



IV SINGEP

Simposio Internacional de Gestao de Projetos, Inovacao e Sustentabilidade
International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317 - 8302

INOVAÇÃO ARQUITETURAL E CRIAÇÃO DE COMPETÊNCIAS EM PROGRAMAS DE DESENVOLVIMENTO DE SOFTWARES

DANIEL DE LIMA PIRES

UNINOVE – Universidade Nove de Julho
dlpires@uninove.edu.br

MARCOS PAIXÃO GARCEZ

UNINOVE – Universidade Nove de Julho
mpgarcez@gmail.com

Agradeço ao meu orientador e minha família pelo suporte e apoio irrestrito em todos os momentos.



INOVAÇÃO ARQUITETURAL E CRIAÇÃO DE COMPETÊNCIAS EM PROGRAMAS DE DESENVOLVIMENTO DE SOFTWARES

Resumo

O presente relato técnico foi realizado numa empresa brasileira líder no setor de telecomunicações. Decorrente de sua estratégia corporativa, parte dos seus *softwares* foi substituída por novas soluções, ou seja, sofreram melhorias ou foram substituídos. Tal processo de mudança foi orientado pela implantação de projetos integrados de tecnologia da informação em um programa com gestão unificada, caracterizando um caso de inovação arquitetural. Foi realizado um estudo de caso baseado em observação participante e análise documental, visando responder a seguinte questão: como ocorre a integração entre diferentes projetos simultâneos de desenvolvimento de *software* na realização de um programa com gestão unificada? Mais especificamente, foi analisada a atuação e os obstáculos encontrados por uma equipe transversal de parametrização de *software* com o objetivo de entender e descrever a importância da integração entre os projetos. Os resultados encontrados permitem concluir que os esforços de integração entre os diferentes projetos de *software* são fundamentais para o sucesso do programa e que uma visão fragmentada dos projetos deve ser evitada, a fim de mitigar riscos que afetem negativamente a qualidade de suas entregas, devido a complexidade da integração das diferentes equipes de projetos envolvidas.

Palavras-chave: inovação arquitetural, desenvolvimento de competências, competências de equipe, gestão de programas.

Abstract

This technical report was conducted in a leading Brazilian organization in the telecommunications industry. Due to the corporate strategy, a part of the software used in operations was replaced by new software solutions that were improved or replaced. This change was driven by the implementation of integrated information technology projects in a program with unified management, featuring a case of architectural innovation. A case study research based on participant observation and documental analysis was conducted to answer the question: how does the integration between different simultaneous software development projects occur in a program with unified management? Specifically, the parameterization crew actions and obstacles were analysed aiming to understand and describe the importance of integration between projects. It was possible to conclude that the integration efforts between the different software projects are critical to the success of the program and that a fragmented view should be avoided in order to mitigate risks that adversely affect the quality of their deliveries, due to the complexity of integration of a number of different project teams.

Keywords: *architectural innovation, competence development, team competencies, program management.*



1 Introdução

Quando trata-se de operadoras do setor de telecomunicações, seus sistemas podem ser classificados em duas categorias: de suporte às operações ou de suporte ao negócio. Os sistemas de suporte às operações, ou simplesmente OSS (*operations support systems*), são os sistemas e redes de equipamentos que, interconectados, formam a rede de comunicação em si tornando possível que a operadora preste os seus serviços aos clientes. Um exemplo desses sistemas é o sistema de monitoramento de interrupções de sinal (*outage*) entre um elemento e outro da rede. Já os sistemas de suporte ao negócio, chamados BSS (*business support systems*) são os sistemas que permitem que o negócio gire em torno da rede OSS, ou seja, são os sistemas de relacionamento com o cliente e de faturamento por exemplo.

Daqui pra frente, a fim de tornar mais clara a distinção entre termos e não suscitar dúvidas devido à falta de padronização, “*software*” significará uma aplicação de TI individual e “sistema” um conjunto de *softwares* interligados formando uma malha de aplicações interligadas.

