

DISCIPLINA DE
GERÊNCIA DE CONFIGURAÇÃO,
CONTROLE DE VERSÃO E MUDANÇAS



SUMÁRIO

Sobre o Autor.....	4
Apresentação.....	5
UNIDADE 1	7
Introdução	7
O Que é Gerência de Configuração	8
Relevância	10
Atividades da Gerência de Configuração.....	14
Onde utilizamos a Gerência de Configuração.....	18
Considerações Finais	22
UNIDADE 2	24
Introdução	24
Definição de Artefatos	26
Objetos Básicos	28
Agrupamento de Objetos.....	30
Características de Artefatos	32
Considerações Finais	35
UNIDADE 3	37
Introdução	37
Versões de Artefatos	39
Controle de Versão.....	40
Baselines.....	47
Ferramentas.....	49
Considerações Finais	53
UNIDADE 4	55
Introdução	55
Solicitação	57
Análise e Priorização.....	59

Aprovação	62
Desenvolvimento	65
Verificação	67
Ferramentas	69
Considerações Finais	73
UNIDADE 5	77
Introdução	77
Auditoria de Configuração	79
Status Report	82
Construção de Versão Estável do Sistema.....	84
Homologação	86
Considerações Finais	88
Conclusão	90
Referências.....	95

Sobre o Autor

Professor Esp. Pedro Henrique Malvestio da Silva

Bacharel em Sistemas de Informação pela Faculdade Alvorada de Tecnologia (2013). Especialização em MBA em Gerenciamento de Projetos em TI pela Faculdade Cidade Verde (2015). Certificado em Engenharia de Requisitos pelo IREB (International Requirements Engineering Board) CPRE-FL (Certified Professional in Requirements Engineering). Trabalhei na iniciativa privada na área de Desenvolvimento de Software, Análise de Negócios e Análise de Sistemas. Tenho experiência na área de Engenharia de Software com ênfase em Desenvolvimento de Aplicações desktop, web e mobile, Análise e Elaboração de Requisitos, Liderança e Gestão de Projetos de Desenvolvimento. Atualmente sou Líder de Desenvolvimento de Projetos Ad-Rock e Salesforce na Objective Solutions. Nas horas vagas gosto muito de desenvolver jogos além de jogá-los é claro.

Caso queiram entrar em contato comigo para dúvidas ou bater um papo sobre tecnologia:

- E-mail: pedro.malvestio@gmail.com
- LinkedIn: <https://www.linkedin.com/in/pedrohmsilva/>

Apresentação

Gerência de Configuração, Controle de Versão e Mudanças

SEJA BEM VINDO(A)!

Prezado(a) acadêmico(a)! Seja bem vindo(a) à disciplina de Gerência de Configuração, Controle de Versão e Mudanças. Neste módulo, iremos abordar as atividades existentes na gerência de configuração de software, que é parte integral no desenvolvimento de um software de qualidade.

As unidades do livro foram organizadas de forma contínua separadas por atividades do processo de Gerência de Configuração, ou seja, a unidade seguinte sempre estará vinculada com a unidade anterior, portanto é bom que você leia e entenda todo o conteúdo de uma unidade e assim prosseguir para a próxima.

Na primeira unidade vamos realizar a definição da Gerência de Configuração, suas características, relevância e respectivas atividades como Controle de Versão e Mudanças, além das demais serão apresentadas.

Com os conceitos de Gerência de Configuração claros, seguimos para a Identificação de Artefatos da Gerência de Configuração, que define quais são todos os possíveis elementos gerenciados, e onde as atividades de gerência atuam.

Após entender com o que trabalhamos, apresentamos a atividade de Controle de Versão na terceira unidade, onde vamos focar no controle dos artefatos envolvidos no desenvolvimento de software, nos processos de armazenamento de versões referente às mudanças realizadas, a visualização do histórico e o agrupamento destas mudanças, e concluir com a apresentação de ferramentas utilizadas para realizar esta atividade.

Depois do Controle de Versão, na quarta unidade será abordado o Controle de Mudanças, que é o momento em que as alterações realizadas nestes artefatos serão identificadas e analisadas de forma individual para avaliar porque motivo elas foram realizadas, e com isto conseguimos controlar estas mudanças para avaliar elas são realmente necessárias, ou quando precisarmos saber o porque destas alterações seja possível rastrear as razões das modificações e quais artefatos foram afetados por elas.

Finalizamos com a atividade de Integração Contínua, onde após controlar as versões e mudanças definimos se elas atendem o objetivo, com revisões e testes, realizamos a geração

de uma versão estável do sistema para a entrega ao usuário final e entendemos as atividades de auditoria e status report que são utilizadas para monitorar estas mudanças.

Com todos estes conceitos, etapas e processos bem claros espero que você tenha o consolidado o conhecimento em Gerência de Configuração, Controle de Versão e Mudanças, de uma forma agradável contribuindo para seu crescimento pessoal e profissional.

Lembre-se “A essência do conhecimento consiste em aplicá-lo, uma vez possuído. - Confúcio” faça bom proveito deste conteúdo! Bom módulo!

UNIDADE 1

Definição de Gerência de Configuração

Introdução

A Gerência de Configuração, segundo Pressman, é uma das etapas da Engenharia de Software, que aborda o Controle de Versão dos itens que compõem nosso sistema, e também o Controle de Mudanças que ocorrem durante a execução do projeto de desenvolvimento, de modo geral é o gerenciamento realizado durante a atuação dos membros do time no desenvolvimento do sistema, que tem o objetivo de garantir que as necessidades do cliente sejam atendidas com qualidade, e atender aos critérios que definem a conclusão do projeto com sucesso, que são as características citadas a seguir:

- **Prazo:** Define se o tempo previsto para a conclusão do projeto é menor ou igual ao realizado;
- **Custo:** Verifica se o preço de execução do projeto permitiu a obtenção de lucro, e se existiu desperdício da força de trabalho, com a necessidade de retrabalho;
- **Valor:** Avalia os requisitos de sistema que foram solicitados pelo cliente foram entregues atendendo as expectativas, e também se o resultado final agrega valor satisfazendo as necessidades apresentadas.



Para possibilitar o cumprimento destes itens, e consequentemente que nosso projeto seja finalizado com sucesso, precisamos entender o que é a Gerência de Configuração, quais as suas características e esclarecer o que é realizado nesta etapa da Engenharia de Software, definir qual é a sua relevância neste contexto, com a apresentação de dados que demonstram a necessidade deste processo, e a importância da sua aplicação durante a implementação ou manutenção de um sistema, definir quais são as Atividades da Gerência de configuração

para viabilizar a execução deste processo em nosso projeto, suas responsabilidades e também em quais cenários elas se fazem necessárias e situações onde é possível utilizá-las parcialmente, fazendo com que nosso projeto este aderente aos critérios de prazo, custo e valor.

O Que é Gerência de Configuração

O termo Configuração normalmente é relacionado a figura de uma engrenagem, com esta imagem em mente podemos pensar que a Gerência de Configuração é a atividade ou o processo realizado para manter esta engrenagem funcionando, manter ela girando na nossa máquina. Voltando esta visão para a área de Tecnologia da Informação (TI) também relacionamos com o responsável pela infraestrutura da nossa empresa, com seus consoles abertos realizando a configuração de uma rede, monitorando o tráfego das informações ou criando um novo servidor para hospedar as nossas aplicações.



De modo geral a Gerência de Configuração está relacionada com estas atividades citadas acima, onde temos a responsabilidade de garantir que o nosso sistema continue funcionando, independente das interferências que ele sofra durante seu ciclo de vida, ou seja, a responsabilidade deste processo é garantir a qualidade da nossa aplicação independente da situação que ela seja exposta, e para que isto ocorra precisamos executar as seguintes atividades da Gerência de Configuração, são elas:

- Identificação de Artefatos;
- Controle de Versão;
- Controle de Mudanças;
- Integração Contínua.

Para facilitar a compreensão destas atividades vamos relacioná-las com o dia a dia de um artista que irá pintar um quadro. Antes de começar a pintura ele precisa selecionar os materiais que serão utilizados, neste caso a tela, o lápis, a borracha, as tintas, espátula, entre

outros, com os materiais selecionados ele inicia os rabiscos a lápis na tela em branco para definir qual é o desenho que será realizado, durante esta etapa inicial são necessárias algumas correções então ele apaga diversas vezes o seu desenho inicial até chegar na etapa desejada e inicia a pintura, muitas vezes precisa remover a tinta do quadro algumas vezes, trocar a paleta de cores para que sejam compatíveis, este processo é repetido inúmeras vezes, pode até ser necessário pintar a tela de branco e começar tudo de novo. Se pensarmos em cada etapa da pintura deste quadro, foram necessárias diversas mudanças que surgiram por necessidades do pintor, para realizá-las foram geradas diversas versões do quadro, diversos desenhos a lápis e várias variações de cores foram inseridas e retiradas deste quadro, e após concluir cada uma destas alterações ele avaliou se elas atendiam às suas expectativas até que ele foi possível chegar a uma versão final onde ele chega na sua obra prima.



Na Gerência de Configuração, iniciamos com a Identificação de quais elementos serão utilizados para atingir um objetivo, neste caso são os materiais, importante lembrar que cada lápis, tinta, tela possui uma característica como por exemplo a densidade do lápis 2B, 4H, a numeração da cor da tinta 238, 456 e o tamanho da tela 20x30, 100x90. Com os materiais definidos o Controle de Versão atua em cada alteração realizada, para esta situação a mudança do material, a aplicação dele na tela, cada traçado gerado na tela, para que seja possível identificar qualquer alteração realizada. Um conjunto de alterações representam uma mudança, e com isto o Controle de Mudanças visa identificar quais são as necessidades que motivam aquela mudança e qual o resultado esperado para que elas sejam consideradas como atendidas, e após concluir cada mudança é realizada a Interação Contínua, onde garantimos se as necessidades foram atendidas de acordo com a expectativa do cliente.

Como mencionado anteriormente o objetivo final é a garantir a qualidade da execução do desenvolvimento de um sistema, por isso a Gerência de Configuração é uma das etapas da Qualidade de Software abordadas na Engenharia de Software, será possível visualizar que algumas atividades abordadas dentro do processo de Garantia de Qualidade também são realizadas na Gerência de Configuração, para que possamos gerar um software que atenda as necessidades de nossos clientes e usuários finais.

Segundo Pressman “não importa onde você esteja no ciclo de vida de um sistema, ele irá mudar, e o desejo de mudança irá persistir por todo este ciclo de vida”, ou seja, a mudança é algo constante durante o desenvolvimento de software, é a única certeza que temos, as leis, necessidades, usuários, hardwares, fornecedores, processos e etc mudam, e na maioria das vezes temos que reagir a estas mudanças para que nosso sistema continue útil e atenda ao seu objetivo, vale lembrar que sem usuários ou alguém que precise da nossa aplicação ela não terá razões para existir e com isso nossa função também, então temos por obrigação manter o software vivo e necessário, uma relação que pode facilitar a visualização desta necessidade é a realização de uma cirurgia, mesmo que seja estética, tem como objetivo melhorar a qualidade de vida do paciente, estamos aqui para tratar dos problemas de nossos usuários para que eles continuem bem com o sistema.



Relevância

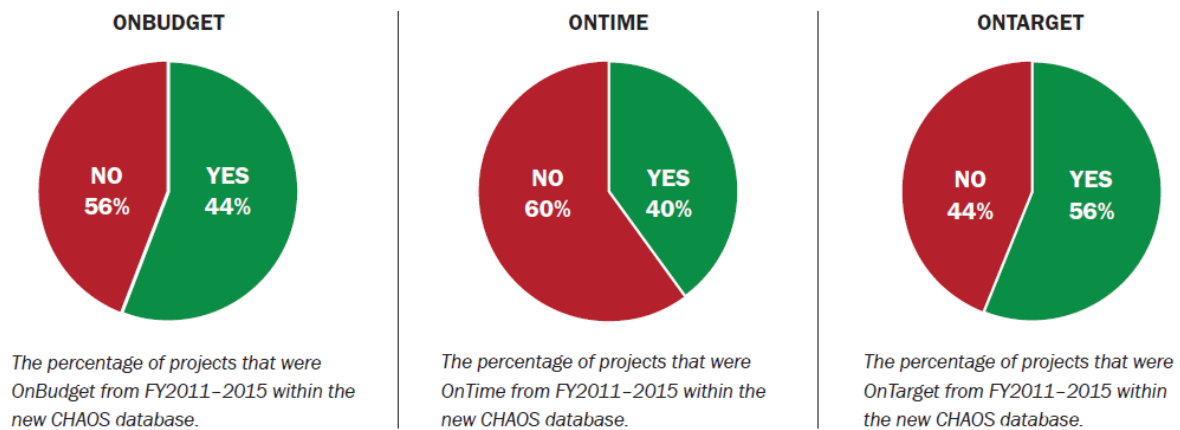
O grupo The Standish Group International realiza uma pesquisa anual chamada CHAOS Report, onde é realizada uma análise anual de 5 mil projetos de diferentes portes, do gigantesco até o pequeno, e publicados os resultados para validar a taxa de sucesso desses desenvolvimento de software, além das diversas causas que levam eles ao sucesso ou o fracasso, dentro dos parâmetros de sucesso definidos pelo grupo que é voltado para a satisfação do cliente, são eles Valuable (Agregou Valor), OnGoal (Atingiu o Objetivo) e Satisfactory (Satisfatório), o resultado da pesquisa realizada em 2015 constatou uma taxa de sucesso de 29%, ou seja, dos 5 mil projetos somente 1450 atenderam as necessidades de seus clientes, os outros 71%, que representam 3550, restantes não foram entregues ou

tiveram problemas para que a entrega fosse realizada, o The Standish Group chama estes de “Projetos Desafiadores”.

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Analisando a perspectiva de Prazo e Orçamento/Custo vemos que entre 56 a 60% dos projetos não atendem estas necessidades, ou seja, a data de entrega não é cumprida e os custos são maiores do que o esperado, sejam eles arcados pela empresa ou pelo cliente. Por ser um processo de Qualidade de Software ligado diretamente a estes dois aspectos citados anteriormente, ao aplicarmos as Atividades da Gerência de Configuração a um projeto de desenvolvimento de software garantimos que não vamos precisar ter esforço com alterações não implementadas, perdidas ou não realizadas conforme solicitado, o que faz com que o tempo e o orçamento do projeto seja atendido e com isto aumentamos a chance de atingirmos o sucesso e a satisfação de nossos clientes, ou seja, quando a Gerência de Configuração é aplicada no desenvolvimento de um sistema conseguimos amenizar os fatores que normalmente fazem com que o projeto fracasse, e conseguimos alcançar o nosso principal objetivo, que é manter o nosso sistema relevante que ele entregue valor para nossos clientes e com isso nós continuemos necessários.



