

A importância do gerenciamento de configuração para o ciclo de vida do software: um estudo de caso baseado nas diretrizes da engenharia de software

The importance of configuration management for the software life cycle: a case study based on software engineering guidelines

Mariana Magalhães Malhone¹
Mônica Frigeri²

Resumo: *Devido à cobrança de uma boa qualidade e funcionalidade de um software, as empresas vêm procurando formas de organizar e trabalhar uma qualidade por meio da gestão de qualidade e gestão de projetos. Porém, dentro desses dois ambientes, há diferentes tipos de processos a serem seguidos e há casos em que é preciso criar funções de suporte para o gerenciamento. Tratando-se da qualidade do software, desde seu desenvolvimento até a entrega ao cliente final, o Gerenciamento de Configuração de Software mostra-se como um meio de gestão essencial para a identificação e controle de mudanças no ciclo de vida do software. Sendo assim, este artigo tem como objetivo analisar a importância do gerenciamento de configuração para o ciclo de vida do software, a partir da aplicação de um estudo de caso, com base nas diretrizes da Engenharia de Software. Com o desenvolvimento da pesquisa, é possível verificar como o processo de Gerenciamento de Configuração de Software é aplicado em uma empresa, apresentando as vantagens da aplicação do processo e as desvantagens da não aplicação do mesmo.*

Palavras-Chave: *Gerenciamento de Configuração; Software; Qualidade; Projeto.*

Abstract. *Due to the large collection of good software quality and functionality, companies have been looking for ways to organize and work quality through quality management and project management. However, within these two environments there are different types of processes to follow and there are cases where you need to create functions support for management. Regarding the quality of the software, from its development to delivery to the end customer, Software Configuration Management is an essential management tool for identifying and controlling changes in the software life cycle. Therefore, this article aims to analyze the importance of configuration management for the software life cycle, with the application of a case study and based on the Software Engineering guideline. With the development of the research, it is possible to verify how the Process of Software Configuration Management is applied in a company, presenting the advantages of the application of the process and the disadvantages of not applying it.*

Keywords: *Configuration Management; Software; Quality; Project.*

¹ Faculdade de Tecnologia de Campinas, Campinas (SP). E-mail: magmalhone@gmail.com

² Faculdade de Tecnologia de Campinas, Campinas (SP). E-mail: monica.frigeri@fatec.sp.gov.br

1. Introdução

A necessidade de uma Gestão de Qualidade (GQ) adequada atrelada à Gestão de Projetos (GP) vem crescendo em grande escala nas empresas de desenvolvimento de software. A manutenção do ciclo de vida do software em período de desenvolvimento, a preocupação de relatar todos os problemas envolvidos, os artefatos modificados e a manutenção de um histórico dessas mudanças demonstram uma preocupação com a qualidade envolvida no produto.

Durante o período de desenvolvimento de um software, diversos fatores podem levar à necessidade de alterações, desde *bugs*³ encontrados em testes de qualidade de software, novos requisitos solicitados para o produto e até a modificação de dados. A partir disso entra em cena o Gerenciamento de Configuração de Software (GCS), com a responsabilidade de identificar, organizar, controlar essas modificações e minimizar esses erros.

Segundo Pressman e Maxin (2016, p.624), a Primeira Lei de Engenharia de Sistemas [Ber80] destaca que “não importa onde você esteja no ciclo de vida do sistema, o sistema mudará, e o desejo de alterá-lo persistirá por todo o ciclo de vida”, porém muitas empresas ainda não conseguem mensurar tamanha importância da inserção do Gerenciamento de Configuração de Software em seu processo de GQ e GP. Com isso, não ficam documentadas todas as mudanças realizadas e versões antigas podem acabar ficando indisponíveis.

O processo de Gestão de Configuração de Software mostra-se, além de uma preocupação do Projeto de garantir uma base de dados e rastreamento do produto, uma garantia da qualidade do software aplicável em todo o processo do software. Estes processos citados são tomados “quando um projeto de engenharia de software começa e terminados quando o software sai de operação” (PRESSMAN; MAXIN, 2016, p.624).