Este estudo foi realizado em uma grande empresa brasileira de telecomunicações que tomou a decisão estratégica de modernizar alguns de seus *softwares* de OSS e BSS – por volta de vinte *softwares* – de uma vez, o que foi viabilizado por meio de um programa composto por mais de vinte projetos simultâneos de desenvolvimento ou de customização de *software*, além de projetos transversais de suporte (de arquitetura, de parametrizações, de medições de desempenho, entre outros). A fim de ilustrar a diversidade de usuários afetados pelo novo sistema, são exemplos entre muitos outros possíveis: técnicos de rede, atendentes de *call center*, clientes através da internet, escritório de faturamento.

O cronograma de implantações foi planejado conforme definição de quais cidades deveriam ser atendidas pela operadora em cada implantação, de modo que durante a primeira implantação todo o sistema já estava funcionando e foi implantado para operar somente em uma cidade. A duração do projeto desde o início do planejamento até a implantação da primeira entrega foi de três anos e ocorreu em meados da década de 2010, e é dela que tratará este relato.

1.1 Situação-problema

Como característica primordial dos novos *softwares* implantados, todos foram liderados pela área de TI, e muitos deles funcionam de maneira integrada o que traz o foco do presente trabalho para sua arquitetura de integração. Além disso todos os softwares foram desenvolvidos por empresas terceirizadas, doravante chamadas de fornecedores, sendo que algumas delas desenvolveram mais de um sistema, dependendo de sua especialização.

De maneira a permitir maior flexibilidade para o negócio, foi determinado que algumas funcionalidades dos *softwares* funcionariam a partir de parâmetros, ou seja, são parametrizáveis. Para melhor explicar o conceito de parametrização utilizado segue um exemplo no parágrafo que seguinte.

Um cliente pode escolher o dia de vencimento de sua fatura através do *site* da empresa na internet, desde que esse dia seja um dos dias permitidos: 05, 10, 15 ou 20. Essa lista de valores são parâmetros que devem ser definidos pelos responsáveis de negócio e inseridos no *software* do *site* pela equipe de TI responsável pela parametrização, de maneira que, se inseridos corretamente o *site* funcionará corretamente. Entretanto essa mesma lista de parâmetros deve estar parametrizada no *software* que realiza os cálculos de faturamento, para que no dia correto o cliente receba sua fatura em casa. Caso haja incongruência entre os valores presentes nos dois *softwares*, então o cliente poderá escolher uma data de vencimento



inválida e não receberá a fatura no vencimento escolhido, causando transtornos a ele e danos à imagem da empresa.

A situação-problema que será abordada, é a de realizar a correta parametrização nos *softwares* de maneira integrada. Dependendo da sensibilidade da informação em questão ela poderá alimentar dois ou mais *softwares*. A inconsistência das informações parametrizadas causa atraso nas implantações do programa devido à extensão da fase de testes de qualidade para além do cronograma inicialmente acordado e, em última análise, traz consequências negativas a qualidade do sistema implantado, já que o correto fluxo de comunicação entre seus componentes é de suma importância para a correta prestação de serviços da empresa.

A análise dos resultados foi realizada por meio da observação participante e da análise de documentação secundária produzida no decorrer da intervenção. Nas próximas seções serão detalhados o referencial teórico, a metodologia empregada no estudo, a intervenção utilizada, a análise e os resultados obtidos, as considerações finais e as referências bibliográficas.

2 Referencial teórico

Alguns eixos teóricos e seus conceitos foram selecionados para darem suporte a discussão apresentada nesse relato técnico, e são apresentados nas subseções a seguir.

2.1 Inovação de componentes *versus* de arquitetura

Para Henderson e Clark (1990), a inovação pode ser mensurada devido ao seu impacto em dois eixos de conhecimento: componente *versus* arquitetura, sendo que o nível de mudança do componente mensura a inovação como grau de conhecimento de um componente particular, enquanto que o nível de mudança da arquitetura mensura a inovação como grau de conhecimento da maneira como os componentes são integrados entre si.

Henderson e Clark (1990) mostraram que o conhecimento do nível arquitetural fica incrustado nas estruturas e procedimentos da organização o que torna difícil seu reconhecimento e correção, sendo um desafio sutil que afeta significativamente a competitividade das organizações. Além disso há necessidade de renovar os canais provedores de conhecimento (pessoas e documentos), já que se não forem atualizados não servirão mais para essa finalidade.