Após a relevância da Gerência de Configuração ficar clara quando analisamos os dados dos projetos que falharam, precisamos entender como conseguimos garantir que as alterações não implementadas, perdidas ou não realizadas conforme solicitadas, que definem o fracasso de um projeto, não aconteçam durante o desenvolvimento de um sistema, para isso é necessário entendermos as atividades realizadas neste processo de Qualidade de Software, e com isso o nosso projeto seja concluído com sucesso.

Como já realizamos anteriormente, vamos utilizar um cenário um pouco fora do contexto para facilitar a compreensão destas atividades, quando lemos um livro e assimilamos os conceitos apresentados pelo autor, na maioria das vezes não paramos para pensar na quantidade de esforço realizado pelas pessoas que contribuíram para que ele fosse publicado, mesmo que seja um escrito independente o manuscrito passa por diversas etapas até que chegue em seu estado final, o diagrama a seguir representa os processos realizados desde a escrita do livro até sua publicação:

desenvolvimento de software e a interação dos membros da equipe com o sistema, e com isso diminuir a possibilidade de falhas humanas durante todo o processo, para conseguirmos reduzir os riscos gerados por esta interação, quando eles não são extintos.

Atividades da Gerência de Configuração



Por diversas vezes definimos a Gerência de Configuração como um processo, que por definição, segundo o dicionário Oxford, é uma série de atividades ou passos em ordem para atingir um determinado fim, ou seja, são etapas realizadas em uma sequência específica que têm um objetivo final, e para entendermos de fato como executamos a Gerência de Configuração precisamos entender quais são suas atividades, cada uma delas será apresentada em sua sequência cronológica a seguir:

1. Identificação de Artefatos

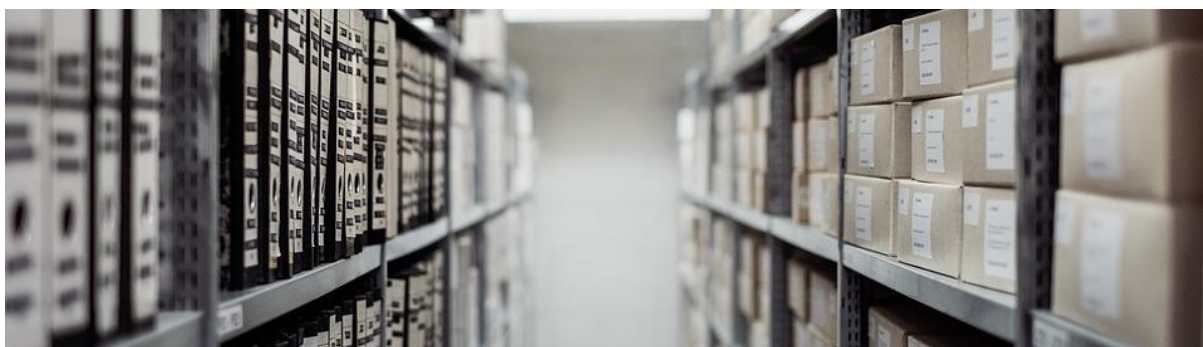
A Identificação de Artefatos, ou a Identificação dos Itens de Configuração, é a atividade responsável por selecionar todos os “arquivos” que serão controlados durante todo o processo de desenvolvimento de um software, como existe a possibilidade destes elementos não serem especificamente arquivos, como softwares, o estado de um servidor, e até mesmo um conjunto de arquivos, para facilitar o entendimento vamos chamá-los de artefatos. Alguns já foram citados anteriormente mas podem ser também o código fonte do sistema, a documentação de uma funcionalidade, a configuração de um servidor, as evidências de teste da homologação do sistema, entre outros.



Quanto maior o tamanho do sistema, maior também será a quantidade de artefatos controlados pela Gerência de Configuração, e muitas vezes a identificação deve chegar ao vínculo entre estes itens, pois a versão de framework pode afetar o funcionamento de uma funcionalidade, a troca de um software utilizado pode gerar a incompatibilidade e fazer com que seja necessário a criação de um novo arquivo, que por sua vez pode desencadear outros impactos no restante do sistema. Por isso é importante realizar este mapeamento e controlar o máximo de artefatos possíveis para que caso uma alteração seja necessária, um serviço ou produto seja descontinuado, ou alguma licença vença durante a execução do nosso projeto tenhamos plena ciência de quais impactos terão que ser tratados para atender a necessidade gerada por estes imprevistos ou solicitações.

2. Controle de Versão

Com todos os nossos artefatos identificados e sabendo que durante o desenvolvimento de software nossa única certeza é a que ocorrerão mudanças, precisamos controlar cada estado interação realizada pela equipe em cada um destes itens controlados. Para isto a atividade de Controle de Versão visa armazenar cada uma das alterações realizadas nos artefatos possibilitando a visualização da evolução de cada uma delas e também a reversão de alguma alteração que está paralisada, gerou algum erro ou ainda não será necessária para o cliente, esta atividade é chamada de Controle de Versão pois é comum nomearmos um arquivo ou alterarmos a propriedade definindo uma versão com uma sequência de números para facilitar a distinção e visualização desta sequência de alterações realizadas.



No exemplo do Autor, é possível visualizar cada versão gerada para o livro durante o seu processo de elaboração, neste caso o controle foi realizado de forma manual onde a versão deste arquivo foi definida e inserida no nome do documento, porém para o desenvolvimento de software, focando principalmente na concorrência entre os desenvolvedores por exemplo onde dois ou mais estão atuando no mesmo projeto ao mesmo tempo, e a alteração do nome de um arquivo pode fazer com que o sistema inteiro pare, e até mesmo a definição manual de cada uma das versões geradas até que a implementação seja finalizada, é inviável a mudança manual do arquivo, por este motivo existem estratégias e ferramentas que auxiliam este controle com o objetivo não só de versionar este arquivo, mas também assimilar alterações realizadas ao mesmo tempo.

3. Controle de Mudanças

O Controle de Mudanças apesar de ter uma nomenclatura que lembre o Controle de Versão, que como acabamos de ver controla as mudanças dos arquivos, tem uma função diferente, o seu objetivo é gerenciar as mudanças como um todo, fazendo uma relação entre as atividades o Controle de Versão tem como função atender a uma necessidade demandada pelo Controle de Mudanças, utilizando um exemplo prático quando precisamos perder peso, realizamos uma série de atividades, que podemos chamar alterações em nosso comportamento como: evitar comer doces, fazer atividades físicas, controlar quantas calorias consumimos e também quantas gastamos até que atingimos nossa necessidade inicial, que é perder peso. A Atividade de Controle de Mudanças gerencia o que motivou uma necessidade mudança e do que ela se trata, quais são as alterações necessárias para que ela seja executada, e também quais as características que fazem ela ser considerada como concluída.



Relacionando com o exemplo apresentado a nossa necessidade de mudança é perder peso, os motivos que geram esta necessidade podem ser orientação médica, auto estima ou operação verão, as alterações necessárias são mudança de hábitos alimentares, prática de exercício físico e controle de consumo e queima de calorias, e somente concluímos nossa necessidade quando emagrecemos 10kg. Quando avaliamos estas alterações como um todo

elas possuem uma definição muito vaga, é normal dividirmos em mudanças específicas, no caso da mudança de hábitos alimentares, podemos dividir em consultar nutricionista, preparar cardápio semanal, comprar alimentos orgânicos, diminuir consumo de produtos industrializados, e assim por diante, cada uma destas mudanças representam uma etapa do agrupamento e a conclusão de todas representa a conclusão de um agrupamento. No caso do desenvolvimento de software cada uma das necessidades são representadas por funcionalidades do sistema, e as alterações realizadas nos artefatos de software geram comportamentos que representam o objetivo daquele comportamento solicitado.

4. Integração Contínua

Após controlarmos as alterações realizadas nos artefatos identificados para atender a uma necessidade de mudança específica, precisamos garantir que aquela mudança seja realizada entregue ao nosso cliente conforme ele solicitou sem erros, sejam eles de desenvolvimento ou publicação. Na atividade de Integração Contínua realizamos o fechamento do nosso desenvolvimento, onde agrupamos todas as versões finais de artefatos geradas pelas mudanças solicitadas, avaliamos se foram realizadas conforme os padrões de desenvolvimentos definidos, testamos para garantir se que não apresentem erros, geramos relatórios de auditoria para comprovar quais foram as alterações realizadas e também evidenciar que estão funcionando conforme o esperado e após entregamos para que seja possível a homologação por parte do cliente, e também o teste de disponibilização da funcionalidade. Em síntese, são as atividades necessárias para garantir que tudo que foi feito esteja de acordo com o que foi solicitado e em pleno funcionamento para satisfazer a necessidade do usuário.



A nomenclatura Integração Contínua é referente a constante necessidade de mudança, onde a cada momento ocorrem entregas para atender demanda de solicitação do cliente, e com isso tudo que estamos executando está sendo disponibilizado continuamente ao nosso usuário. Até mesmo pela introdução do conceito do desenvolvimento ágil que visa disponibilizar o sistema em pequenas partes viabilizando a validação do atendimento da

necessidade do cliente, sem que ele espera por meses até receber o produto final, e com isto seja possível assimilar as mudanças que ocorreram este período, permitindo que o problema atual do cliente seja resolvido. Além das entregas de sistema são disponibilizados diversos outros documentos e relatórios permitem a visualização do que ocorreu durante o desenvolvimento da mudança em questão e como ela afetou o sistema como um todo, para que sejam utilizadas como insumo no momento de solicitar as demais necessidades do cliente.

Por representarem o que é a Gerência de configuração, cada uma destas atividades possuem um capítulo específico para uma abordagem mais branda e também para que sejam apresentados os cenários de execução de cada uma delas, além da apresentação de ferramentas e conceitos de mercado que permitirão a aplicação prática de cada um destes tópicos, e como um todo a execução da Gerência de Configuração.

Onde utilizamos a Gerência de Configuração

Normalmente quando abordamos os termos gerência ou configuração, nós vem da figura de um responsável pela execução de uma tarefa relacionada a isto, como no caso do gerente de projetos ou do responsável pela infraestrutura configurando os servidores de rede, na Gerência de Configuração o responsável normalmente agrega o acompanhamento das atividades e realiza uma função dentro do projeto como desenvolvimento, análise e gerência de projeto, isto é possível pois cada um dos membros do time é responsável pela garantia da qualidade do software, por esta particularidade temos facilidade executar estas atividades durante o desenvolvimento do nosso sistema.

Por se tratar de um processo de controle, onde as atividades têm foco em analisar cada uma das modificações realizadas dentro do nosso sistema, a aplicação da Gerência de Configuração pode ser realizada de forma parcial, pois temos que avaliar qual é a necessidade do nosso projeto, e até mesmo prazo e custo são fatores determinantes para definir a inclusão das atividades dentro do nosso processo de desenvolvimento. Uma forma simples de identificar a necessidade de aplicação é o pelo tamanho do projeto, quando estamos desenvolvendo um projeto pessoal, para aprender uma nova linguagem por exemplo ou protótipo de um sistema, onde as mudanças não terão objetivos específicos ou serão solicitadas e realizadas por nós mesmos a necessidade de gerenciar os artefatos, controlar cada versão e mudança realizada além da geração de relatórios e testes demanda um esforço maior muitas vezes que o tempo disponível para realizar aquela atividade. Porém quando atuamos em um projeto de grande porte, onde temos diversos membros na equipe, prazo definido e restrições de custo, fica evidente que é necessária a realização de um controle

WEEK 3: Design

WEEK 4: Design

WEEK 5: Design

WEEK 6: Dev

WEEK 7: Dev

Finalize API Goals

Infrastructure Recs: - Laravel + Backbone

Create Fixtures for testing

Admin Dashboard B/E theme work

Database API (controllers)

B/E Sorting (for all Dashboard Views) of products

B/E Filter (form for all users)

B/E (Grid for all DB Tables)

Deployment

Admin API

Create Admin API

PLEASE DO NOT ERASE

Welcome to the project

Além da necessidade situacional de inserção do processo de Gerência de Configurações no desenvolvimento de nossas aplicações, existem entidades que padronizam e também atestam que executamos as atividades no dia a dia, até mesmo definem a necessidade inclusão com base na maturidade da equipe ou empresa como um todo, ou seja, existem entidades que especificam quais Atividades da Gerência de Configuração precisa ser executadas com base no tamanho do nosso projeto, além de facilitar a definição do nível de adesão do processo um instituto de padronização atesta que aplicamos a Gerência de

Configuração no nosso dia a dia e com isto conseguimos comprovar para possíveis clientes que exigem o cumprimento de determinadas atividades que atendemos a este requisito, e com os certificados agregamos valor nas entregas realizadas pela nossa equipe de desenvolvimento e também conseguimos embasar a cobrança de um valor maior na contratação de nossos serviços. Segue a listagem de entidades e suas respectivas padronizações que contemplam a Gerência de Configuração:

Entidade	Certificação	Descrição
IEEE Standards	IEEE 828	Software Configuration Management Plans
IEEE Standards	IEEE 1042	Software Configuration Management
ISO Standards	ISO 10007-1995	Quality Management, Guidance for CM
ISO Standards	ISO/IEC 12207	Information Technology-Software Life Cycle Processes
ISO Standards	ISO/IEC TR 15271	Guide for ISO/IEC 12207
ISO Standards	ISO/IEC TR 15846	Software Engineering-Software Life Cycle Process-Configuration Management for Software Order
EIA Standards	EIA 649	National Consensus Standard for Configuration Management
EIA Standards	EIA CMB4-1A	Configuration Management Definitions for Digital Computer Programs
EIA Standards	EIA CMB4-2	Configuration Identification for Digital Computer Programs
EIA Standards	EIA CMB4-3	Computer Software Libraries
EIA Standards	EIA CMB4-4	Configuration Change Control for Digital Computer Programs
EIA Standards	EIA CMB6-1C	Configuration and Data Management References Order
EIA Standards	EIA CMB6-3	Configuration Identification
EIA Standards	EIA CMB6-4	Configuration Control
EIA Standards	EIA CMB6-5	Textbook for Configuration Status Accounting
EIA Standards	EIA CMB7-1	Electronic Interchange of Configuration Management Data
U.S. Military Standards	DoD MIL STD-973	Configuration Management
U.S. Military Standards	MIL-HDBK-61	Configuration Management Guidance
Outras Entidades	DO-178B	Guidelines for the Development of Aviation Software
Outras Entidades	ESA PSS-05-09	Guide to Software Configuration Management
Outras Entidades	AECL CE-1001-STD	Standard for Software Engineering of Safety Critical Software
Outras Entidades	DOE SCM checklist:	http://cio.doe.gov/ITReform/sqse/download/cmcklst.doc
Outras Entidades	BS-6488	British Std., Configuration Management of Computer-Based Systems
Outras Entidades	Best Practice—UK	Office of Government Commerce: www.ogc.gov.uk
Outras Entidades	CMII	Institute of CM Best Practices: www.icmhq.com

Cada uma das certificações das entidades listadas acima possuem atividades específicas da Gerência de Configuração e também definem para qual situação elas se aplicam, para que o nosso projeto seja certificado é necessária a aplicação de uma avaliação que é realizada por um comitê avaliador formado por especialistas no assunto, em sua maioria não só a execução das atividades é necessária, é preciso apresentar as evidências para comprovar que as mesmas são auditadas e existe um controle interno sobre elas, por este motivo a aplicação Integrada Contínua é essencial para a qualificação de um projeto.