2. Gerenciamento de projetos de software

O gerenciamento de projetos de software é uma parte essencial da engenharia de software, pois os projetos precisam ser gerenciados devido aos orçamentos organizacionais e cronogramas a serem seguidos. O papel do gerente de projetos, neste contexto, é garantir que o projeto atenda essas questões e entregue um software de qualidade. Segundo Sommerville (2011, p. 414), as metas para alcançar o sucesso são, respectivamente:

- 1) Fornecer o software ao cliente no prazo estabelecido;
- 2) Manter os custos gerais dentro do orçamento;
- 3) Entregar o software que atenda às expectativas do cliente;
- 4) Manter uma equipe de desenvolvimento que trabalhe bem e feliz.

³ Bugs – Defeito, falha ou erro no código de um programa que provoca seu mau funcionamento.

Tal planejamento mantém-se em um nível mínimo, onde a equipe escolhe sua estratégia, limitada aos requisitos do projeto e padrões organizacionais. Um exemplo de metodologia ágil muito utilizada hoje é a metodologia Scrum.

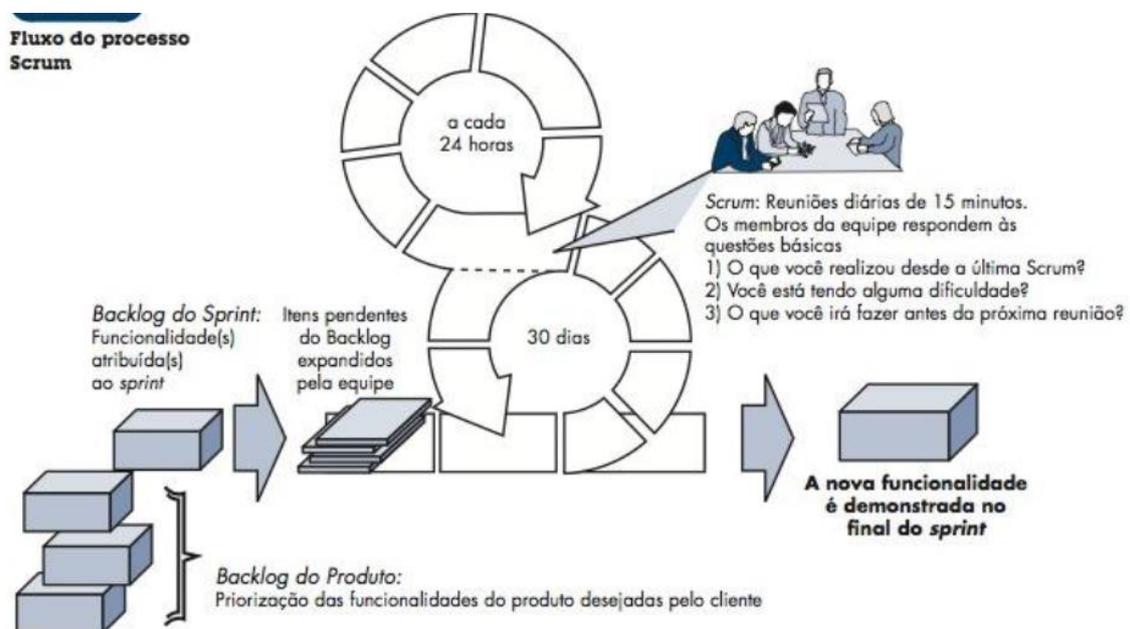
De acordo com Pressman e Maxin (2016, p. 78), “Scrum é um método de desenvolvimento ágil de software concebido por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 1990.”. Os princípios do Scrum são utilizados para orientar as atividades de desenvolvimento e incorporam as seguintes atividades metodológicas: requisitos, análise, projeto, evolução e entrega.

O Scrum pode ser dividido em três fases:

- 1) Planejamento geral, onde são estabelecidos os objetivos gerais e a arquitetura de software;
- 2) Ciclos de *sprint*, onde cada ciclo desenvolve um incremento do sistema;
- 3) E a última fase é responsável por finalizar a documentação exigida (manuais do usuário, definições do sistema) e são avaliadas as lições aprendidas com o ciclo.

