresa de aviação.

2. Evolução de Software e Arquitetura de Software

Bass Graaf em [Graaf, 2007] distingue claramente dois tipos de transformações durante a vida de um software. Segundo o autor os modelos e processos existentes envolvem tipicamente transformações verticais, do abstrato para o concreto, como acontece no ciclo de desenvolvimento de software. Por outro lado atividades como manutenção e evolução, típicas em qualquer software, envolvem transformações horizontais como a migração do sistema de uma plataforma para outra. Este trabalho propõe o planejamento para execução de transformações horizontais que envolvem aspectos arquiteturais do sistema. Chávez (2009) em seu trabalho cita os autores Krutchen, Obbink e Staford (2006) que argumentam que do ponto de vista prático é possível controlar e supervisionar o desenvolvimento e evolução de sistemas através da arquitetura de software. De acordo com Garlan e Perry em [Garlan; Perry, 1995] a arquitetura de software pode expor as dimensões através das quais um sistema deve evoluir.

Evolução subtende a idéia de mudança essencial que não esta clara no termo manutenção. A evolução sugere novos designs evoluindo de sistemas antigos. Finalmente pode ser argumentado que manutenção e evolução oferecem diferentes perspectivas da natureza da mudança. [Godfrey; German, 2008]. É a inovação e não a preservação, que direciona a mudança de software: um sistema moderno adaptado a novos ambientes evolui de um sistema antigo. [Godfrey; German, 2008]. Em [Tu; Godfrey, 2002] os autores acreditam que um dos desafios na pesquisa em evolução de software é como analisar as mudanças estruturais dos sistemas. Para obter o sucesso neste processo, a evolução da arquitetura deve ser identificada e gerenciada para manter a coerência da arquitetura. [Sadou; Tamzalit; Oussalah, 2005]. Desta forma, enquanto a manutenção degrada a vida e a confiabilidade do software, a evolução planeja mudanças que permitirão um tempo de vida longo ao software. Esta pesquisa estuda uma estratégia que planeja a criação de uma nova versão para possibilitar a sobrevivência do software no mercado. A seguir será apresentado roteiro para gerar plano de evolução do software.

3. Roteiro para avaliação de arquitetura

O roteiro gerado para criação do plano de evolução tem como referência os 9 passos do método de avaliação de arquiteturas – ATAM que foram adaptados no sentido de direcionar a avaliação para tratar questões ligadas à evolução do software. O Quadro 1 apresenta as fases do roteiro e os passos envolvidos em cada uma.

Quadro 1: Etapas do Método ATAM

	ETAPAS	PONTOS IMPORTANTES
APRESEN-TAÇÃO	1 - Apresentar o ATAM 2 - Apresentar os objetivos de negócio 3 - Apresentação da arquitetura	Nesta fase são conhecidas as necessidades do cliente e a realidade do sistema avaliado.
INVESTIGAÇÃO E ANÁLISE	4 - Identificando métodos arquiteturais 5 - Gerar a árvore de utilidade. 6 – Análise dos métodos arquiteturais	Nesta fase os participantes sugerem agrupamentos de cenários. Depois que a votação esta completa, os avaliadores determinam um ponto de corte com até 15 cenários.
TESTES	7 – Brainstorm e priorização de cenários 8 – Análise dos métodos arquiteturais	Nesta fase são levantados os pontos de conflitos. É necessário localizar todos os elementos arquiteturais importantes onde existem múltiplos pontos de sensibilidades.
ENTREGA	9 – Consolidar os resultados	Esse plano é um conjunto de recomendações para reestruturar a arquitetura sob a luz dos resultados da analise.

Na etapa de **definição das diretrizes** (Figura 1) é realizado o processo de avaliação do sistema. A fase de entrega, aonde se encontra o passo 9 (Consolidar resultados) (Quadro 1), resume os resultados que serão utilizados na consolidação das diretrizes.