Considerações Finais

Os termos Gerência e Configuração de modo geral estão ligados a atividades que são responsáveis por acompanhar e garantir o funcionamento de determinado componente, seja ele de software ou não, aliado a certeza que as mudanças ocorrem em nosso dia a dia, quanto mais durante o desenvolvimento de um sistema que irá atender diversas pessoas com necessidades distintas, visões diferentes do que é necessário para aquela aplicação, a Gerência de Configuração é responsável por manter o nosso sistema em funcionamento, mesmo em cenários de pouca ou muita mudança, fazendo com que a qualidade seja mantida mesmo com manutenções ou alterações realizadas em nossa aplicação, principalmente por ser uma etapa do processo de Qualidade de Software.

Quando avaliamos as principais causas que levam um projeto a fracassar, vemos que as maiores motivações são porque os custos do projeto foram excedidos, o prazo não foi atendido ou o que foi desenvolvido não resolve o problema do cliente, o famoso não foi isso que eu pedi, que são falhas graves que podem ainda acontecer ao mesmo tempo, quando relacionamos estes dados com o objetivo da Gerência de Configuração que é manter a qualidade, não só nas entregas ao nossos clientes mas também prevenir que falhas ocorram durante o ciclo de desenvolvimento de software, fica evidente que incluir este processo na execução do nosso projeto diminui a possibilidade do fracasso, tratando da sua principal causa que é a mudança.

Para garantir que o nosso sistema continue em funcionamento, e se adapte às mudanças que ocorrem durante o ciclo de desenvolvimento, é necessário que as Atividades da Gerência da Configuração sejam aplicadas em nosso projeto, fazendo com que a possibilidade de ocorrerem falhas sejam reduzidas ou até mesmo extinguidas, para que isto aconteça as seguintes atividades e suas respectivas definições devem ser realizadas: Identificação de Artefatos, realiza a descoberta de quais serão os artefatos de software que devem ser controlados; Controle de Versão, cada um destes itens definidos anteriormente têm suas alterações versionadas, ou seja, é gerada uma base de comparação entre o estado atual e os estados anteriores destes artefatos para gerenciar as modificações realizadas; Controle de Mudanças, define cada necessidade de mudança, quais os resultados esperados e os itens que compõem aquela alteração; Integração Contínua, garante que as modificações foram realizadas de acordo com seus requisitos, gera relatórios para auditar o foi realizado durante estas alterações e entrega a funcionalidade solicitada para o cliente.

Com os conceitos da Gerência de Configuração e suas Atividades detalhados, abordamos o contexto de aplicação deste processo com base na complexidade e tamanho do projeto, onde

apresentamos sua utilização no cenário otimista e pessimista, e também qual a abordagem em situações intermediárias, onde é preciso selecionar quais etapas são necessárias para que a Garantia da Qualidade ocorra, sem que ocorram desperdícios de tempo e recursos durante a execução do nosso projeto. Para apoiar inclusão da Gerência de Configuração no ciclo de desenvolvimento é possível utilizar as certificações criadas pelas entidades padronizadoras que definem as situações e porte de projetos que precisam deste processo, além de avaliarem a sua execução, e no caso de aprovação, atestam que contramos com as atividades necessárias por meio de um certificado.

UNIDADE 2

Identificação de Artefatos da Gerência de Configuração

Introdução

Quando identificamos algo no dia a dia normalmente realizamos isto para diferenciar algum objeto dos demais, normalmente colocamos nome no nosso material escolar para não confundir com o amigo, colocamos um nome ou referência em pastas para facilitar definir qual seu conteúdo, e datamos as fotos e até vinculamos localização para lembrarmos quando e onde foram tiradas. A organização de papéis, objetos de uso diário e pessoais, como presentes ou lembranças, é importante para encontrarmos aquele item quando necessário, o mesmo acontece na Engenharia de Software, precisamos identificar os itens que compõem nosso projeto, para controlar a evolução do nosso sistema, e pensando no ciclo de vida da aplicação o volume de dados gerados por cada uma das atividades realizadas durante a construção de um software é grande, e para que seja possível controlar esta quantidade de itens é preciso identificar cada um deles e com isso encontrá-los quando necessário.



A Gerência de Configuração se refere aos objetos que fazem parte do projeto como Artefatos ou Itens de Configuração, que são todos os itens que estão envolvidos direta ou indiretamente com o desenvolvimento do software, e atividade de Identificação de Artefatos tem como objetivo avaliar quais são os objetos que precisam ser controlados, a fim de que não sejam definidos mais do que o necessário ou o adequados ao porte e complexidade do sistema, evitando que esta atividade consuma um esforço maior ou menor do que o necessário para o cenário. Para selecionar os Artefatos que fazem sentido para o projeto, precisamos separar estas entidades em dois grupos os Objetos Básicos e o Agrupamento de Objetos.

Objetos Básicos são a menor unidade controlada pela Gerência de Configuração, e também são os mais numerosos no sistema, o formato mais comum para este Artefato é um arquivo como código fonte, requisitos, planos de teste, evidências de teste, porém em casos mais específicos podem chegar a ser um pedaço de cada um destes itens citados acima, e este grau de granularidade para o Objeto Básico é definido de acordo com o objetivo do projeto e também as restrições impostas, como prazo e custo, pois quanto mais granulares forem os Artefatos maior é o esforço e consequentemente o custo necessário, e este é um dos pontos que pode viabilizar ou não a execução do desenvolvimento.

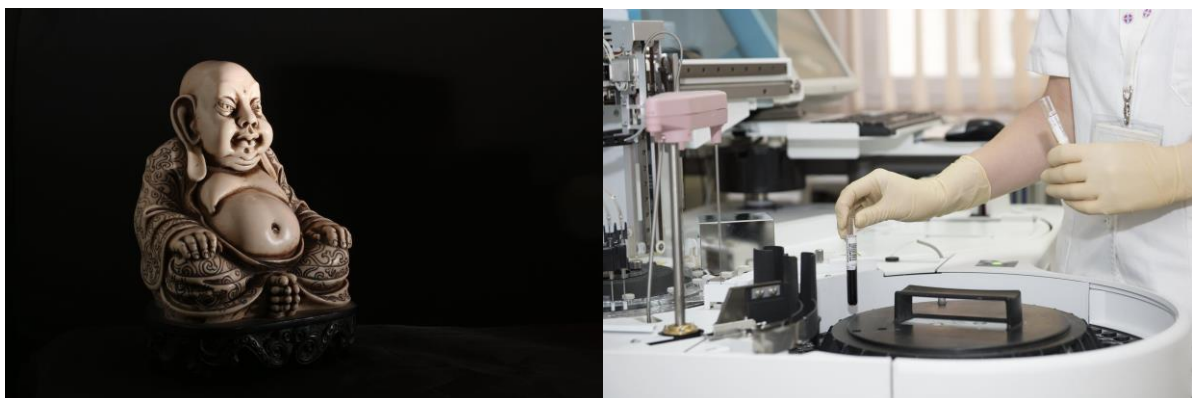
Após definir como são e quais são nossos Itens de Configuração, realizamos o Agrupamento de Objetos para facilitar seu gerenciamento, a identificação do relacionamento entre os Artefatos e a sua responsabilidade ou funcionalidade dentro da aplicação. Ao reunir os

objetos em grupos, é possível visualizar qual o papel deles como um todo e facilitar a identificação dos Artefatos que precisam ser alterados no momento em que uma demanda é solicitada pelo cliente, como um grupo pode ser composto não só por um tipo de Item de Configuração mas por diversos tipos, ele permite a visualização de quais recursos serão necessários para implementar aquela necessidade com sucesso.

Para identificarmos de forma única cada um destes Artefatos precisamos definir os metadados de cada um deles, são informações que fazem aquele objeto ser único dentro do projeto ou do contexto em que ele está inserido, ou seja, as Características que permitem a diferenciação de cada Item de Configuração. Estas informações são consideradas metadados pois são dados externos controlados por um sistema ou documento, que seguem convenções definidas pelo Gerente de Configuração, que seleciona itens necessários não só para as distinção dos Artefatos mas também para o controle realizado pelas demais Atividades da Gerência de Configuração.

Definição de Artefatos

A Gerência de Configuração atua no controle das mudanças que ocorrem durante o desenvolvimento de um sistema, o alvo deste gerenciamento é o Item de Configuração ou Artefato de Software que de acordo com o dicionário de Oxford possui duas definições, a primeira é “Um objeto feito pelo ser humano, tipicamente um item com interesse histórico ou cultural”, e a segunda é “Alguma coisa observada em uma investigação científica ou experimento que não existe em sua forma natural, mas ocorre como resultado de um procedimento preparatório ou investigativo”, parte de cada uma geram a definição que traduz para o nosso contexto, é um “objeto” feito pelo homem como resultado de um processo ou procedimento.



A definição de Artefato ou Item da Gerência de Configuração segundo Pressman é a “Informação criada como parte do processo de Engenharia de Software”, considerando o

desenvolvimento de um sistema onde diversos dados são gerados, e que o conjunto deles é a informação, o agrupamento dos arquivos de código fonte por exemplo gera o executável do sistema, para o nosso contexto são especificamente os itens que compõem a configuração estável do sistema de acordo com sua evolução. Porém nem todos os Artefatos gerados necessariamente precisam ser gerenciados durante o desenvolvimento de um software, por este motivo entender quais destes itens serão necessários para o nosso projeto, e quais não, vai diminuir o desperdício de tempo e esforço das demais Atividades da Gerência de Configuração com Artefatos desnecessários, garantido que o Prazo e Custo do projeto sejam atendidos ou até reduzidos.

Normalmente relacionamos os itens de configuração, com arquivos que conseguimos acessar e dar manutenção porém até mesmo comportamentos e processos realizados durante implementação do sistema podem ser considerados Artefatos, como o horário de publicação de uma versão para o cliente, o dia da semana que o software é atualizado, o passo a passo necessário para disponibilizar a aplicação no servidor, até o período onde o sistema tem o menor fluxo de acesso pode ser uma configuração pois viabiliza a entrada de uma correção de um bug de produção que pode eventualmente ocorrer. É importante ter em mente que não só do cliente mas também da equipe como horários do dia onde temos pico de produtividade, perfil dos recursos, como Júnior, Pleno e Sênior, para definir qual deve executar atividades com maior complexidade, ou qual está apto a atuar em demandas mais complexas para subir de nível.

Ao iniciarmos um projeto e até mesmo quando ele está em andamento, é comum não pararmos para pensar quais são os itens de configuração que estão sendo gerenciados nele, principalmente quando estamos executando um projeto de curto prazo, que precisa ser entregue rápido, neste caso não só o porte do projeto irá influenciar diretamente na Identificação de Artefatos gerenciados, mas também o contexto do projeto, pois na situação de um projeto pequeno onde só uma pessoa irá atuar, o volume de dados que precisa ser tratado não necessita de um controle específico, porém caso uma equipe esteja atuando o controle é necessário para que nada se perca durante o caminho e o prazo não seja afetado por falhas onde alterações foram sobrepostas ou perdidas, arquivos não foram entregues ao cliente, ou uma ata de reunião não foi enviada, e é neste momento que a Gerência de Configuração se faz mais necessária pois são nestes projetos que não nos atentamos aos detalhes que podem fazer com que ele seja um fracasso.



A inclusão Gerência de Configuração na execução do projeto é delimitada de acordo com seu porte, e quais Artefatos de Software serão gerenciados é determinado de acordo com a quantidade e qualidade de pessoas envolvidas na implementação do sistema, a junção destes dois parâmetros aliado ao objetivo do projeto define a o grau de granularidade do gerenciamento, partindo do maior até o menor nível respectivamente, o primeiro cada método e função do sistema, tópico de requisito ou tabela do sistema é um Artefato, no segundo o sistema como um todo, o banco de dados, a documentação do software são itens de configuração. Por este motivo definir como será realizado este controle, gera a visão de qual o esforço necessário para gerenciar estes artefatos, e quanto maior a granularidade definida para o Item de Configuração maior será o esforço demandado para a execução do controle, mantendo ele válido durante a evolução do sistema além da suas versões estáveis para cada configuração do software, estes Artefatos são divididos em dois grupos: Objetos Básicos e Agrupamento de Objetos, ou Objetos Agrupados, que serão apresentados a seguir.

Objetos Básicos

São unidades de informação criadas durante a análise, design, desenvolvimento ou teste, ou seja, são os dados gerados a partir de uma das atividades da Engenharia de Software, na sua forma mais pura, ou seja, a maior granularidade, é um método do código fonte, um trecho do documento de requisitos do sistema, uma tabela ou uma query do banco de dados, uma das configurações do servidor, porém de acordo com a necessidade do projeto, e seus demais fatores, pode ser um arquivo de código fonte, o requisito, o estado atual do banco de dados, ou scripts que geram a estrutura atual do banco de dados e também a configuração atual do servidor.



A geração de objetos básicos durante o ciclo de vida do sistema é contínua, por isso é necessária a definição de qual o nível de controle necessário, a fim de garantir que o tempo e o custo necessário para realizar o controle desses artefatos não seja excedido e atenda aos valores definidos para o sucesso do projeto, pois a única certeza é que após a sua criação estes artefatos sofrerão alteração. Normalmente a definição de um objeto básico está relacionada diretamente ao objetivo do projeto, e suas restrições, por exemplo caso a margem de lucro do sistema esteja relacionada com a quantidade de bugs encontrados pelo cliente, ou cada método da aplicação será utilizado por outro software para realizar operações e a validação de compatibilidade é a versão do serviço, um controle com a granularidade máxima onde o gerenciamento das funções da classe é realizado e irá garantir um controle que permita a verificação do atendimento da necessidade de forma mais específica o que diminui a chance de falha, porém nos casos onde a data de entrega é mais importante, e o não cumprir com o prazo gera multa, quanto maior for o esforço para realizar este controle, ou seja, quanto mais específico e granular for o objeto básico, maior a quantidade de tarefas para gerenciar as alterações e também a dificuldade em realizá-las, e consequentemente maior o desperdício de tempo com atividades que não agregam valor para o cliente.