Em cada atividade, as tarefas são realizadas dentro de um padrão de processo, denominado *sprint*. O trabalho em si é adaptado e definido de acordo com o problema em questão e é modificado em tempo real pela equipe Scrum. A Figura 1 representa esse fluxo:

Figura 1 - Fluxo do processo Scrum



Fonte: Pressman e Maxin (2016)

A metodologia Scrum enfatiza a importância do uso de um conjunto de padrões de processos de software, pois provam-se eficazes em projetos com entregas em um período curto de tempo, requisitos mutáveis e urgência no negócio.

2.1. Garantia de qualidade do software

A Garantia de Qualidade de Software (SQA, Software Quality Assurance), é uma atividade universal aplicada em todo o processo de software e abrange os seguintes parâmetros: um processo de SQA; tarefas específicas de garantia e controle da qualidade, prática efetiva de engenharia de software; controle dos artefatos e mudanças no software; garantia da conformidade com os padrões de software e mecanismos de medição e geração de relatórios. A Garantia de Qualidade de Software possui atividades que concentram na gestão de qualidade de software, sendo elas:

- 1) Padrões, como ISO e CMMI;
- 2) Revisões e auditorias;
- 3) Testes de software;
- 4) Coleta e análise de erros/defeitos;
- 5) Gerenciamento de mudanças;
- 6) Educação, como forma de melhorar suas práticas de engenharia de software;
- 7) Gerência dos fornecedores;
- 8) Administração da segurança;
- 9) Proteção, redução de riscos em defeitos ocultos;
- 10) Gestão de riscos.

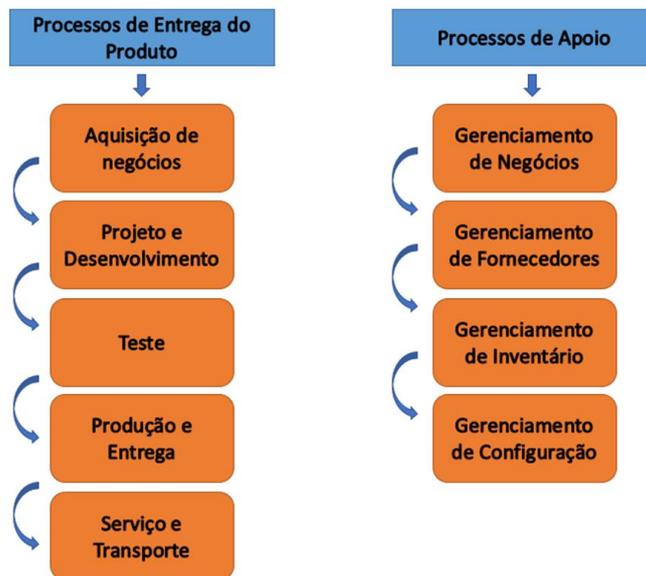
Além disso, a Garantia da Qualidade de Software trabalha para garantir que atividades como manutenção, suporte online, documentação e manuais sejam produzidas de acordo com a qualidade exigida.

2.1.1 ISO 9000/9001 e CMMI

Os sistemas de garantia de qualidade são criados com o objetivo de ajudar as organizações a alcançar que seus produtos e serviços satisfaçam às expectativas do cliente por meio do atendimento às suas especificações (PRESSMAN e MAXIN, 2016). Esses sistemas cobrem diversas atividades, como o ciclo de vida do produto, planejamento, controle, medições, testes e geração de relatórios. A ISO 9000 descreve tais métricas de garantia de qualidade e pode ser aplicada em qualquer organização, independente do ramo.

As empresas, além de estabelecerem as políticas e procedimentos, devem documentar como seus processos funcionam, definindo e mantendo registros que comprovem que os processos estão sendo seguidos. Além disso, o manual de qualidade da organização deve descrever e coletar dados de processos relevantes. A Figura 2 exemplifica os processos essenciais da ISO 9001.

Figura 2 – Os processos essenciais da ISO 9001



Fonte: Sommerville (2011), adaptado

O CMMI (*Capability Maturity Model Integration*), criado pelo Software Engineering Institute, é um modelo de gerenciamento onde o enfoque é voltado para a maturidade de processos de software. Como trata-se de maturidade, é necessário que tenha um conhecimento sobre o nível de qualidade que o processo atinge em um resultado esperado pela organização. Na Figura 3, é possível visualizar os níveis de maturidade:

Figura 3 – Os cinco níveis de maturidade do CMMI



Fonte: GROFFE, Renato José (2019)

Dentro do processo do CMMI, podemos ressaltar as principais chaves de melhoria que o modelo propõe, sendo elas: melhoria do desempenho dos negócios; alavancagem das melhores práticas atuais; construção da resiliência e escala ágeis, focadas na metodologia SCRUM; aumento do valor do benchmarking; aceleração da adoção do modelo na organização (CMMI Institute, 2019).

