

Migração de Sistemas Monolíticos para Microserviços

Guilherme L. D. Villaca¹

¹Universidade Estadual do Oeste do Paraná (Unioeste)
Cascavel – PR – Brazil

guidvillaca@gmail.com

Abstract. *Migrating legacy systems through microservices has been a tendency to solve old problems faced during software development, such as scalability and maintenance. However, it is not simple to identify when these systems should migrate and what the consequences are. Unioeste Information Technology Center (NTI) has faced these problems in recent years, systems are not easily scalable and are difficult to maintain ...*

Resumo. *Migrar sistemas legados por microserviços tem sido uma tendência para resolver velhos problemas enfrentados durante o desenvolvimento de software, como escalabilidade e manutenção. Porém não é simples identificar quando estes sistemas devem migrar, e quais as consequências disso. O Núcleo de Tecnologia da Informação(NTI) da Unioeste tem enfrentado estes problemas nos últimos anos, os sistemas não são facilmente escaláveis e são de difícil manutenção...*

1. Introdução

A tecnologia de software baseado em reuso é um processo de design de software que pode resultar em redução de tempo e custos de desenvolvimento além de aumentar a flexibilidade, manutenibilidade e confiabilidade do software. A principal razão para pensar em reuso é evitar trabalhos repetidos no desenvolvimento de software e usar do conhecimento e experiência adquiridos previamente e concentrar os esforços em partes mais críticas da aplicação. Seu foco é não necessitar mais desenvolver um software do zero [Yang et al.]. Reengenharia é uma forma para atingir o reuso de software e para entender os conceitos ocultos no domínio da aplicação. Seu objetivo é obter e manter o conhecimento incluído nos sistemas legados, usando-o como base para a continuidade e evolução estruturada do sistema. O código legado tem lógicas programadas, decisões de projeto, requisitos de usuários e regras de negócio que podem ser recuperadas e reconstruídas sem perder a semântica [Garcia et al. 2004]

O desenvolvimento de sistemas pelas organizações segue um padrão, onde o desenvolvimento começa pela solução específica de um problema e a partir daí novas soluções são adicionadas, formando um ou mais sistemas maiores e complexos. Com o passar dos anos esses sistemas começam a gerar problemas para a equipe de desenvolvedores. Podemos citar:

Manutenibilidade: Que é a capacidade de um produto de software ser modificado. Se um software não for construído com capacidade para lidar com a demanda de constantes mudanças, certamente irá consumir mais recursos e tornar a manutenção uma tarefa tediosa [Velmourougan et al. 2014];

Escalabilidade: É a capacidade de um software para acomodar crescimento em termos de

tamanho e complexidade do problema e em um segundo aspecto está relacionado a uma medida de eficácia quando usada em um contexto maior em escopo e complexidade do que o contexto para o qual foi originalmente projetado. [Ibrahim et al. 2009]

Confiabilidade: Capacidade do sistema estar livre de falhas [Pan 1999]. Não importa o tamanho do erro ou o quanto ele afeta o sistema, corrigir um erro de forma equivocada pode gerar novos problemas.

Qualidade

Documentação pobre ou mal feita: pode gerar problemas futuros, em relação a origem ou rastreabilidade de um requisito, por que, quando como um requisito surgiu.

O jeito tradicional de desenvolvimento resulta em sistemas monolíticos que é um sistema composto por módulos que não são independentes da aplicação a que pertencem [Dragoni et al.]

Algumas abordagens foram propostas para resolver estes problemas clássicos dos sistemas monolíticos, como o SOA (Service Oriented Architecture) que foi algo surgido na academia, porém sua implementação não era tão simples e acabou não sendo adotado pela indústria. Alguns anos atrás surgiu algo similar ao SOA, os microserviços porém dessa vez proposto pela indústria

Este cenário com estes problemas relatados podem ser resolvidos de diversas maneiras, neste trabalho iremos propor uma solução baseada em microserviços, a migração de um sistema monolítico tem se tornado bastante comum na indústria, segundo [Carvalho et al.] alguns desenvolvedores até atribuem a mudança pelo hype ou seja, por esta abordagem estar na moda e não querer ficar pra trás.

O objetivo deste trabalho é analisar os sistemas desenvolvidos pelo Núcleo de Tecnologia da Informação (NTI) da Unioeste e verificar como pode ser resolvido os problemas enfrentados pelo setor. Atualmente no NTI existem mais de 30 sistemas em produção, destes em torno de 28 são sistemas WEB,

2. Desenvolvimento

Desenvolver

Titulo Identificação autor e filiação Resumo palavras chave Introdução Desenvolvimento (uma ou mais seções, incluindo revisão bibliográfica, trabalhos relacionados e metodologia) resultados esperados cronograma de execução

3. Resultados Esperados

Resultados

4. Cronograma de Execução

Cronograma

Referências

Carvalho, L., Garcia, A., and Assunção, W. K. G. Extraction of Configurable and Reusable Microservices from Legacy Systems An Exploratory Study.pdf.

Dragoni, N., Giallorenzo, S., Lafuente, A. L., and Mazzara, M. Microservices : yesterday , today , and tomorrow. pages 1–17.

- Garcia, V. C., Lucrédio, D., Do Prado, A. F., De Almeida, E. S., and Alvaro, A. (2004). Using reengineering and aspect-based techniques to retrieve knowledge embedded in object-oriented legacy system. *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI-2004*, pages 30–35.
- Ibrahim, H., Far, B. H., and Eberlein, A. (2009). Scalability Improvement in Software Evaluation Methodologies. *2009 IEEE International Conference on Information Reuse & Integration*, pages 236–241.
- Pan, J. (1999). Software Reliability. pages 1–9.
- Velmourougan, S., Dhavachelvan, P., Baskaran, R., and Ravikumar, B. (2014). Software development Life cycle model to improve maintainability of software applications. *2014 Fourth International Conference on Advances in Computing and Communications*, pages 270–273.
- Yang, X., Chen, L., Wang, X., and Cristoforo, J. A Dual-Spiral Reengineering Model for Legacy System.