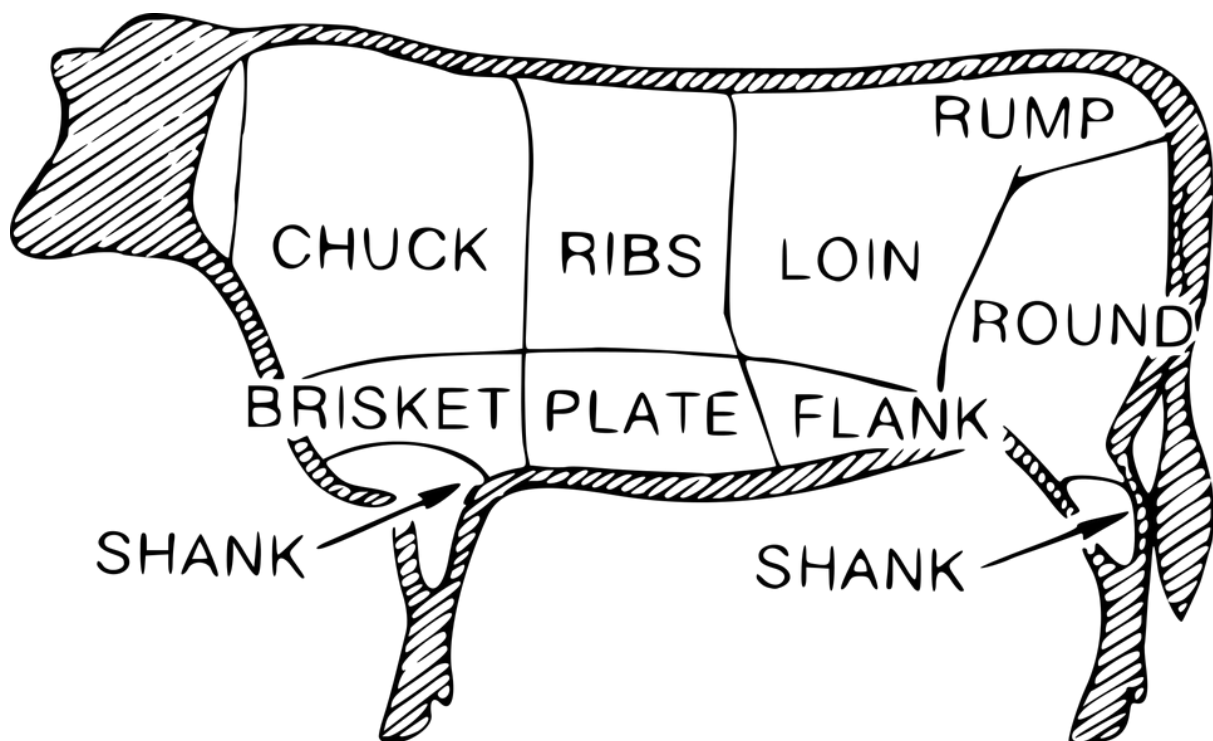
```
40
41 $(function){cards();});
42 $(window).on('resize', function(){cards();});
43
44 function cards(){
45     var width = $(window).width();
46     if(width < 750){
47         cardssmallscreen();
48     }else{
49         cardsbigscreen();
50     }
51 }
52 function cardssmallscreen(){
    var cards = $('<div>.card').length;
    var height = 0;
    card2 = 1; i<=cards; i++){
        i = $('<div>.card').length;
    }
}
```

Por este motivo é importante identificar qual é a definição ideal dos Objetos Básicos de acordo com o contexto e objetivo de cada projeto, e utilizar o fundamento da Gerência da Configuração a nosso favor, que é a mudança, pois caso durante o gerenciamento dos Artefatos Identificados fique evidente que esta atividade está levando muito mais tempo do que é necessário para o nosso projeto, mesmo que a definição esteja alinhada com o objetivo, como gerentes de configuração temos que identificar que a alteração é necessária e atuar

para que a definição seja adequada a necessidade do projeto, para o caso contrário também, estão sendo gerados muitos erros no sistema, existe a possibilidade de aumentar a granularidade para facilitar a avaliação do código. Além de que para diversas partes do projeto este nível de detalhamento pode ser diferente, por exemplo para os requisitos o documento inteiro e para o software os métodos.

Agrupamento de Objetos

O Agrupamento de Objetos, ou Objetos Agrupados, é, como o nome diz, o conjunto de Objetos Básicos e também outros Agrupamentos de Objetos, pode ser visualizado como uma lista de identificadores, ou ponteiros que especificam o grupo, como no caso do Modelo de Arquitetura e o Modelo de Dados, o Diagrama de Classes e os Componentes de cada Classe. Agrupando esses objetos e grupos facilitamos é possível identificar ou definir a qual funcionalidade aquele agrupamento pertence ou é responsável, mesmo que os Objetos Básicos não estejam física, ou logicamente, no mesmo local, é possível utilizar a listagem com os ponteiros para apontar cada arquivo inserido no grupo e qual sua localização.



No caso de um frigorífico, ou açougue, que produz cortes de vaca, porco e galinha, entre outros derivados, é necessária a definição de quais são os tipos de carne de cada parte dos animais que irá gerar o corte que será comercializado, e de acordo com a estratégia de negócio são definidos quais são os tipos de carnes e partes do boi que são utilizadas e outras partes que podem ser necessárias para outras empresas como por exemplo o couro para

vestuário ou fabricação de chiclete, neste cenário podemos dizer que o grupo Boi é composto pelos cortes de carne vermelha, e o agrupamento de carne branca possui os grupos Peixe e Frango. Fazendo uma relação com o desenvolvimento de software, os Objetos Básicos são os cortes ou a qualidade da carne, e o Agrupamento de Objeto são os animais de onde a carne é retirada, e também o tipo de carne que cada um possui, no caso do Peixe e Frango que possuem a mesma característica por serem carne branca.

Na Gerência de Configuração os agrupamentos são comumente utilizados para organizar e facilitar a manutenção de Artefatos do sistema, em grande parte agrupam itens de configuração do mesmo tipo, e fazem com que eles sejam tratados como um bloco único, além da possibilidade de definir comportamentos ou funcionalidades do sistema, a seguir serão listados os exemplos de agrupamentos dentro do projeto de desenvolvimento de software:

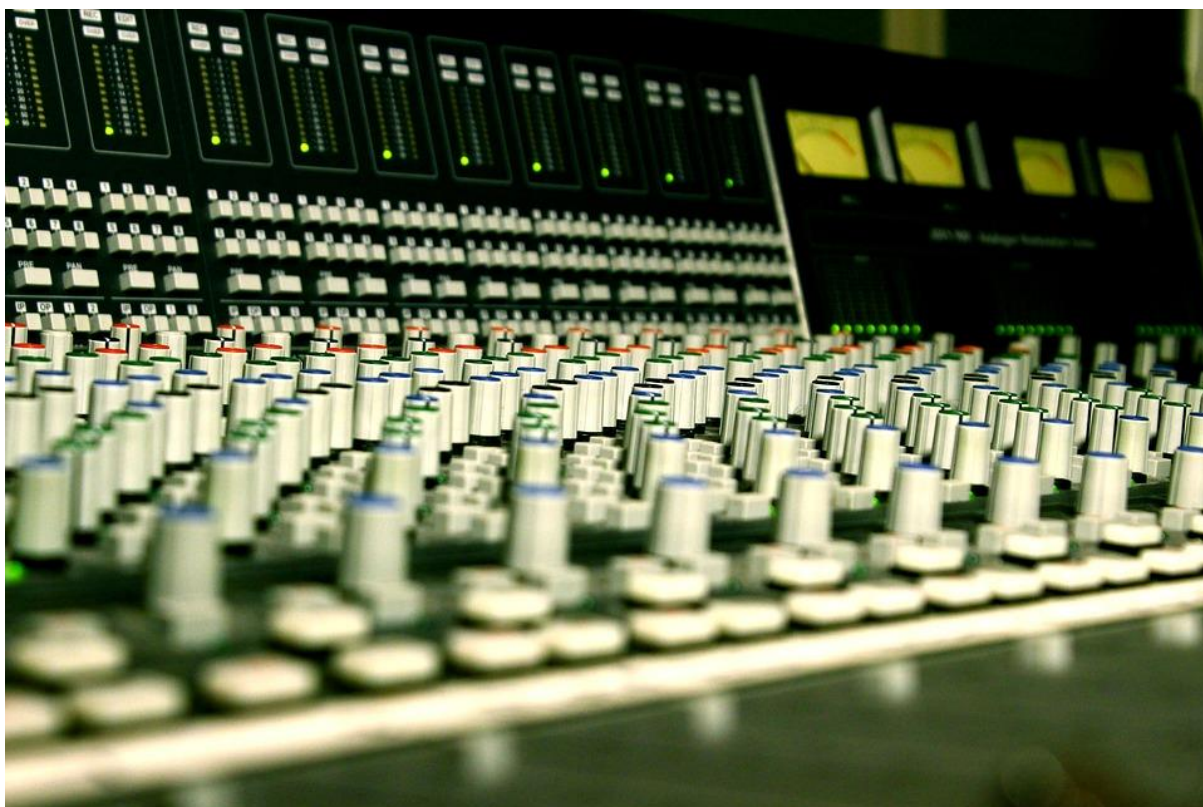
- **Grupo: Sistema**
 - Grupo: Código Fonte
 - Grupo: Usuários
 - Arquivo de Código: CadastroUsuario;
 - Arquivo de Código: ExclusaoUsuario;
 - Arquivo de Código: AlteracaoUsuario.
 - Grupo: Produtos
 - Arquivo de Código: CadastroTipoProduto;
 - Arquivo de Código: CadastroUnidadeProduto;
 - Arquivo de Código: CadastroProduto;
 - Grupo: Requisitos
 - Requisito: Definição de Usuário;
 - Requisito: Definição de Tipo de Produto;
 - Requisito: Definição de Unidade de Produto;
 - Requisito: Definição de Produto.
 - Grupo: Banco de Dados
 - Script: Criação do Banco;
 - Diagrama: Entidade e Relacionamento;
 - Documento: Configurações do Banco de Dados.
- **Grupo: Funcionalidade Pedido de Venda**
 - Grupo: Requisitos
 - Requisito: Pedido de Venda
 - Grupo: Código Fonte
 - Grupo: Pedido de Venda

- Arquivo de Código: CadastroPedidoVenda
- Arquivo de Código: CadastroPedidoVendaProduto
- Arquivo de Código: ExclusãoPedidoVendaProduto
- Arquivo de Código: EdicaoPedidoVendaProduto
- Grupo: Banco de Dados
 - Script: Criação de Pedido de Venda;
 - Script: Criação de Pedido de Venda Produto;
 - Diagrama: Entidade e Relacionamento com Pedido de Venda e Pedido de Venda Produto;
 - Documento: Configurações do Banco de Dados.

Como foi apresentado na lista anterior, os maiores agrupamentos são o Sistema e a Funcionalidade Pedido de Venda, neste caso podemos considerar que o Pedido de Venda faz parte do nosso Software e por este motivo os Artefatos listados neste grupo estão inseridos no Sistema, porém quando forem disponibilizados para o cliente, caso o objetivo for somente disponibilizar a Funcionalidade Pedido de Venda, é possível enviar somente o necessário, evitando disponibilizar um arquivo desnecessário que não têm relação com a entrega e possa gerar algum conflito.

Características de Artefatos

Segundo HASS “o propósito da atividade de identificação é determinar os metadados para um item de configuração”, ou seja, após definir qual é o nível de granularidade do projeto, quais são os Objetos Básicos e seus possíveis Agrupamentos, mesmo assim existe a possibilidade de que dois Artefatos possuam o mesmo nome, e com isto a alteração necessária seja realizada no errado, pode ocorrer a troca no momento da publicação, e também o vínculo de um item de configuração que não é necessário. Por isto Anne Hass fala da definição de “metadados”, que segundo o Dicionário de Oxford é o conjunto de dados que descreve e dá informações sobre outro dado, é o momento que utilizamos Artefatos para identificar Artefatos, para não entrarmos em uma referência cíclica, pois sempre será necessário um Item de Configuração pai, o que gera uma dependência infinita, a maneira mais simples de concluir a Identificação dos Artefatos é definindo as Características ou Propriedades que fazem eles serem únicos dentro do nosso projeto.



Quando algo é característico, quem possui aquela determinada Característica se difere dos outros, como por exemplo a nossa impressão digital que cada um possui a sua, e é possível identificar alguém por ela, por exemplo em uma cena de crime, da mesma forma os Parâmetros são configurações realizada para atender a alguém em específico, ou alguma necessidade, como no caso de uma mesa de som, onde cada entrada têm suas parametrizações de áudio para se adequar ao conjunto, e são muitas configurações possíveis para ajustar o som de diversas formas. Assim também é na Identificação dos Artefatos, onde é elencada uma característica irá diferenciar aquele Item de Configuração de forma única dentro do projeto, para isto são utilizadas convenções que serão seguidas pelo time de desenvolvimento para que um padrão de identificação seja seguido, segundo HASS, na maioria dos casos uma convenção geral não é o suficiente, precisamos de um grupo de convenções para os tipos distintos de Artefatos, como requisitos, código, planos de testes, evidências de testes, entre outros, que relacionando com a mesa de som, são cada um dos botões ajustados para cada Item de Configuração, onde são definidos os Parâmetros.

A forma utilizada pela Gerência de Configuração para Identificar seus Artefatos de forma única são os metadados definidos anteriormente, onde além do nome único dentro da estrutura, também são armazenados os demais itens definidos na convenção, como versão, tipo de codificação, data de criação, data de alteração, usuário de criação, usuário de alteração, e quantos dados necessários para que o controle destes arquivos seja realizado

com qualidade. As ferramentas utilizadas para o Controle de Versão e Controle de Mudanças, realizam o apoio a esta atividade, e serão apresentadas em seus respectivos capítulo, como estamos tratando conceitualmente deste item, o controle manual pode ser feito por um arquivo de apoio, ou pela inclusão dos itens da convenção dentro do arquivo como apresentado a seguir:

Item	Informação
Identificador	REQ-PEDIDO DE VENDA
Tipo do Arquivo	Requisito
Usuário de Criação	Analista de Negócios
Data de Criação	10/10/2010
Usuário de Alteração	Analista de Negócios
Data de Alteração	20/11/2011

Item	Informação
Identificador	CadastroPedidoVenda
Tipo do Arquivo	Código Fonte
Usuário de Criação	Desenvolvedor
Data de Criação	10/11/2010
Usuário de Alteração	Desenvolvedor
Data de Alteração	02/12/2012
Requisito Relacionado	REQ-PEDIDO DE VENDA

Considerações Finais

O controle das alterações realizado pela Gerência de Configuração só é possível caso os objetos que compõem o sistema sejam definidos, e Atividade de Identificação dos Artefatos é responsável por realizar esta descoberta dos Itens de Configuração. Os processos envolvidos na Engenharia de Software geram como resultado diversos Objetos, e como citado anteriormente nosso software não é composto somente do código fonte, existem diversos outros produtos criados durante o ciclo de vida do projeto, que dentro da Gerência de Configuração são chamados de Artefatos, ou Itens de Configuração, e a partir deles é possível definir o sistema e também suas funcionalidades.