O CMMI possui soluções para as áreas de desenvolvimento, serviços e gerenciamento. No desenvolvimento, especificamente, devido a criação de soluções complexas, o ciclo de vida do desenvolvimento de software está se tornando cada dia mais difícil de gerenciar, causando custos a mais para as organizações e atrasos em entregas para o cliente. Com isso, o CMMI utiliza do Gerenciamento de Configuração de Software como uma *KPI*⁴ (*Key Performance Indicator*) em seu processo para garantir práticas que garantem o desenvolvimento do software com qualidade e entregas de sucesso para os clientes.

2.2. Gerenciamento de configuração de software

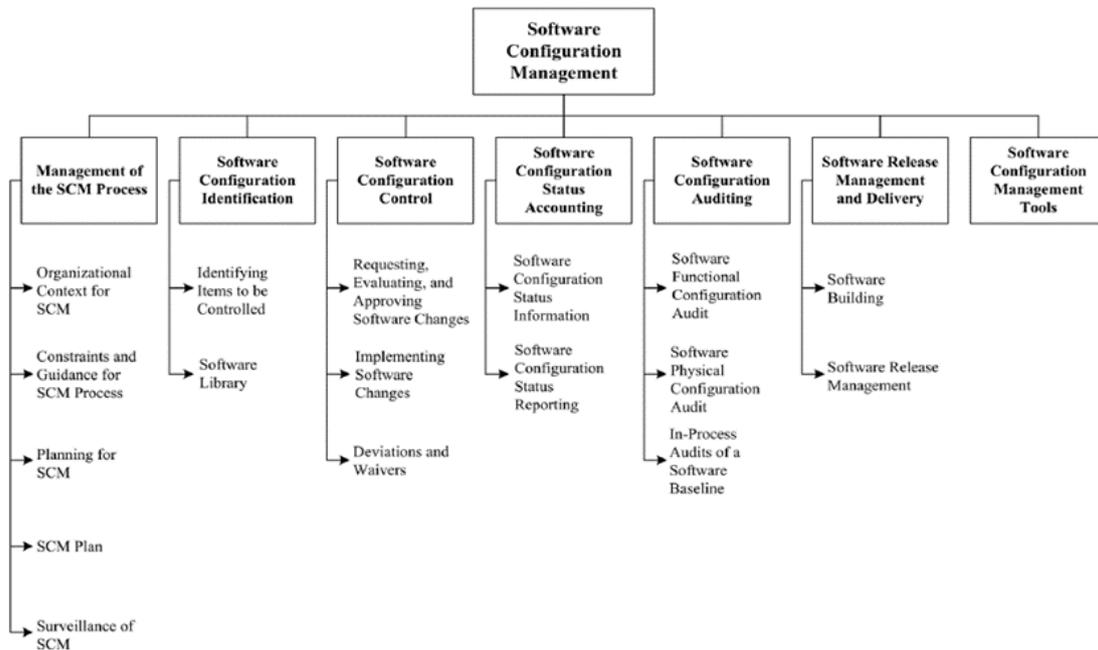
Os sistemas de software estão em constante evolução e necessitam de modificações ao longo do ciclo de vida. Estima-se que tais modificações chegam a consumir cerca de 75% do custo total de sua produção. 20% dos esforços são voltados para a correção de erros e 80% são utilizados para a manutenção e adaptação das funções do software. O Gerenciamento de Configuração surgiu da necessidade de controlar essas mudanças por meio de métodos e ferramentas, maximizando a produtividade do desenvolvimento da solução e minimizando erros durante sua evolução (DANTAS, 2018).

O Gerenciamento de Configuração de Software pode ser visto sob duas perspectivas: na perspectiva gerencial e na perspectiva do desenvolvimento. Na perspectiva gerencial, o GCS possui quatro atribuições, sendo elas: identificação; gerenciamento de alterações; facilitação em versionamento; qualidade do produto. Já na perspectiva do desenvolvimento, engloba três sistemas: controle de modificações, controle de versões e gerenciamento de construção.