Figura 1 - Estrutura do roteiro utilizado na pesquisa

A etapa de **consolidação das diretrizes** consiste na análise dos dados e classificação das táticas geradas pelos avaliadores de acordo com os pontos de vista do RM-ODP (*Reference Model - Open Distributed Processing*) [ISO/IEC 10746-3 *apud* Becerra, 1998]. De posse dos resultados da avaliação é realizada uma análise dos resultados no sentido de elaborar o plano de evolução. Informações como analise arquitetural dos cenários prioritários, pontos de sensibilidades e *trade-offs* são consolidados em um documento. Os artefatos resultantes do processo de avaliação através do ATAM são adequados para pautar a estratégia da evolução do software. Neste documento estão: os Requisitos de Qualidade; os Riscos e não riscos; os Pontos de sensibilidades e críticos; os Pontos de Conflitos e o Plano de Monitoramento de Riscos.

As táticas resultantes do processo irão direcionar as ações que devem ser tomadas para evolução do software de acordo com os requisitos colocados como meta. Os *trade-offs* irão indicar quais as perdas e ganhos entre as táticas conflitantes. E por fim os riscos apontam os fatores e variáveis que devem ser observados, monitorados e controlados durante o processo de evolução. Os resultados obtidos formam as decisões arquiteturais que serão analisadas com o objetivo de traçar as diretivas para evolução de cada ponto de vista do RM-ODP, observando o seu nível de abstração que são determinados pelos Pontos de Vista ODP (Empresa, Informação, Computação, Engenharia e Tecnologia), [ISO/IEC 10746-3 *apud* Becerra, 1998].

4. Aplicação Prática

O sistema avaliado tem como contexto a Gestão de Companhias Aéreas, consiste em um sistema integrado para gestão de linhas áreas que contempla todos os setores da companhia. O sistema atende ao mercado atual, mas apresenta algumas características que começa a colocá-lo em desvantagem em relação à concorrência, uma vez que o sistema não foi desenvolvido para web, possui um processo de manutenção lento e com muitos efeitos colaterais. Outro ponto relevante e a falta de documentação arquitetural do sistema.

O objetivo da avaliação é gerar informação que possa guiar a evolução do sistema para o lançamento de uma nova versão do software que atenda as necessidades estratégicas das gerencias e dos atuais clientes no sentido de alcançar a continuidade do software no mercado atual e futuro.

4.1. Consolidação dos Resultados

Para este exemplo de aplicação, as decisões arquiteturais resultantes do processo, foram adequadas. A seguir expõem-se os pontos a comentar sobre os resultados obtidos:

- A aplicação do método começou com 7 cenários de usuários (demanda dos usuários do sistema) e 14 cenários de negócios (visão estratégica do sócio) que foram levantados na identificação dos objetivos do negócio junto ao cliente. Os cenários levantados foram classificados de acordo com os atributos de qualidade.
- Na fase de análise foram detectados que dentre as abordagens arquiteturais existentes atendem totalmente somente 1 cenário, atendem parcialmente a 12 cenários e não atendem a 8 cenários. Esses números comprovam a necessidade da reestruturação da arquitetura para adequar as novas necessidades.
- No passo 7 foi observado que alguns cenários eram semelhantes e poderiam ser agrupados. Foi elaborado um Mapa de Cenários com o objetivo deste mapa de visualizar os cenários semelhantes, facilitando um possível agrupamento. Este passo de agrupamento foi incorporado pelos avaliadores para facilitar a resolução de conflitos que poderiam surgir. Com o agrupamento o número de cenários caiu de 21 para 16.
- Na priorização os principais cenários de um sistema são classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. Cinco cenários foram classificados como alta prioridade.
- No passo seguinte foi realizada a análise arquitetural de cada cenário prioritário e então traçadas as táticas para soluções das questões de cada cenário. Também foram identificados: Ambiente, Estímulo Resposta e Abordagem Arquitetural Existente.