Para facilitar a distinção de cada um destes itens, eles são separados entre Objetos básicos e Agrupamento de Objetos, o primeiro é a menor unidade existente dentro de um sistema, que têm o grau de granularidade dependente das características do projeto, que vai definir se o Objeto Básico é um trecho do código ou um tópico do requisito, até no grau máximo onde será o software como um todo ou o banco de dados. Com esta definição é possível verificar o vínculo entre estes Artefatos possibilitando seu tratamento em conjuntos, o que facilita a visualização do objetivo final de cada um deles.

Ao vincular cada um destes Itens de Configuração geramos o Agrupamento de Objetos, ou Objetos Agrupados, onde não só os Artefatos são ligados mas também os grupos, como no caso de módulos de um sistema onde cada parte é responsável por realizar um processo em específico como: Cadastro de Clientes; Cadastro de Produtos; Cadastro de Pedidos; Controle de estoque. O agrupamento de cada uma destas partes gera um Objeto Agrupado que é o nosso software, e consecutivamente cada um deles é composto por um conjunto de Artefatos que fazem são responsáveis pela execução de suas funcionalidades.

Com os Itens de Configuração definidos para nosso projeto, é necessário a criação de informações externas a eles que facilita sua identificação de forma única para evitar a ambiguidade ou a troca destes Artefatos, para isto geramos metadados, ou como definimos as Características, que devem seguir as convenções definidas pelo Gerente de Configuração, os dados básicos de utilizados são:

- **Identificador:** não pode ser duplicado dentro do projeto ou da estrutura em que ele está inserido;
- **Tipo do Arquivo:** que define o tipo do artefato gerado pelo time, pode ser um arquivo de código fonte, requisito, roteiro de teste, diagrama de uml, modelagem do banco de dados, entre outros;
- **Usuário de Criação:** armazena o usuário que gerou o arquivo;

- **Data de Criação:** armazena a data de criação do arquivo;
- **Usuário de Alteração:** armazena o usuário que realizou a última alteração no arquivo;
- **Data de Alteração:** armazena a data da última alteração do arquivo.

A Identificação dos Artefatos que farão parte da Gerência de Configuração do nosso projeto, permite a visualização de todos os objetos que representam as funcionalidades do sistema, que podem variar desde um método de um código fonte, até o software como um todo, e também podem ser agrupados para facilitar a contextualização do objetivo de cada uma das partes da aplicação. Para gerar insumos as próximas atividades, é necessária a definição dos metadados ou características que fazem aquele Artefato ser único dentro do projeto e com isto facilitar o tratamento futuro realizado pelas próximas atividades citadas a seguir.

Controle de Versão

Introdução

Após identificarmos os Artefatos que necessitam de controle durante o processo de Gerência de Configuração, precisamos de uma atividade ou ferramenta que permita o gerenciamento da evolução destes itens enquanto eles sofrem as alterações necessárias durante o ciclo de desenvolvimento do software. Para que este controle seja realizado executamos a atividade de Controle de Versão, que é responsável por monitorar cada modificação realizada nos Artefatos, e garantir que elas sejam identificadas pela demarcação de cada versão com uma numeração ou até pela inclusão de uma descrição específica para cada um destes estados, estas marcações são utilizadas para possibilitar a construção de uma versão estável do sistema, ou para a entrega de uma funcionalidade específica, onde são selecionadas via identificador as versões de cada um dos Itens de Configuração e com isto é realizado seu agrupamento para gerar o software como um todo.



De acordo com Pressman o Controle de Versão se resume a capacidade de realizar as seguintes tarefas:

1. Armazenar todas as configurações relevantes dos objetos em um repositório ou banco de dados do projeto;
2. Gerenciar as Versões armazenando todas as configurações do Objeto, permitindo que qualquer versão seja construída com base nas diferenças das versões anteriores;
3. Facilitar a seleção das configurações relevantes dos objetos e construir a versão específica do software.

Com base nestes tópicos é possível visualizar que o controle da evolução de cada um dos Objetos ou Artefatos do Software é necessário para que seja possível disponibilizar o Sistema como um todo para nossos usuários, onde é necessário armazenar cada uma destas configurações, permitir que elas sejam recuperadas ou realizar a comparação entre as versões, além de viabilizar o agrupamento de cada uma destas versões compatíveis para que seja construída uma versão do sistema.

O Controle de Versão é a atividade responsável pela gerenciamento das Configurações ou Versões dos Artefatos do Software, para que seja possível armazenar cada um destes estados é necessário o armazenamento dos dados complementares ao Item de Configuração para que seja possível realizar este controle, onde além do Artefato os seus metadados são versionados para compor um Versão, para que seja possível realizar o controle das configurações é necessária a execução das tarefas do Controle de Versão, onde é realizado o armazenamento dos objetos em um repositório, para que todos os envolvidos no projeto tenham acesso, a extração destes arquivos é realizada para que as alterações sejam realizadas no ambiente local e não diretamente nos arquivos em produção, e também seja possível receber as alterações realizadas por outros desenvolvedores, caso aconteça algum conflito nestas alterações a mistura entre as novas alterações e as alterações locais para assimilar as mudanças que ocorreram, armazenar no ambiente local as alterações para que elas não sejam perdidas, e ao concluir uma alteração o envio da mesma para o repositório.

Após realizar todas as tarefas do controle de versão e realizar a identificação das versões dos Artefatos de acordo com as suas funcionalidades é necessária a realização do agrupamento e seleção da configuração dos Objetos para que seja gerada uma versão do sistema, o mapeamento dos Itens de Configuração e suas respectivas Versões é chamado de Baseline. Para facilitar a realização tanto das Tarefas quando da geração da Baseline do Sistema, as ferramentas de Controle de Versão complementam a estruturação do processo para a execução desta atividade, utilizando os conceitos aliados a ferramenta que atenda as necessidades do projeto conseguimos otimizar o esforço demandado para a execução desta atividade e amenizar o impacto gerado pela concorrência entre desenvolvedores.

Versões de Artefatos

Considerando que a mudança faz parte do processo de evolução de um sistema, e que a cada etapa do desenvolvimento os Artefatos são criados e alterados para atender as necessidades ou solicitações do cliente, com a definição e agrupamento dos Itens de Configuração de acordo com os objetivos e restrições do projeto, precisamos garantir que cada um dos membros da equipe consigam atuar ao mesmo tempo, e caso seja necessário no mesmo Artefato, além de disponibilizarmos o acesso a cada uma destas mudanças realizadas para visualizar a evolução de cada um destes itens, caso seja necessária à análise de algum erro gerado, ou a restauração de algum estado anterior para resolver de imediato algum impacto gerado durante o desenvolvimento do nosso sistema. Na Identificação dos Artefatos precisamos definir quais entidades do sistema serão controladas durante a sua evolução, no Controle de Versão realizamos o gerenciamento destas alterações para que seja possível acompanhar os estados estáveis destes Artefatos, ou seja, as versões geradas pela equipe de desenvolvimento para atender as necessidades do cliente.

Quando damos os primeiros passos no desenvolvimento de software, normalmente salvamos o primeiro projeto em qualquer pasta do nosso computador, criamos diversas funcionalidades e as alteramos diversas vezes para consolidar os conceitos estudados, não temos preocupação nenhuma em organizar estas informações, então surge a idéia de um aplicativo que vai resolver um grande problema, pensamos em um novo facebook, instagram, spotify, e tantos outros sistemas que deixaram seus donos milionários, então neste momento o desenvolvimento se inicia de uma forma mais organizada, construímos um diretório elaborado, e começamos a utilizar os conceitos da Engenharia de Software para desenvolver nosso sistema, e armazenando dentro do nosso computador. Durante um certo tempo é possível controlar aquele projeto, porém conforme as novas funcionalidades vão ficando prontas e grandes alterações são realizadas, o banco de dados é alterado, os comportamentos esperados para algumas funções mudam e o sistema começa a apresentar erros, como não temos controle para identificar quais suas origens, e não é possível reverter as alterações pois já foram salvas, gastamos mais tempo realizando correções do que criando as funcionalidades que farão do sistema um sucesso. Este cenário parece caótico e nesta situação em específico temos somente um desenvolvedor, no caso do compartilhamento do código em rede e mais desenvolvedores atuam ao mesmo tempo nele, o volume de alterações é bem maior e são realizadas direto no sistema, então cenários de sobreposição, exclusão, impactos em demais partes do sistema são mais comuns.



Segundo o SWEBOK “A versão de um item do software é uma instância identificada deste item”, ou seja, quando concluímos a alteração de um Artefato precisamos marcar aquele estado para diferenciar da situação anterior daquele objeto, e cada uma destas identificações são definidas como Versão e nomeadas para realizar o rastreamento. A evolução de um sistema se traduz em seus Itens de Configuração, porém nem todas as alterações realizadas durante o desenvolvimento de uma funcionalidade precisam ser controladas, na maioria dos casos precisamos comparar que foi alterado quando a implementação foi finalizada, é neste momento que identificamos ou versionamos um Item de Configuração. Pelo exemplo apresentado é possível identificar que estas alterações precisam ser controladas, como acabamos de definir cada modificação realizada nos Itens de Configuração precisa ser identificada ou nomeada pelo desenvolvedor, a fim de que seja possível o rastreamento das alterações, para isto a Gerência de Configuração executa a Atividade de Controle de Versão, que em síntese é o controle de cada versão ou estado dos Artefatos do sistema, uma forma de solucionar ou facilitar a implementação do nosso sistema é realizar um backup manual antes de realizar toda a alteração, ou seja, antes de começarmos o desenvolvimento de uma funcionalidade guardamos todos os arquivos em um local separado e definimos uma versão ou nomenclatura única para eles, e com isto garantimos que caso esta mudança gere erros, temos a possibilidade de comparar os arquivos que foram alterados com os atuais, ou caso necessário recuperar a última versão estável dos mesmos.

Controle de Versão

Ao armazenar cada uma das alterações dos Artefatos estamos gerenciando a evolução do nosso sistema, e quando marcamos cada uma destas versões conseguimos identificar qual

a correção ou funcionalidade inserida naquele estado do Item de Configuração, e estas informações são inseridas nos metadados desses objetos para que o acesso e a visualização desta evolução seja possível, e facilitar a tomada de decisão no caso da identificação de um erro, ou a avaliação do custo para a inclusão de uma nova funcionalidade, pois é possível se basear em quais Artefatos foram alterados para a entrega de uma implementação ou correção relacionada.

O Controle de Versão é responsável por realizar o processo citado anteriormente, onde um cópia de cada Item de Configuração do sistema, independente de ter sofrido alguma alteração ou não durante o desenvolvimento, é armazenada em um local separado do ambiente de desenvolvimento, gerando uma base para a comparação entre a versão atual e o backup realizado, e permitindo também a restauração dos Itens de Configuração específicos caso seja necessário. Ao realizarmos o procedimento citado anteriormente resolvemos parte do problema, pois conseguimos garantir que as versões estáveis dos Artefatos estão seguras durante o desenvolvimento de uma função para o sistema, porém o controle destes backups pode demandar muito esforço, existe a possibilidade de esquecermos de realizá-lo, ou invertemos estas versões, e no caso do trabalho em equipe onde cada membro utiliza uma cópia local, como vamos realizar o gerenciamento para garantir que todos estejam atuando na mesma cópia da versão, ou recebam as alterações geradas pelos outros integrantes da equipe e até armazenar cada umas destas versões sem impactar na execução do projeto, nesta situação este controle se torna inviável.



Como já sabemos a Gerência de Configuração não têm um único responsável, mas é de responsabilidade de toda a equipe, e no caso do Controle de Versão cada um dos membros do time precisa realizar seu gerenciamento de cada alteração realizada, pois precisamos incorporar cada uma delas ao backup estável do sistema sem que haja qualquer conflito, além da possibilidade de sermos os únicos a ter contexto da demanda, o que dificulta a atuação de outro membro do time para realizar o ajuste caso seja necessário. Para que a execução manual desta atividade não seja um caos, precisamos seguir algumas tarefas que servem de guia para direcionar este controle, são elas: Armazenar; Extrair; Guardar; Enviar; Misturar, que quando executadas dentro do seu objetivo fazem com que as Versões sejam armazenadas e recuperadas sem grandes impactos para o restante do time. A seguir será apresentado cada uma destas tarefas e suas respectivas definições:

1. Armazenar (Store/Storage)

Quando guardamos ou gravamos algo é para que possamos encontrar ou reproduzir aquilo novamente, no Controle de Versão armazenamos os Artefatos do sistema para que ele possa ser replicado em outras instâncias ou locais, pode ser enviado para o servidor de homologação ou produção onde o usuário conseguirá acessar o sistema, ou para o computador de um desenvolvedor, onde ele irá atuar na implementação de uma funcionalidade, é importante lembrar que em nenhum momento quando executamos a Gerência de Configuração atuamos diretamente nos dados armazenados, nós utilizamos as

outras tarefas do Controle de Versão para que nossas alterações sejam assimiladas pela versão estável, evitando interferências diretas na aplicação.



O local onde armazenamos o nosso sistema chamado de Repositório pela Engenharia de Software, e ele é responsável pelo gerenciamento das demais tarefas do controle de versão, uma boa representação desta estrutura de armazenamento é a fita cassete, representada pela imagem acima, é basicamente um fita magnética utilizada para armazenar trilhas de áudio, as músicas ou sons eram reproduzidos diversas vezes na sequência que foram gravados, e caso seja necessário armazenar algum outro áudio na fita caso ela não possua espaço livre é necessário o sobrepor algum outro já existente, e para realizar este procedimento precisamos ter muito cuidado pois podemos cortar parte de outra música. Da mesma forma nosso Repositório será replicado diversas vezes, porém o envio de alterações para ele deve ser realizado de forma cuidadosa para que não gere impacto nas demais funcionalidades e comportamentos do sistema, ou gerem um erro dentro da aplicação.