Tal metodologia é considerada como um suporte que beneficia a Gestão de Projetos, desenvolvimento, atividades de manutenção, garantia de qualidade e os clientes usuários da solução. A Figura 4 esquematiza o que o GCS abrange e suporta como um todo.

O planejamento do processo do GCS deve ser consistente e estar de acordo com o contexto organizacional, restrições, orientações de acordo com as diretivas da organização e a natureza do projeto, como a criticidade da segurança e a segurança como um todo. Além disso, é de suma importância que sejam bem identificadas as ferramentas e a equipe envolvida com o Gerenciamento de Configuração de Software: como as ferramentas irão comunicar-se uma com as outras e como os membros da equipe (quando houver) irão gerenciar suas atividades comumente, aplicando o gerenciamento de acordo com o planejamento realizado pelo Gerente de Projetos.

⁴ Indicador-chave de desempenho. Por meio de ferramentas de gestão, é possível mensurar o desempenho do processo/organização.

Figura 4 – Divisão de Tópicos de Gerenciamento de Configuração de Software


Fonte: Bourque e Fairley (2014)

3. Metodologia aplicada

A partir da revisão bibliográfica, foi apresentado um estudo de caso sobre uma empresa de Telecomunicações, visando analisar suas variáveis e demonstrar em um ambiente real de trabalho do ramo de projetos em TI quais são seus processos de entregas – seja ele por *sprint*⁵ ou manutenção contínua – enfatizando seus principais diferenciais e seu impacto direto na qualidade do produto.

A nível organizacional, é verificado como a estratégia empresarial afeta as operações e a função discutida, e com enfoque holístico, busca examinar a natureza do programa/organização. Por fim, tratando-se de uma abordagem qualitativa descritiva, tal pesquisa tem o ambiente natural (empresa) como fonte direta de dados e envolve a obtenção de dados descritivos e interativos pelo contato com o pesquisador e situação estudada, caracterizando também uma visão empírica.

3.1 Dados coletados

O Gerenciamento de Configuração de Software na empresa estudada é uma posição estratégica para a organização. Como empresa de grande porte e que lida com diferentes níveis de desenvolvimento de software e de entrega de funcionalidades para os clientes, possuindo um portfólio abrangente, a posição foi consolidada como essencial para o controle e gerenciamento de versionamento dos projetos.

⁵ *Sprint* – Ciclo de trabalho no Scrum, em que pode durar de 2 a 4 semanas, sempre com a mesma duração.

No processo de GCS, deve-se administrar as versões disponíveis do software, principalmente quando mudanças são solicitadas, para alinhar com os desenvolvedores qual versão poderá ser modificada e qual a próxima versão a ser criada. É preciso estar envolvido em todos os processos de desenvolvimento, pois dúvidas e informações sobre o ciclo do software podem surgir.

Ainda, há a responsabilidade do desenvolvimento de documentações técnicas para documentar as mudanças ocorridas, o versionamento da funcionalidade entregue e quaisquer mudanças ocorridas que podem impactar o desenvolvimento dentro da organização e/ou a instalação no ambiente do cliente. Além disso, é atribuído o gerenciamento de versões e solicitações de mudanças de software de terceiros utilizados pelos times de desenvolvimento.

Quando uma nova funcionalidade deve ser entregue ao cliente, o gerente de configuração deve atentar-se às datas estipuladas pelo gerente do produto e acompanhar os prazos dos responsáveis pelos testes de software, documentação de suporte e verificação, já que é diretamente impactado caso quaisquer das ações anteriores sofram com atrasos. Por isso, o gerente de configuração deve estar presente em todas as reuniões aplicadas pelo gerente do produto, pois assim é possível levantar possíveis problemas para as entregas.

As ações cotidianas do gerenciamento de configuração de software mostraram-se de extrema importância para o controle e qualidade dos processos que envolvem mudanças dentro do desenvolvimento de software. Também, quaisquer ações que envolvem clientes, conformidades comerciais e mudanças do ciclo do projeto, o GCS é acionado, demonstrando assim, que possui impacto direto nas diretivas de mudanças, organização e qualidade do projeto.