Somente acatar as decisões arquiteturais resultantes da avaliação não irá resolver as questões da evolução, muitas vezes uma decisão arquitetural afeta o atendimento de outro requisito. Em casos de arquiteturas complexas onde estão envolvidos vários sistemas e subsistemas ou em uma linha de produto se faz necessária uma análise mais consistente nas escolhas e nos efeitos que elas provocam nos sistemas envolvidos. Dentro deste trabalho foi possível destacar dois pontos de sensibilidades:

- S.1.2. Estratégias de normalização podem degradar a *performance.*
- S.1.1 O aumento do número de componentes pode degradar a *performance.*

Levando em consideração todos os requisitos da árvore de utilidade e analisando as táticas sugeridas, foram identificados 13 *trade-off* dos quais 5 podem ser considerados pontos de sensibilidades pois afetam requisitos que foram considerados com alta prioridade pelos *stakeholders*. Esses conflitos envolveram cenários das mais diversas classificações tais como: Eficiência x Confiabilidade, Manutenibilidade x Confiabilidade, Funcionalidade e Atributo de negócio x Eficiência, Portabilidade x Eficiência, Manutenibilidade x Portabilidade.

A aplicação do método mostrou sua eficiência quando se trata de evolução do software, pois permite nortear a avaliação a partir da arquitetura atual do software e dos cenários que o requisitante deseja alcançar. A aplicação foi bem sucedida na avaliação da arquitetura atual mesmo quando ela não possui documentação e na questão de traçar estratégias de atuação considerando a opinião e a necessidade de todos os envolvidos (*stakholders*). Em relação às praticas sem método, fica claro a vantagem do uso de um método estruturado que conduza a avaliação. Os resultados são concretos e passiveis de serem utilizados pela equipe do cliente.

4.2 Aspectos do Plano de Ação para Evolução

Os artefatos resultantes do processo de avaliação através do ATAM se mostraram adequados para pautar a estratégia da evolução do software. O plano de evolução terá como base os: Requisitos de Qualidade, Riscos e não riscos, Pontos de sensibilidades e críticos, Pontos de Conflitos e Plano de Monitoramento de Riscos.

As táticas sugeridas direcionam as ações que devem ser tomadas para evolução do software de acordo com os requisitos colocados como meta. Os *trade-offs* indicam quais as perdas e ganhos entre táticas conflitantes. E por fim os riscos apontam os fatores e variáveis que devem ser observados, monitorados e controlados durante o processo de evolução. Estas decisões arquiteturais são analisadas com o objetivo de traçar as diretivas para evolução de cada ponto de vista RM-ODP de acordo com o seu nível de abstração. Desta forma o arquiteto e sua equipe poderão elaborar um planejamento estratégico para o desenvolvimento da nova versão do produto. Estas informações irão compor a estratégia da empresa na elaboração do plano para evolução do software.

Dentro do contexto da aplicação exemplo, as táticas são classificadas de acordo com os pontos de vista RM-ODP. Esta classificação facilitará a ação das diversas competências da empresa, cada uma na sua área específica. No exemplo em questão essa divisão facilitou a separação das tarefas e monitoramento da execução de cada uma pelos seus responsáveis. Os responsáveis pelos pontos de vistas devem traçar uma estratégia para cada tática sugerida, e acompanhar sua evolução. No quadro 2 estão listadas as táticas e algumas estratégias sugeridas pela equipe de avaliadores para o ponto de vista de empresa.

Quadro 2 - Táticas Sugeridas para o ponto de vista empresa.

	Tática Arquitetural	Estratégia
Ponto de Vsta Empresa	-Utilização do processo certificado. -Treinamento da equipe. -Manter checklist de teste dos componentes utilizados ou novos. -Executar teste e homologação da nova funcionalidade. -Realizar controle de versão.	- Elaborado um cronograma que considere como tempo máximo 3 anos para finalizar o projeto. - Analisar cada uma das táticas sugeridas e criar um cronograma de ação de acordo com a ordem de prioridade das atividades, observando as prioridades e relacionamentos do tipo dependência entre as táticas. - São responsáveis pelas ações de gerenciamento acompanhando o cumprimento das decisões tomadas