2. Extrair (Pull/Get)

Com nossos Artefatos armazenados no Repositório precisamos de um meio para realizar uma cópia dos mesmos em nosso ambiente local ou atualizar estes itens com as alterações realizadas pelos outros membros da equipe, esta tarefa é chamada de Extração ou Extrair, onde recebemos uma cópia de cada um dos Itens de Configuração que sofreram alteração ou foram criados desde a última vez que puxamos estas informações, o controle de quais arquivos serão enviados é realizado de acordo com o controle interno do Repositório, que utiliza os metadados dos Artefatos como base para esta definição, além da comparação com a nossa cópia local, onde será verificado quais arquivos não existem e precisam ser criados, e quais dados foram inseridos ou removidos dos itens para que eles fiquem de acordo com a versão mais recente.



Ao Extrair os dados do Repositório, este comportamento é chamado de Pull o Get em algumas ferramentas de Controle de Versão, estamos literalmente puxando estes dados para dentro de nossas cópias locais, este controle é realizado linha a linha pelo repositório, quando é possível caso contrário os arquivos como um todo são substituídos, onde a cada nova versão armazenada no Repositório verificado quais partes do Artefato foram alteradas, removidas ou criadas, e este mesmo processo é replicado em nosso ambiente quase que simulando este comportamento. Como existe a possibilidade da versão local possuir divergências nos pontos que serão aplicados pela extração, contamos com a tarefa de Mistura, que será abordada na sequência, para lidarmos com esta risco que é algo corriqueiro quando estamos desenvolvendo este software em uma equipe grande e atuamos na mesmas funcionalidade ou em necessidades relacionadas.

3. Misturar (Merge)

Como citamos a pouco independente do porte do projeto quanto atuamos com mais de um membro na equipe, e até mesmo quando atuamos com um só, os Artefatos que são compartilhados entre as demandas em implementação podem ser alterados ao mesmo tempo e exatamente no mesmo ponto, ao Extrair estes dados para nossa cópia local somente a substituição destas linhas por vigentes não resolve esta situação, e existe a possibilidade desta sobreposição limpar dias, semanas ou meses de desenvolvimento, o que pode custar um projeto inteiro, por uma única linha removida erroneamente. Para isto a tarefa da Mistura ou Merge, é realizada para garantir que em casos onde existam estes conflitos de informação, tenhamos uma forma de resolvê-los com base em nossa avaliação. Ao Extrair os dados do Repositório é realizada a verificação dos pontos de substituição, e caso eles estejam na mesma Versão onde a alteração foi armazenada a substituição é realizada sem críticas, porém caso não esteja esta parte é marcada como conflito e após todas as verificações serem realizadas, um relatório de pendências é apresentado para tratamento.



Se compararmos esta tarefa com a mistura de um drink, é preciso medir as quantidades de xarope, suco, a bebida alcoólica, açúcar, água com gás, tônica, enfim existem infinitudes de combinações possíveis para criar diferentes bebidas, e misturar a quantidade correta vai fazer o drink chegar ao sabor ideal, e não ficar muito doce ou muito alcoólico. O mesmo acontece no tratamento destas críticas durante o recebimento de uma nova versão dos artefatos, pois precisamos combinar corretamente nossos dados para que as funcionalidades do Repositório continuem funcionando, e também para não impactar a nossa implementação, esta etapa é chamada de diferenciação, ou diff, onde é possível visualizar não só os itens em conflito mas também o que foi inserido e removido na última versão, e nesta situação três tratamentos realizados:

- **Assumir as mudanças do Repositório:** Onde definimos como válidas as mudanças Extraídas e permitimos a sua aplicação;
- **Manter as alterações locais:** Neste caso nossas Versões locais são mantidas, e com isto as mudanças do Repositório são descartadas;
- **Avaliar conflito a conflito:** Este tratamento é realizado de forma manual onde cada um dos conflitos é avaliado pela visualização da Versão local e recebida do Repositório, e a partir desta verificação é definido ponto a ponto qual fonte será utilizada.

4. Guardar (Commit)

Quando Controlamos as Versões dos Artefatos precisamos garantir que a nossa implementação em desenvolvimento esteja devidamente armazenada, pois estamos sujeitos a ações externas a nossa funcionalidade, independente se estamos trabalhando somente em nosso computador ou em um Repositório de desenvolvimento. A evolução do sistema que ocorre enquanto executamos nossa implementação é a principal ação externa, e a assimilação dessas mudanças em nosso ambiente com o Merge é a principal ferramenta para que estas alterações sejam inseridas em nosso ambiente, supondo que neste tratamento sobrepomos alguma funcionalidade que já havíamos implementado, ou todo o nosso trabalho, a solução é refazer todo o desenvolvimento gerando custos desnecessários para o projeto.

Ao realizar a tarefa Guardar, mais chamada de Commit pelas ferramentas de Controle de Versão, armazenamos as alterações realizadas em nosso ambiente local sem enviar para o Repositório do sistema, ou seja, conforme codificamos uma funcionalidade, as alterações geradas são armazenadas primeiramente em nosso controle local ou em um local reservado, e após enviado para o nosso sistema.



Um das definições de Commit segundo o dicionário de Oxford é “Transferir alguma coisa para um estado ou lugar, onde possa ficar guardado ou preservado”, o conceito de Guardar dentro da Gerência de Configuração está diretamente relacionado com esta descrição, como em um cofre onde deixamos seguros nossos pertences para que não sejam roubados, ou guardamos coisas que para proteger o ambiente externo, como na figura acima onde as bebidas estão a salvo pelo seu alto valor, ou estão guardadas para evitar que sejam consumidas por alguma criança, dentro da Engenharia de Software o commit não só protege a implementação que está em execução, mas também resguarda o sistema para que não seja afetado por uma implementação que ainda não está finalizada e pode afetar o sistema em produção ou a implementação de um outro desenvolvedor.

5. Enviar (Push)

Uma vez que estamos realizando o desenvolvimento, armazenando aos poucos cada mudança e inserindo a evolução do sistema em nossa implementação, quando vamos finalizar a implementação e disponibilizar a funcionalidade para o cliente, neste ponto após concluir todas as alterações realizadas é necessário inserir estas alterações na versão estável do sistema, porém precisamos de muito cuidado neste processo, pois além de garantir que a nossa implementação atenda a demanda do cliente e não possua erros, ela não pode impactar o sistema ou sobrepor alguma funcionalidade já existente. Como em nossa versão local, no Repositório do sistema utilizamos o controle por Commit para acompanhar a evolução dele, e neste ponto de disponibilização utilizamos a funcionalidade do Merge para que as alterações possam ser inseridas sem problemas e caso aconteça algum conflito o mesmo possa ser resolvido pelo responsável da implementação que possui mais contexto que os demais.



Como quando enviamos uma carta, precisamos colocar o destinatário, endereço completo, cidade, estado, país e CEP, além dos mesmos dados para o remetente, o conteúdo da carta precisa estar bem redigido, para quando ela for recebida não seja gerado ambiguidade ou interpretações erradas e esteja de acordo com o assunto tratado, ao enviar uma implementação para a versão de produção do sistema, precisamos garantir que o que foi implementado esteja de acordo com o sistema, que as funcionalidades estejam de acordo com os princípios de reutilização e padrões de projetos utilizados, além de funcionarem e não afetarem os desenvolvimentos já realizados e utilizados pelo usuário.

Baselines

As demandas ou solicitações do cliente normalmente são o que motivam a realização de alterações no sistema, o Controle de Versão é responsável pelo gerenciamento dessas alterações fazendo com que elas sejam realizadas sem impactar as funcionalidades pré-existentes no sistema, e também sem afetar as alterações em desenvolvimento pelos demais membros da equipe. Cada uma dessas necessidades têm uma data limite para ser entregue, que normalmente é definida pela equipe com base na sua complexidade além de considerar a expectativa de entrega do cliente, para atender a este prazo realizamos a paralelização da implementação, ou seja, mais de um desenvolvedor atuando ao mesmo tempo no desenvolvimento de funcionalidades distintas, e este volume de alterações gera os seguintes impactos:

- Como identificar se a última versão de cada Artefato representa a funcionalidade a ser entregue?
- As alterações realizadas foram testadas e atendem ao padrão de qualidade?
- O atendimento da definição de negócio foi validado?

Para possibilitar a disponibilização destes objetos precisamos de uma estrutura de controle para eles e também seus respectivos agrupamentos, onde a entrega contenha não só as versões estáveis dos Itens de Configuração mas contemple as funcionalidades esperadas para aquele momento, permitindo o funcionamento do sistema e o atendimento aos requisitos

vinculados à necessidade do cliente sem que seja entregue algo a mais que possa afetar a utilização do software.



Durante a execução das tarefas de Controle de Versão além do armazenamento de cada estado das alterações dos Artefatos, os seus respectivos metadados também são versionados e armazenados no repositório, com isto temos a possibilidade de rastrear e avaliar a evolução individual de cada um destes itens além de identificar os usuários responsáveis por cada uma das mudanças e datas de realização das mesmas. Segundo o SWEBOK “A Baseline do Software é uma versão formalmente aprovada de um item de configuração que é formalmente designada e fixada em um momento específico durante o ciclo de vida de um item de configuração.” como na organização de documentos, onde agrupamos separadamente em pastas onde definimos uma classificação, seja ela por nome, contexto, objetivo, data, isto também no Controle de Versão, onde para garantir que um Item de Configuração atenda ao requisito é necessária a realização dos testes do sistema e validação do cumprimento desta definição, com estas etapas devidamente realizadas e com o Artefato aprovado em cada uma delas é necessário realizar a marcação da versão específica para que a mesma componha o conjunto de itens que serão entregues ao Cliente. No contexto da Gerência de Configuração a Baseline é utilizada de diferentes modos, porém os mais importantes são para controlar os Objetos de forma individual e também para controlar o conjunto de Itens de Configurações que compõem a versão que atende determinado requisito do sistema, no caso dos Artefatos o controle é realizado de forma individual, porém quando falamos de um software como um todo diferentes Baselines são geradas para mais de um Itens de Configurações compondo uma configuração específica do sistema.

Ferramentas

O Controle de Versão é uma atividade vital para que um projeto seja executado com sucesso, e a ausência desta atividade pode fazer que nossa tentativa de construir um sistema seja um fracasso. Segundo Pressman “O controle de versão combina procedimentos e ferramentas para gerenciar diferentes versões dos objetos de configuração que são criados durante o processo do software.”, com o entendimento dos conceitos necessário para que seja possível aplicar esta etapa do processo durante a implementação da nossa aplicação é possível realizar o gerenciamento dos Artefatos, porém cada uma das tarefas demanda muito esforço se realizada de forma manual, por este motivo existem ferramentas que nos auxiliam no Controle de Versão, fazendo com que a maioria destas atividades sejam realizadas de forma automática, e somente quando o conjunto de ações pré definidas não consegue resolver os possíveis problemas sozinho. Como em grande parte dos sistemas da área de TI, existem soluções pagas e também soluções gratuitas que são mantidas pela comunidade, isto nem sempre quer dizer que as soluções de código aberto sejam piores que as pagas, cada uma delas soluciona um problema diferente e cabe a você como Gerente de Configuração definir qual a melhor solução para o seu problema ou para a sua necessidade.



Ferramentas de Código Aberto

As ferramentas de código fonte aberto, ou ferramentas gratuitas, normalmente foram criadas por um grupo de pessoas para resolver algum problema específico sem fim comercial, e por isto tanto a ferramenta quando seu código fonte são compartilhados para que os demais possam utilizar ou contribuir de forma gratuita, o grupo de pessoas que apóiam e dão suporte para este tipo de sistema são chamadas de comunidade e garantem a evolução dos sistemas para ajudar o maior número de pessoas possível, e fazer com que não só as soluções pagas sejam uma alternativa, principalmente para gerar uma alternativa para aquelas empresas, equipes ou desenvolvedores individuais que não tem condições de bancar uma ferramenta paga. Ainda sim por estar sempre em contato com seus usuários e ser mantida, receber

atualizações, sugestões de melhorias e suporte por eles, acaba se tornando um padrão de mercado pela quantidade de usuários que possui fazendo com que seja mais completa que as demais.



Alguns exemplos de sistema de controle de versão ou repositório gratuitos são:

1. Mercurial



O Mercurial é uma das primeiras ferramentas criadas para a realização do controle de versão, segundo O'Sullivan o Mercurial veio a vida em 2005, com foco na facilidade de uso, alta performance e escalabilidade para projetos muito grandes. As principais vantagens em utilizar o Mercurial são:

- Facilidade de aprender e usar;
- É muito leve;
- Altamente escalável;
- É fácil de customizar.

2. Subversion



Segundo Collins-Susman o Subversion foi lançado em 2001 após 14 meses de desenvolvimento, com o objetivo de gerar uma alternativa ao seu predecessor o CVS. Ele realiza o gerenciamento de arquivos e diretórios, além das mudanças realizadas neles durante o tempo. As maiores vantagens do Subversion são a sua performance para o dia-a-dia e seu suporte para o merge entre os branches, que são as ramificações do código-fonte.

3. Git



Criado em 2005 para substituir o BitKeeper, o Git segundo Chacon “É muito rápido, muito eficiente em projetos grandes e possui um sistema incrível de branching (ramificação) para desenvolvimento não linear”, é uma das ferramentas mais utilizadas atualmente pela facilidade de aprendizado e colaboração. As maiores vantagens do Git são:

- Velocidade;
- Design Simples;
- Suporte para o desenvolvimento não linear;
- Totalmente distribuído;
- Capaz de lidar com grandes projetos como o kernel do Linux com eficiência.

Ferramentas Pagas

Pela criticidade da execução da Atividade de Controle de Versão algumas empresas visualizaram esta necessidade como uma oportunidade de negócio e criaram sua própria ferramenta para a comercialização, inclusive o surgimento de ferramentas gratuitas como no caso do Git e do Mercurial se deu pelo início da comercialização de um software que era gratuito, que no caso foi a remoção da licença open source do BitKeeper. O motivo que leva a seleção de uma ferramenta paga é o suporte e a garantia prestados pela empresa que o comercializa, pois ela é responsável por dar total apoio a utilização e garantir que o sistema funcione, o caso algum erro seja encontrado disponibilize a correção, e até em alguns casos disponibilizar um customização para que o sistema se adapte a realidade do cliente. Neste ponto a utilização de um software gratuito têm seus riscos, pois caso ele contenha um erro você precisa do apoio da comunidade para resolver, e nem sempre sua necessidade será priorizada, ou você mesmo precisa realizar aquela solução, o que pode fazer com que a sua versão perca a compatibilidade com as próximas versões disponibilizadas, o que irá impedir uma atualização, ou seja, mesmo que seja um software muito reconhecido, sendo livre quem o utiliza é beneficiado pela ausência de custo porém assume o risco de não contar com um suporte obrigatório, e com isso ficar refém de algum erro que pode impactar todo o projeto.