		Nível de mudança dos componentes	
		Baixa	Alta
Nível de mudança da arquitetura	Baixa	Incremental	Modular
	Alta	Arquitetural	Radical

Figura 1. Modelo de inovação de Henderson e Clark (1990)

Fonte: Henderson, R. M., & Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35(1), 9–30.



2.2 Competências *versus* aprendizado

Indo ao encontro do raciocínio apresentado na subseção anterior, Drejer e Riis (1999) apontam que a inovação tecnológica gera a necessidade de organizar o trabalho de novas maneiras, o que exige o desenvolvimento de novas competências nas empresas. Um aumento da competência organizacional para realizar uma determinada tarefa, significa realizar essa tarefa de maneira cada vez melhor, e para que isso aconteça é necessário que cada indivíduo envolvido evolua em seu aprendizado a respeito do que precisa ser feito.

Drejer e Riis (1999) apresentam as competências como compostas por tecnologia, pessoas, organização e cultura, e as competências mais difíceis de desenvolver são aquelas que ligam diversos departamentos da organização, onde processos em nível organizacional são necessários para coordenar as diferentes atividades e tecnologias envolvidas. Nesse caso a ação combinada dos conhecimentos tácitos de vários indivíduos geram um conhecimento maior do que a soma individual dos conhecimentos.

2.3 Complexidade

Para Baccarini (1996) o conceito de projeto ou sistema complexo está associado à existência de vários componentes inter-relacionados, ou seja, a singularidade ou diferenciação de cada um e a interdependência entre eles. Uma exigência fundamental para a complexidade é a integração das partes a partir de coordenação, comunicação e controle.

2.4 Riscos em projetos de software

Wallace, Keil e Rai (2004) realizaram um estudo para entendimento dos riscos que atuam nos projetos de software e seus impactos. Apesar de outras definições de riscos, os riscos aqui considerados influenciam o ambiente contribuindo para a geração de resultados negativos nos projetos. Dos resultados encontrados, os riscos relativos a complexidade afetam principalmente os projetos estratégicos das organizações, enquanto que projetos terceirizados apresentam grande incidência de riscos ligados a equipe.

Complexidade para Wallace *et al.* (2004) possui o sentido de serem empreendidos com inovação tecnológica, de existirem ligações com outros *softwares*, de possuírem diversos fornecedores e por serem extensos se comparados com demais projetos da organização. Entre os fatores de riscos de equipe, o conhecimento técnico dos seus integrantes e suas habilidades de comunicação são exemplos que afetam principalmente os projetos terceirizados, pois envolvem compartilhamento de conhecimento entre duas ou mais organizações.

2.5 Comunicação em programas

Visando compreender como a comunicação ocorre em programas, foram analisados por Turkulainen, Ruuska, Brady e Artto (2015) duas *interfaces* de comunicação: projeto-para-projeto e projeto-para-organização. O primeiro tipo de comunicação é o que está sendo observado neste relato e é o mais complexo, já que os projetos são empreendimentos diferentes entre si, o que exige grande esforço de integração. Nele três mecanismos diferentes de comunicação são largamente utilizados e são citados e exemplificados brevemente a seguir: o impessoal, que é conseguido através da definição de regras, padrões, manuais e procedimentos; o pessoal, alcançado através de funções integradoras e de ligação que atravessam as fronteiras dos projetos individuais; e o de grupo, que acontece ao serem estabelecidas equipes transversais ou forças-tarefa.



3 Intervenção e mecanismos adotados

A fim de concentrar o conhecimento necessário para integração dos parâmetros das diferentes soluções de *software*, foi criada uma equipe transversal de parametrização com foco de atuação em todo o sistema. As equipes de cada projeto de *software* eram diferentes; cada uma delas tratava do desenvolvimento de um *software* diferente dos demais, e cada *software* estava sob a responsabilidade de um fornecedor.