4. Resultados e discussão

Como resultado do estudo de caso, foi possível comprovar que o Gerenciamento de Configuração é de extrema importância e o ponto chave da garantia de qualidade durante todo o ciclo de vida e desenvolvimento do software, o que também garante a entrega da solução para o cliente de forma satisfatória e precisa, promovendo assim um processo eficiente e eficaz. O GCS mostra-se em uma posição com grande poder decisivo, onde desde o desenvolvedor até a gerência utiliza como referência para a precisão de informações.

De acordo com Pressman e Maxin (2016, p. 625), “a gestão de configuração de software é um conjunto de atividades desenvolvidas para gerenciar alterações ao longo de todo o ciclo de vida de um software. A SCM pode ser vista como uma atividade de garantia de qualidade do software aplicada em todo o processo do software”. Dessa forma, o gerente de configuração é envolvido em todos os processos para justamente garantir a qualidade e a rastreabilidade de todas as mudanças e versões envolvidas no desenvolvimento do projeto.

Quando uma empresa insere o gerenciamento de configuração em seu processo, mostra-se interessada em manter a qualidade de seu portfólio e ter uma organização e controle como algo essencial, fazendo parte de sua identidade. Para o Guia PMBoK

(PROJECT MANAGEMENT INSTITUTE, 2017), a qualidade traduz o plano de gerenciamento da mesma com atividades que incluem a política de qualidade no projeto em si, assim, facilitando a identificação de causas da má qualidade e melhorando o cumprimento dos objetivos.

5. Considerações finais

Com o aprimoramento do processo de desenvolvimento de software e consequentemente o aumento da demanda de novas soluções e funcionalidades, a Engenharia de Software consolida-se como principal processo de desenvolvimento de software. Com o avanço da tecnologia, fez-se necessário que novos processos integrassem a Engenharia de Software para que os softwares a serem desenvolvidos pudessem seguir um processo mais eficaz.

Com o surgimento da Garantia da Qualidade de Software, que é algo relativamente novo nos ambientes de desenvolvimento, tal ambiente foi mudando, focando ainda mais na qualidade atrelada às metodologias ágeis e uso de ferramentas de controle e automação dos processos. Com a necessidade do controle de mudanças, surgiu o Gerenciamento de Configuração de Software, uma posição com a responsabilidade de gerenciar todo o ciclo de vida de um projeto.

Sendo do mesmo ramo da qualidade de software, o Gerenciamento de Configuração de Software mostrou-se uma posição de controle e organização para o dia a dia do desenvolvimento de software. Atrelado às ferramentas específicas de controle de versões e mudanças, o GCS consolidou-se no processo de Engenharia de Software, criando um ambiente de desenvolvimento organizado, controlado e pouco suscetível a erros.

Referências Bibliográficas

BOURQUE, Pierre. FAIRLEY, Richard E. **SWEBOK V3.0. Guide to the software engineering body of knowledge**. 3ª edition. Hoes Lane, Piscataway: IEEE, 2014.

CMMI Institute. Disponível em: <<https://cmmiinstitute.com/cmmi>>. Acesso em: 13 mai. 2019.

DANTAS, Cristine. **Gerência de configuração de software: Saiba como desenvolver software de forma eficiente e disciplinada**. Disponível em: <<https://www.devmedia.com.br/gerencia-de-configuracao-de-software/9145>>. Acesso em: 27 ago. 2018.

GROFFE, Renato José. **Maturidade no desenvolvimento de software: CMMI e MPS-BR**. Disponível em: <<https://www.devmedia.com.br/maturidade-no-desenvolvimento-de-software-cmmi-e-mps-br/27010>>. Acesso em: 20 abr. 2019.

SOMMERVILLE, Ian. **Engenharia de software**. 9ª edição. São Paulo: Pearson Prentice Hall, 2011.

PRESSMAN, Roger S., MAXIN, Bruce R. **Engenharia de Software: Uma abordagem profissional**. 7ª edição. São Paulo: AMGH Editora Ltda, 2016.

PROJECT MANAGEMENT INSTITUTE. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)/Project Management Institute.** 6^a edição. 14 Newtown Square, Pensilvânia: Project Management Institute, 2017.