Alguns exemplos de sistema de controle de versão ou repositório pagos são:

1. Helix Core



Lançado em Julho de 2016, o Helix Core é a ferramenta de controle de versão da Perforce, por ser uma das ferramentas mais recentes utiliza do que deu certo nas demais para gerar uma vantagem de mercado. Dentre as vantagens do Helix Core segundo seu *Product Brief*, que é a descrição do produto, a principal é a capacidade de armazenar arquivos de grande porte, além de disponibilizar uma integração com o Helix4Git que é a plataforma para utilização do Git da Perforce, além de outras integrações e ferramentas que permitem o aumento da produtividade do time de desenvolvimento.

2. AWS CodeCommit



AWS CodeCommit

O AWS CodeCommit é o serviço de hospedagem de repositórios Git lançado pela Amazon em 2015, segundo Jeff Barr o “CodeCommit é um serviço de gerenciamento controle e revisão que hospeda repositórios Git e funciona com ferramentas baseadas em Git.” a ferramenta estabelecida e gratuita, no caso o Git, é vendida em conjunto aos serviços de hospedagens da Amazon já reconhecidos pelo mercado. As maiores vantagens do serviço são a alta disponibilidade, onde a infraestrutura utilizada reduz a possibilidade de indisponibilidade, e é totalmente gerenciado, não é necessária a contratação ou compra de um servidor e configuração o serviço já está disponibilizado com toda esta estrutura.

3. TFS

A Microsoft por sua vez lançou um sistema de controle de versão chamado Team Foundation Version Control, TFVC, em conjunto ao Team Foundation Server, TFS, em 2013, que é um serviço similar ao AWS CodeCommit, porém onde só era possível a utilização da ferramenta



de controle de versão da Microsoft, além da ferramenta também disponibiliza a estrutura para armazenar estes dados além de um conjunto de outras ferramentas Microsoft que são adquiridas em uma licença única. Atualmente o TFS se chama Visual Studio Team Services, VSTS, e permite que seja utilizado o repositório em Git.

Como é possível visualizar cada uma das ferramentas possuem características específicas que devem ser analisadas de acordo com a necessidade do projeto, o custo pode fazer que a opção mais viável seja uma ferramenta gratuita, ou caso seja necessário um conjunto de serviços e suporte e a contratação destes serviços a parte fique mais cara que a contratação de uma das ferramentas pagas, ou seja, como de costume nas áreas relacionadas à Gerência de Configuração a definição de uma ferramenta ou serviço vai depender da análise e o atendimento às necessidades específicas do projeto, para isto é necessário analisar os prós e contras de cada uma delas, atualmente grande parte do mercado utiliza estas ferramentas listadas acima, o que facilita um pouco esta tarefa.

Considerações Finais

A partir da seleção dos Artefatos que compõem o Software e a definição de seus metadados é possível utilizar estas informações para realizar o Controle de Versão destes itens, esta atividade é responsável por armazenar cada uma das configurações geradas pelas alterações realizadas nestes Itens de Configuração, onde a versão estável fica guardada em um repositório e estes objetos são replicados para que seja possível a realização das modificações sem afetar a configuração armazenada, e quando a implementação ou correção do Artefato for realizada as mesmas são aplicadas na versão do repositório para que a configuração mais recente sempre esteja disponível para todos os integrantes do projeto.

Para organizar o envio e recuperação de arquivos do repositório, o Controle de Versão conta com as suas Tarefas que facilitam esta interação com as versões armazenadas, a primeira é o Armazenamento dos arquivos identificados ou gerados durante o ciclo de vida do sistema, onde eles são guardados no repositório para que seja possível sua distribuição entre os integrantes do time, para que seja possível atuar em uma implementação ou correção sem afetar as versões estáveis realizamos a Extração das informações do repositório, ou seja, realizamos uma cópia local destes arquivos para que seja possível alterar os arquivos deixando o repositório inalterado durante estas modificações, durante o desenvolvimento precisamos Guardar a evolução da nossa cópia local para que seja possível manter um backup independente do que foi enviado para o repositório, e após concluir as implementações é necessário realizar o Envio das alterações realizadas para o repositório

onde elas ficarão armazenadas e disponíveis para os demais membros da equipe, durante todo este processo o repositório recebe novas versões dos artefatos que podem afetar ou facilitar o nosso desenvolvimento, além de impedir que as modificações realizadas sejam enviadas, por este motivo tanto na Extração quanto no Envio a tarefa de Mistura é realizada para que seja realizada a comparação entre as diferenças encontradas nas Configurações dos Objetos, caso não existam conflitos estas alterações podem ser assimiladas sem avaliação, porém se ocorreram alterações no mesmo ponto, será necessária a análise destas divergência para avaliar qual será considerada a versão estável daquele Artefato.

Após gerenciar a evolução de todos os Itens de Configuração do projeto, nem sempre vamos disponibilizar as últimas versões para os clientes, a cada envio para o repositório realizamos uma marcação ou a identificação destes artefatos, para que seja possível diferenciar qual é o objetivo da alteração realizada ou referente a qual funcionalidade ela está relacionada, com estas informações e de acordo com os prazos e datas alinhados com os clientes vamos construir uma versão do sistema, a chamada Baseline do Software que é o agrupamento de diferentes versões dos Artefatos, que relacionadas atendem a um objetivo ou funcionalidade referente à evolução esperada do sistema esperada pelo usuário.

A execução de cada uma das Tarefas do Controle de Versão de forma manual fica mais complexa conforme a quantidade de Artefatos do projeto aumenta e também pelo volume de alterações, para isto foram criadas ferramentas para solucionar e facilitar a realização desta atividade, onde cada uma destas tarefas foi automatizada ou construída para que o repositório receba sempre as últimas versões sem impactar o restante da equipe e também o projeto como um todo. Como em grande parte da área de TI existem ferramentas pagas e também grátis, ou como chamamos de código aberto, que possuem vantagens e desvantagens e precisam ser avaliadas caso a caso para identificar se realmente atendem a situação ou as necessidades do seu projeto em si, e a utilização de uma ferramenta de Controle de Versão é indispensável para o sucesso de um projeto de grande porte.

Controle de Mudanças

Introdução

As mudanças fazem parte da evolução do sistema, e segundo Boller “Mudanças são inevitáveis” precisamos saber gerenciar essas mudanças para que elas não causem o caos dentro do projeto, ou o levem ao fracasso como já foi apresentado pelo CHAOS Report, onde a principal causa do não cumprimento do prazo é o aumento do tamanho do sistema, e mesmo que elas não sejam tão graves podem gerar um incômodo, de acordo com Pressman “Controle de Mudança é vital. Mas as forças que o fazem necessário também fazem isto irritante.”, para que estas alterações não se tornem um aborrecimento dentro do do nosso processo de desenvolvimento de software precisamos realizar a Atividade de Controle de Mudanças onde essas chateações ou problemas são transformados em soluções para o cliente e também para o sistema.



A comunicação via telefone foi um grande avanço tecnológico, a invenção de Alexander Graham Bell foi um marco muito importante para a humanidade, pois facilitou a comunicação entre pessoas, porém como nossa única certeza é que mudanças irão ocorrer não só a forma como nos comunicamos evoluiu como também os telefones, atualmente utilizamos os telefones celulares para nos comunicar, e o telefone convencional em algumas situações se tornou desnecessário. Se o nosso meio de comunicação sofreu alterações e até evoluiu a

ponto de ser descartado, o nosso sistema não sairia ileso e para que estas alterações sejam realizadas de para que o nosso sistema também evolua sem gerar falhas, precisamos gerenciar quais os fatores ou motivações geraram as necessidades de mudança, quais são os resultados esperados para elas e também o prazo para sua conclusão, além de acompanhar a execução e conclusão destas modificações fazendo com que elas sejam entregues ao cliente com qualidade. A realização deste acompanhamento é possível com a execução da Atividade do Controle de Mudanças, que ao invés de realizar o controle dos Artefatos do sistema como no Controle de Versão, gerência a origem, os objetivos e o andamento das alterações ou novas funcionalidades requisitadas pelo cliente, segundo Hass “Um processo de mudança é um projeto de desenvolvimento em miniatura em si” ou seja, é a execução de todos os processos de desenvolvimento descritos na Engenharia de Software porém para um pedaço do projeto, com isso é possível aplicar os principais conceitos da construção do sistema para que as melhorias sejam realizadas com qualidade e sejam entregues ao cliente sem gerar impacto no restante do sistema.



Para que o Controle de Mudança seja realizado com sucesso é necessária a execução das tarefas definidas por esta Atividade, elas estão listadas a seguir:

- **Solicitação:** O cliente apresenta a necessidade de mudança via Solicitação, é necessário que esta ela seja bem descrita, e conte com todas as funcionalidades desejadas com a maior quantidade de informações para que seja possível realizar uma avaliação de qualidade;

- **Análise e Priorização:** Após receber todos os dados da Solicitação, precisamos definir como ela será realizada no sistema, e também quais são as atividades contempladas no desenvolvimento, o prazo de conclusão do desenvolvimento além da prioridade desta atividade em relação às demais, esta tarefa é chamada de Análise e Priorização;
- **Aprovação:** Ao finalizar a Análise e Priorização da Solicitação do cliente, ela é avaliada por integrantes da equipe, e após a sua aprovação interna é disponibilizada para o aceite do cliente;
- **Desenvolvimento:** Com a aprovação por parte da equipe e do cliente é possível iniciar a implementação da Solicitação, e a execução das tarefas do Controle de Versão;
- **Verificação:** Após concluir o desenvolvimento das alterações ou criação de funcionalidades, elas precisam passar pelo processo de Validação onde o que foi desenvolvido é revisado tecnicamente por outro desenvolvedor e após são realizados os testes funcionais para avaliar se as necessidades foram implementadas de acordo com a descrição da Solicitação e a definição realizadas pela Análise.

Nos tópicos a seguir será realizado o detalhamento destas tarefas, e após a apresentação das ferramentas utilizadas para que seja possível realizar o gerenciamento via software do Controle de Mudança.

Solicitação

Cada alteração realizada dentro do projeto é baseada em alguma necessidade, normalmente gerada por algum problema que o cliente enfrenta no dia a dia, ou por uma mudança imposta pelo governo, até mesmo a criação do sistema vêm para resolver estas questões, para que essas demandas sejam inseridas dentre as funcionalidades do software, elas precisam ser enviadas via Solicitação para o time de desenvolvimento, segundo o SWEBOK “A solicitação de mudança é uma solicitação para aumentar ou diminuir o escopo do projeto”, ou seja, é o meio utilizado pelos usuários ou clientes para adicionar ou remover uma necessidade no projeto, também é chamada de *Change Request (CR)*, ou Solicitação de Mudança, para possibilitar a análise e consideração desta alteração é necessário que ele contenha o detalhamento das informações referentes a motivação daquela mudança e a descrição do comportamento realizado pelo sistema, inclusive em um cenário onde sistema ainda não existe estes dados são cruciais para possibilitar o entendimento da expectativa do cliente e quais são as funcionalidades que atendem a esta necessidade.



Em determinados casos estas Solicitações podem ser geradas pela equipe interna de desenvolvimento, pois quando um desenvolvedor ou analista começa a atuar com a regra de negócio do cliente e entendimento das necessidades, normalmente ele começa a ter a visão do todo que muitas vezes o usuário por estar focado em uma parte do processo não consegue visualizar, e com isso surgem sugestões de melhorias propostas pelo time, nesta situação o cliente precisa validar se aquela sugestão está aderente a realidade do cliente e também a regra de negócio, porém estas sugestões normalmente são aceitas pela sintonia que a equipe de desenvolvimento possui com a expectativa e especificidades do sistema.

A seguir serão detalhados os itens básicos que uma Solicitação de Mudança deve possuir:

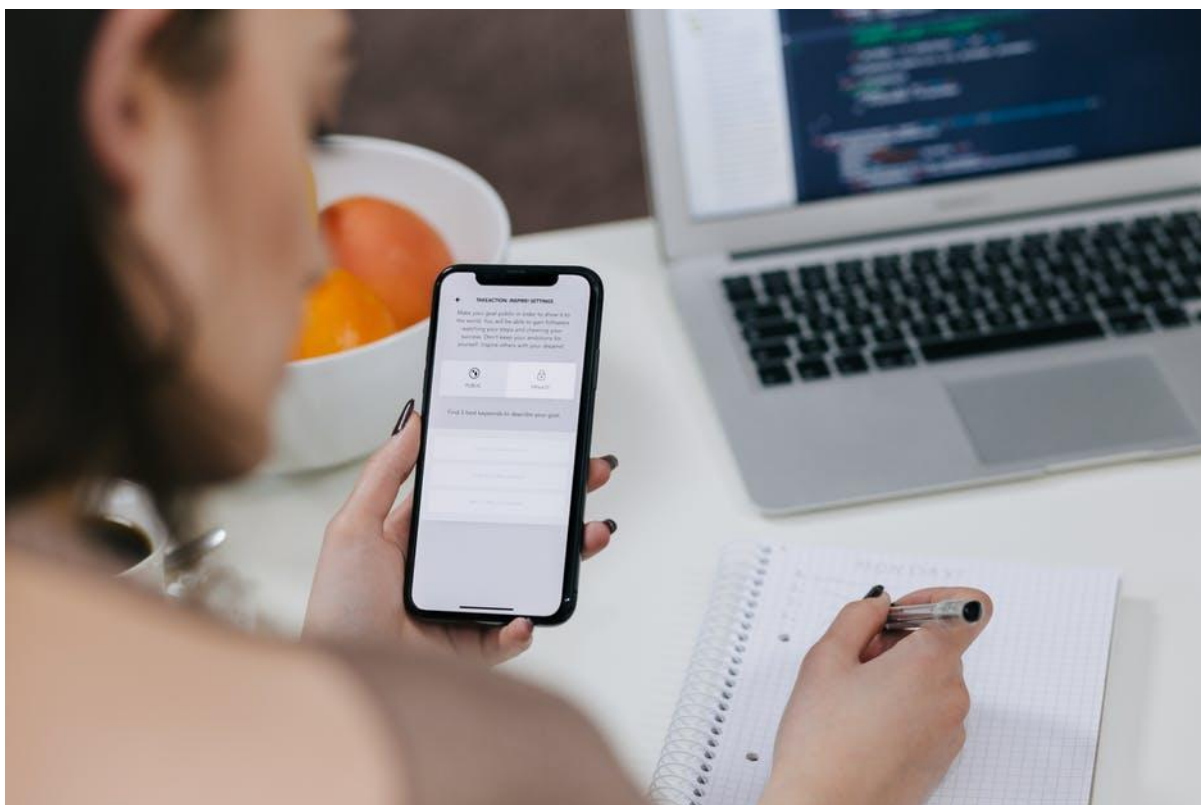
- Responsável pela solicitação:
 - Aqui são apresentadas as informações referentes ao solicitante da mudança, como cargo e empresa, data da solicitação e o nome do responsável.
- Escopo da Mudança:
 - Produtos:
 - Apresentação de quais os produtos que serão afetados ou criados a partir da mudança.
 - Data de Conclusão:
 - Apresentação do prazo com base na expectativa do solicitante que será considerado como base para o dimensionamento de equipe.
- Detalhamento da Mudança:

- É necessário o envio da descrição da mudança como um todo, o que motivou a mudança, qual problema será resolvido com ela, como será o funcionamento esperado, quais as alterações necessárias em outras funcionalidades, entre outros pontos necessários para que seja possível a visualização da funcionalidade como um todo.