Para gerar as competências necessárias para a equipe transversal de parametrização, alguns treinamentos com foco em transferência de conhecimento foram realizados com as empresas fornecedoras desenvolvedoras dos *softwares*. Os treinamentos tiveram alguns formatos diferentes, dependendo da quantidade de parametrizações e da complexidade do *software* em questão. Foram fornecidos manuais de parametrização dos *softwares* em alguns casos, em outros casos foram realizadas forças-tarefa onde pessoas designadas pelo fornecedor passaram o conhecimento através de reuniões de curta-duração para a equipe de parametrização, e em outros casos a equipe de parametrização recebeu treinamento em local próprio para esse fim inclusive com a realização de exercícios práticos.

4 Metodologia

Segundo Godoy (1995) a pesquisa qualitativa parte de um foco amplo de pesquisa que vai se afinando a medida que o estudo se desenvolve, é privilegiada pelo contato direto e duradouro do pesquisador com o fenômeno sendo estudado. A pesquisa deve se desenrolar, quando possível, contemporaneamente ao fenômeno e em seu ambiente natural, ou seja, onde as variáveis envolvidas no estudo não podem ser controladas pelo pesquisador.

Para a coleta de dados, podem ser usados por exemplo o trabalho de campo, documentos arquivados, entrevistas, observações ou qualquer combinação desses (Yin, 1981). Nesse estudo foram utilizados a observação participante e análises de documentos secundários como meios de coleta.

Para Abib, Hoppen, & Hayashi Junior (2013) a observação participante proporciona ao pesquisador ricas percepções dos aspectos individuais, sociais e organizacionais, contudo o pesquisador deve cuidar para não enviesar suas observações com aspectos emocionais e também para não perder sua objetividade.

No presente trabalho, um dos pesquisadores atuou como coordenador da equipe de parametrização em foco durante um ano, desde a sua criação até o momento presente da escrita desse relato, e foi dessa experiência que se originaram os dados de observação participante e de análise documental coletados. O pesquisador no seu papel de coordenador de equipe se reportava para um gerente e quatro analistas se reportavam ao pesquisador. Durante esse período, foram realizadas conversas informais e reuniões de trabalho que captaram as percepções dos analistas, do gerente da equipe e de demais envolvidos com a equipe de parametrização no ambiente natural do fenômeno sendo analisado: foi observado o dia-a-dia de trabalho em toda a sua naturalidade, ou seja, foi observado o aparecimento de problemas e as soluções encontradas, os treinamentos realizados, os processos de trabalho e documentações que foram criados, as reuniões de equipe e reuniões com membros de outras equipes, contudo não foi possível a realização de gravações para que fosse mantida a descrição do pesquisador.



5 Resultados obtidos e análise

Conforme já descrito anteriormente, a interligação de *softwares* novos e existentes baseando-se em uma nova arquitetura era parte da estratégia da organização, que possuía a finalidade de proporcionar maior flexibilidade para atendimento das demandas dos seus clientes. As equipes de projeto desses *softwares* eram diferentes entre si, ou seja, cada uma delas tratava do desenvolvimento de um *software*, e além disso eram realizados por empresas fornecedoras distintas, então não havia na fase de levantamento de requisitos dos projetos, pessoas ou equipes com função de integrar o conhecimento relativo a parametrização dos *softwares*.

O escopo de atuação da equipe de parametrização criada foi o de parametrizar os *softwares* e consequentemente integrar os projetos em torno desse objetivo. Segundo Turkulainen *et al.* (2015), pessoas de papel integrador tem o propósito de facilitar o compartilhamento de informações comuns e conhecimento entre os projetos. A criação dessa equipe se deu um pouco tardiamente, ou seja, com os projetos de desenvolvimento já em execução.

Devido a essa fraca integração inicial, algumas áreas de intersecção entre *softwares* eram desconhecidas pela equipe de parametrizações e também pelas equipes de desenvolvimento dos fornecedores, o que ocasionaram algumas idiossincrasias, conforme os dois exemplos a seguir. Primeiro exemplo: uma informação parametrizável em um *software* era replicada para outro *software*, porém não era parametrizável nesse segundo, ou seja, a alteração de um parâmetro em um sistema significava uma mudança de escopo no outro. Segundo exemplo: uma informação parametrizável foi classificada como sensível pela equipe de desenvolvimento de um *software* e sua alteração estava protegida por senha, enquanto que em outro *software* a alteração dessa mesma informação era facilmente realizada por qualquer usuário. Em ambos os casos, a alteração de uma informação parametrizável em um *software* causaria o mal funcionamento de outro *software*, e os efeitos negativos trazidos por essas duas incongruências poderiam ter sido evitados caso a parametrização do sistema como um todo fosse vista como integrada desde a fase de definição dos requisitos dos *softwares*.