Na análise dos cenários manutenibilidade mostrou ser um requisito prioritário. Uma das táticas sugeridas para resolver esta questão são o baixo acoplamento e a alta coesão. O primeiro diz respeito à dependência entre classe e método, quando existem no código elementos muito acoplados, surgirão problemas com manutenção do código. A coesão trata da quantidade de tarefas especificas que são realizadas dentro de uma classe, ou seja, sobre um mesmo conceito. Aplicar este conceito dentro de projetos exige um bom conhecimento do negócio e domínio da tecnologia utilizada. Uma sugestão de estratégia é utilizar *design patterns*. O padrão Façade ou Facade ajuda a resolver parte deste problema, existem ainda, alguns *frameworks* não intrusos como *Spring* que auxiliam na redução do acoplamento entre as classes.

5. Conclusão

O roteiro apresentado mostrou-se capaz de expandir as capacidades da arquitetura resultante, diferente dos procedimentos tradicionais centrado somente em ajustes funcionais A utilização do roteiro definido nesta pesquisa permitiu a análise dos pontos

fracos da arquitetura do sistema e o estudo de uma abordagem. Para reduzir os riscos associados com a evolução de software, o método de avaliação baseada em cenários pode ser usado para melhorar a arquitetura de sistema durante todo o ciclo de vida do sistema. A vantagem de usá-lo no início das atividades de reestruturação do sistema ou até no início do desenvolvimento é a possibilidade de descobrir problemas de projeto numa fase que ainda é possível tomar as decisões adequadas.

A aplicação do exemplo deixou algumas lições importantes no uso do método ATAM como base para avaliação de arquiteturas, como: Melhorias de Documentação, Coleta de Cenário e foco da avaliação, Problema descobertos durante todo o ciclo de vida de software, Melhorias para a família de produtos de software. O processo de avaliação descrita não é específico para famílias de produtos de software e pode ser aplicada a qualquer arquitetura de software. No entanto considerando o contexto de uma arquitetura de software de uma família de produtos, a avaliação considera a evolução para o conjunto dos produtos.

6.Referências

Becerra, J. (1998) Aplicabilidade do padrão de processamento distribuído e aberto nos projetos de sistemas de automação. Tese (Doutorado), Universidade de São Paulo, São Paulo.

Chávez, M. (2009) Um Processo para o controle da evolução da Arquitetura de Software Baseado em ODP. 2009. Dissertação (Mestrado) - Universidade de São Paulo. São Paulo.

Clements, P; Kazman R; Klein, M. (2009) Evaluating software architecture: Methods and case studies. SEI, 8 edição.

Garlan. D.; Perry. D. (1995) Introduction to the Special Issue on Software Architecture," IEEE Transactions on Software Engineering, vol. 21, no. 4, p. 269-274, Abril.

Godfrey, M.W.; GERMAN, D.M.; The past, present, and future of software evolution, In: Frontiers of Software Maintenance. p. 129-138; Beijing. 09/2008

Graaf. B.; (2007) Model-Driven Evolution of Software Architectures,"Software Maintenance and Reengineering, European Conference on. v. 0; p. 357-360. Los Alamitos, CA, USA

Kruchten , P.; Obbink, H.; Stanford, J. (2006) The past, present, and future for software architecture. Software, IEEE, v. 23, n.2, p. 22-30.

Sadou, N.; Tamzalit D.; Oussalah M. (2005) A unified Approach for Software Architecture Evolution at different abstraction levels, In: Proceedings of the 2005 Eighth International Workshop on Principles of Software Evolution, p. 65-70, IEEE Computer Society Washington, DC, USA.

Svahnberg. M.; Supporting software architecture evolution. 2003. Tese (doutorado). 2003 Blekinge Institute of Technology - Suécia, 2003.

Tu. Q, Gadfrey. M. W., (2002) "An Integrated Approach for Studying Architectural Evolution", In: 10th International Workshop on Program Comprehension, p.127, Paris, France.