Segundo a definição de Baccarini (1996), como esse programa possui características complexas, então a integração é um dos seus alicerces. Wallace *et al.* (2004) observam ainda que os riscos ligados a complexidade são os maiores ofensores em projetos estratégicos. Portanto o papel integrador dessa equipe e o caráter complexo do programa conferem a ela fundamental importância para que o programa seja bem sucedido.

Drejer & Riis (1999) descreveram que as competências da organização são habilitadas pelo desenvolvimento das suas competências internas. O sistema em questão envolve diferentes *softwares* utilizados por diferentes departamentos da empresa em diferentes unidades organizacionais, então segundo Drejer & Riis (1999) a capacidade de aprendizado organizacional é um fator fundamental para o desenvolvimento das diferentes competências.

A fim de gerar as competências de trabalho necessárias para a atuação da equipe de parametrização, o fornecedor com o maior número de *softwares* do programa, responsável pelo desenvolvimento de seis aplicações, ofereceu treinamento para a equipe de parametrização em seus produtos de *software*. Tal treinamento foi pautado na parametrização individual e também integrada de seus produtos, e também envolveu a sugestão de um processo adequado para a realização das parametrizações, o qual era composto das seguintes atividades sequenciais: 1 - fornecimento das informações pelo pessoal de negócio responsável, 2 - transformação dessas informações em dados sistêmicos de parametrização, 3 - implementação da parametrização nos *softwares*. Além disso para cada uma das informações que deveriam ser parametrizadas no sistema, um *software* foi denominado “dono” da informação enquanto que os demais deveriam seguir o que estava parametrizado no primeiro. No nosso exemplo, citado no item 1.1 desse relato, o dono da informação correspondente aos



dias de vencimento possíveis de serem escolhidos é o *software* de faturamento enquanto que o *site* é afetado por essa informação e deve seguir a mesma informação parametrizada no outro. Tal fluxo foi apoiado pelo versionamento e compartilhamento manual de arquivos, sempre coordenado pela equipe de parametrização. Esse processo serviu como base para a criação dos processos de parametrização adotados em todo o programa, ou seja, entre todos os projetos que continham informações parametrizáveis entre seus requisitos de *software*.

Mesmo com um processo bem definido uma grande dificuldade foi constatada para integrar o conhecimento de fornecedores diferentes, pois as informações parametrizáveis e integradas não estavam mapeadas previamente por nenhuma das equipes envolvidas com as integrações. Uma vez que os treinamentos dados pelos fornecedores foi realizado em nível técnico e que não foram abordados aspectos de negócio, foram identificadas lacunas de conhecimento na equipe de parametrização, pois o significado e a relevância de cada informação para o negócio era desconhecido.

A falta desse conhecimento demandou grande esforço para mapeamento das conexões entre *softwares*, através da constante comunicação entre a equipe de parametrizações e os fornecedores. Foram observados todos os tipos de comunicação entre os projetos conforme descreveram Turkulainen *et al.* (2015): impessoal, pessoal e de grupo. Aqui fica evidente importância da característica transversal dessa equipe de parametrização, e ainda é possível corroborar que projetos terceirizados de *software* tem alto impacto nos riscos ligados a equipe, tanto no que concerne às habilidades técnicas quanto aos maiores obstáculos de comunicação (Wallace *et al.*, 2004).

Com a finalidade de compartilhar internamente o conhecimento adquirido através de manuais de procedimentos ou de reuniões de força-tarefa, a equipe de parametrizações procurou produzir documentos que beneficiariam toda a equipe. Entretanto foi percebida a dificuldade de sistematizar esse compartilhamento, devido à falta do domínio da informação sendo aprendida e aos seus diferentes entornos em cada um dos sistemas envolvidos, bem como à falta do conhecimento em técnicas de compartilhamento de conhecimento, que não serão abordadas neste relato. O componente cultural das competências organizacionais (Drejer & Riis, 1999) foi percebido como fundamental para o desenvolvimento da organização já que o compartilhar conhecimento exige disciplina e comprometimento. De modo conforme Henderson e Clark (1990) afirmam que o conhecimento arquitetural exige tempo até que seja dominado pela organização e é gerado através de canais tanto formais como informais de comunicação - é como se uma nova forma de organizar o trabalho emergisse em torno das principais competências exigidas pela nova arquitetura.

6 Considerações finais

Foi buscada a descrição das dificuldades encontradas pela equipe de parametrizações de *softwares* integrados e seus esforços para aquisição e compartilhamento do conhecimento necessários para parametrizar o sistema como um todo. O método de pesquisa utilizado proporcionou convívio diário com a equipe em questão, o que tornou possível a partir da observação a identificação de pontos fortes e fracos do programa observado.

A formação de uma equipe exclusiva para essa finalidade foi entendida como uma necessidade imposta pela integração, entretanto um fator complicador para o sucesso da integração é a terceirização, já que cada fornecedor de um *software* possui integração limitada com o fornecedor de outro *software*. Dessa forma sua atuação é definida como um meio de integração de conhecimento. Todavia o conhecimento da equipe deve estar pulverizado entre todas as soluções integradas, o que gerou a necessidade de transferência de conhecimento entre os fornecedores e a organização contratante.



O conhecimento do negócio pela equipe de parametrização deverá ser tratado com prioridade, e deverá ser obtido com participação ativa na fase de definição de requisitos, a fim de possibilitar um conhecimento de cobertura suficiente para suprir as áreas de integração entre os sistemas. Além disso um meio sistemático e de eficácia comprovada para compartilhamento de conhecimento deve ser usado a fim de viabilizar o desenvolvimento de suas novas competências.

A primeira fase do programa foi completada e novas implantações estão programadas, dessa forma erros cometidos anteriormente e descritos nesse relato poderão contribuir para o sucesso das fases posteriores, bem como contribuir para práticas de inovação arquitetural em sistemas de *software*.

Como limitações desse estudo é possível citar a avaliação apenas da equipe de parametrização, quando um estudo também das dificuldades dos fornecedores envolvidos poderia mostrar novas perspectivas do fenômeno. Para futuras pesquisas fica a sugestão do estudo de outros aspectos de integração de um programa com as características desse aqui apresentado, a fim de entendê-lo com maior profundidade.

7 Referências

- Abib, G., Hoppen, N., & Hayashi Junior, P. (2013). Observação participante em estudos de administração da informação no Brasil. *Revista de Administração de Empresas*, 53(6), 604–616. <http://doi.org/10.1590/S0034-759020130608>
- Baccarini, D. (1996). The concept of project complexity - a review. *International Journal of Project Management*, 14(4), 201–204.
- Biancolino, C. A., Kniess, C. T., Maccari, E. A., & Rabechini Jr., R. (2012). Protocolo para Elaboração de Relatos de Produção Técnica. *Revista Gestão e Projetos*, 3(2), 294-307.
- Drejer, A., & Riis, J. O. (1999). Competence development and technology: How learning and technology can be meaningfully integrated. *Technovation*, 19(10), 631–644.
- Godoy, A. S. (1995). Introdução à pesquisa qualitativa e suas possibilidades, 35(2), 57–63.
- Henderson, R. M., & Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35(1), 9–30.
- Turkulainen, V., Ruuska, I., Brady, T., & Artto, K. (2015). Managing project-to-project and project-to-organization interfaces in programs: Organizational integration in a global operations expansion program. *International Journal of Project Management*, 33(4), 816–827. <http://doi.org/10.1016/j.ijproman.2014.10.008>
- Wallace, L., Keil, M., & Rai, A. (2004). Understanding software project risk: a cluster analysis. *Information & Management*, 42(1), 115–125. <http://doi.org/10.1016/j.im.2003.12.007>
- Yin, R. K. (1981). The Case Study Crisis: Some Answers. *Administrative Science Quarterly*, 26(1), 58. <http://doi.org/10.2307/2392599>