A disponibilização destas informações é necessária não só na construção de um sistema ou na inclusão de uma nova funcionalidade, no caso de um erro ou falha de desenvolvimento estes dados são necessários para constatar se foi um requisito não solicitado pelo cliente ou um erro gerado, e qual o caminho para que ele possa ser replicado e resolvido. Normalmente estas informações são menosprezadas pelo cliente e direcionadas para que um Analista de Sistemas ou Negócios realize busque estas informações a partir da indicação do cliente, porém a inclusão e responsabilização do cliente nesta etapa além de facilitar o trabalho realizado nas demais etapas, também gera a sensação de contribuição e responsabilidade, isto involuntariamente aumenta o engajamento dos envolvidos com o projeto e faz com que eles se importem muito mais com aquela solução. Após receber todas essas informações precisamos realizar o trabalho de relacionar o que foi solicitado com a realidade do sistema, este tratamento é realizado pela etapa de Análise e Priorização.

Análise e Priorização

Ao receber a Solicitação é possível avaliar os motivos pelos quais a mudança foi solicitada e o que será necessário realizar para entregar aquela funcionalidade para o cliente, além da avaliação de qual será o esforço necessário para realizar seu desenvolvimento e o cronograma de execução, e a partir destas duas informações é possível calcular o custo ou o valor cobrado do cliente para viabilizar o projeto. Segundo Pressman “A solicitação de mudança é enviada e estimada para avaliar o detalhamento técnico, potenciais efeitos colaterais, o impacto geral nos outros itens de configuração e funcionalidades do sistema, e o custo projetado para a mudança.”, a partir desta definição é possível constatar o quão importante esta etapa é para o sucesso do projeto, pois ela abrange todas as áreas críticas da Engenharia de Software, com base na definição de negócio é realizada a análise técnica, para verificar quais são os Artefatos criados ou alterados e quais serão as modificações necessárias para atender a Solicitação, e com o detalhamento destas informações é possível identificar quais são os impactos gerados dentro do sistema, quais funcionalidades existentes serão afetadas e quais comportamentos terão que ser ajustados, e com isto projetar o custo final de desenvolvimento.



Com o detalhamento técnico, o detalhamento realizado a partir da análise de negócio do sistema, é realizada a definição das Atividades, Recursos Envolvidos e Cronograma de execução:

- A primeira etapa é o detalhamento, ou como normalmente é chamado, a quebra de um escopo técnico em partes menores, neste ponto são extraídas frações da necessidade para que sejam geradas pequenas Atividades, cada uma com um objetivo específico e organizadas de acordo com sua interdependência, ou seja a dependência entre cada uma delas, onde caso uma não seja concluída não é possível realizar as demais, e também pelas suas individualidades.
- Com as Atividades é possível realizar o dimensionamento dos Recursos Envolvidos, ou Recursos Necessários para a execução da mudança, onde é realizado o dimensionamento dos Desenvolvedores necessários para concluir a atividades, e também dos demais papéis necessários como Analistas de Negócio, Qualidade, Testes e Suporte, Arquitetos de Software, Líder Técnico, Gerente de Projetos, e outros papéis como o responsável do cliente que irá dar apoio na construção da solução.
- Após analisar e coletar todas estas informações montamos o Cronograma de Execução, onde definimos em que momento cada atividade será realizada e qual será o recurso responsável por ela, e após elencar os responsáveis e datas para todas as atividades de acordo com a sequência e agrupamentos definidos, é realizada a

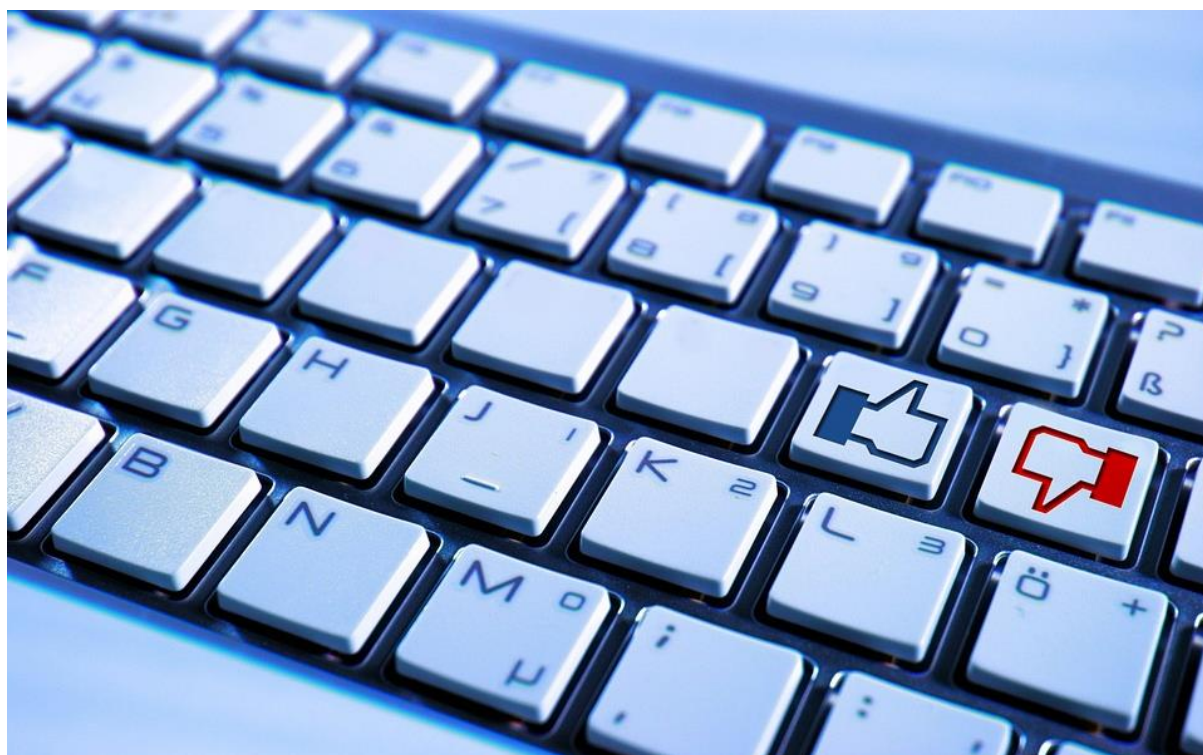
verificação da Data de Conclusão inserida na Solicitação, caso a data do cronograma seja superior a definida pelo cliente será avaliado a possibilidade de redução de escopo, ou seja a remoção de atividades ou a inclusão de novos Recursos para acelerar a execução.



O conceito de elaboração de Cronograma apresentado anteriormente é o mais popular, principalmente em empresas que utilizam o modelo *Waterfall*, cachoeira, de gerenciamento de projetos, onde a execução das atividades é sequenciada com base na disponibilidade dos recursos disponíveis, porém quando relacionamos esta etapa com o problema tratado pela Gerência de Configuração que é a Mudança, precisamos de um framework que tenha um enfoque nesta necessidade, neste caso o modelo *Agile*, ágil, é a metodologia que se popularizou por ser resiliente a mudanças, ou seja, a estrutura de análise seguiria o mesmo processo, porém no momento da elaboração do cronograma, consideramos a capacidade produtiva da equipe, a quantidade de integrantes do time de desenvolvimento e a quantidade de horas calculada para executar as atividades, com base nestes dados é possível avaliar qual é o prazo provável, e neste caso não é apresentado uma data fixa pois a expectativa é que mudanças ocorram durante o processo, e caso surjam novas atividades será trabalhada a priorização das mesmas sobre as que já foram levantadas e com isto definir qual será a estratégia em conjunto do cliente que melhor resolve a situação, o caso mais comum é a remoção de atividades com menor prioridade para atender as mais importantes, e estas restantes serão realizadas em uma outra solicitação de mudança.

Aprovação

Após concluir a Análise e Priorização da Solicitação é necessário que todos os dados gerados a partir dela sejam avaliados, tanto pelos demais membros da equipe quanto pelo cliente ou solicitante, a validação realizada pelo time é voltada para confirmar se as atividades que foram descritas e como foram definidas são possíveis de serem realizadas, avaliar se algum ponto ficou pendente, ou existe algo que precise ser inserido dentro do escopo para viabilizar a execução e revisar o escopo técnico para confirmar se as soluções adotadas são as ideais para aquela situação, já a aprovação por parte do cliente é voltada para as suas restrições, como já foi citado na elaboração do cronograma da Análise a Data de Conclusão que é proposta na Solicitação é um fator que pode ser determinante, em conjunto com o custo para a realização daquela mudança, além do escopo da funcionalidade, pois é necessário que ele concorde com o que será realizado. Caso tanto a equipe técnica quanto o cliente aprove a execução da funcionalidade é dada a continuidade com o desenvolvimento da mesma, caso contrário é necessário ou apresentar os motivos de rejeição e a partir deles é realizada novamente a análise para que todos os pontos sejam ajustados e a atividade de aprovação seja realizada novamente.



Quando a equipe de desenvolvimento avalia a Análise são verificados diversos aspectos relacionados àquela implementação, e caso seja verificado algum motivo que inviabiliza a execução do desenvolvimento a Solicitação é reprovada antes até de ser apresentada para o cliente. Existem inúmeros fatores que podem gerar a negativa e é praticamente impossível

cobrir todos eles, principalmente porque um fator que inviabiliza um projeto pode ser um pré-requisito de outro, porém existem alguns cenários importantes de serem tratados, o primeiro deles é a possibilidade da necessidade ser atendida por outro caminho do sistema, como é de costume tanto a equipe quanto o cliente podemos “Usar uma bazuca para matar um mosquito”, como é falado em um ditado popular, ou seja, corremos o risco de criar soluções mirabolantes e robustas que para atender uma necessidade do cliente, mas muitas vezes o simples passa despercebido, no caso de sistema que já existem ou de produtos a chance de alguma funcionalidade já atender a Solicitação é alta, e não subestimar o que já existe na aplicação, nesta situação a solução proposta é a apresentação do sistema, e caso ele seja adequado para o cliente não será necessária a realização da análise.

Considerando os comportamentos do sistema quando é solicitada uma mudança ou customização em um produto, ou seja, uma ferramenta gerada de forma padronizada para atender diferentes clientes, neste caso por ser a mesma ferramenta disponibilizada para diversos usuários, a chance de uma mudança afetar o restante é muito grande, e na maioria dos casos quando se trata de um produto as alterações são negadas, ou precisam de um esforço alto em relação a testes e análise para que a mudança gere impactos positivos em todos os clientes, e em casos muito extremos o custo para a realização desta mudança faz com que o cliente desista, esta é a razão pela qual as empresas optam pelo desenvolvimento de uma solução customizada que será construída para tratar especificamente dos problemas apresentados.



As razões mais comuns pela qual a equipe nega a Análise de uma Solicitação de Mudança são: Falta de informações; o Impacto na regra de negócio do cliente; e Impacto técnico no sistema. A primeira está relacionada diretamente com uma falha de análise, onde o responsável por realizar o detalhamento ou divisão das atividades deixa passar alguns pontos imprescindíveis para que seja avaliada algum impacto grave no sistema, ou inviabilize a execução por falta de informações, em alguns casos a visão tanto do cliente quanto a do

analista é muito viciada na necessidade em discussão e um dos objetivos desta aprovação é que todos estejam de acordo com o que será realizado, e obter outras visões sobre o problema para obter outras perspectivas de análise e com isso corrigir estas falhas, que se descobertas em tempo de desenvolvimento podem gerar um impacto tão alto que levam projetos ao fracasso. Na segunda razão é levantado um impacto que não foi visualizado por esta visão viciada dos envolvidos na análise, neste caso é apontada uma falha de continuidade lógica com uma outra funcionalidade ou processo do sistema, onde o impacto gerado pode afetar outro comportamento da aplicação e fazer com que um dado seja alterado em um momento errado, um registro necessário seja excluído ou até mesmo um campo que precise ser alterado seja bloqueado. Já no terceiro ponto é avaliado o impacto técnico da funcionalidade, muitas vezes na definição de comportamento do cliente, no caso da regra de negócio o funcionamento é simples, porém existem muitas variantes de software como a diferença entre um sistema desktop e web, dados que não podem ser utilizados em um aplicativo mobile, ou até mesmo uma funcionalidade que deveria ser criada em conjunto com o sistema para ser padronizada e sua mudança pode afetar o comportamento do sistema como um todo, e muitas vezes não é possível avaliar os impactos.

Os motivos mais comuns que levam um cliente a negar uma Análise são o Prazo de Entrega, o Custo, Detalhamento, e Impactos Gerados. A maior importância é dada para os dois primeiros citados, pois normalmente serão fatores determinantes para alguma estratégia comercial o timing de mercado, se não conseguirmos disponibilizar a alteração no prazo e com o valor desejado a Solicitação é rejeitada e corremos o risco do cliente consultar outro fornecedor. Porém nesta situação é necessária a análise dos impactos gerados pela priorização de um dos pontos citados, considerando que os dois últimos são referentes ao escopo do projeto, pois quanto menor o prazo, o escopo precisa ser reduzido ou o custo será maior, quanto menor o orçamento para a mudança maior será o tempo necessário para realizá-lo e a quantidade de atividades é reduzida, e quanto maior for a alteração realizada, o custo e o prazo serão aumentados. Nesta situação cada uma destas visões precisam ser ajustadas para que os dois lados saiam ganhando, e é neste momento que a Análise será reprovada e revisada diversas vezes até que chegue em um ponto comum sendo positiva para ambas as partes.

Referente aos pontos relacionados com o escopo do projeto, e como é normal a o time de desenvolvimento se especializar nos processos e regras de negócio do cliente, muitas vezes os impactos levantados por uma funcionalidade ou as mudanças que serão necessárias em outros processo podem afetar de forma negativa o dia a dia da empresa e com isto a mudança que geraria valor pode se tornar uma problema, neste caso é importante expor estes pontos que não foram visualizados pelo cliente quando a alteração foi solicitada, e com isto podemos

inviabilizar a Solicitação fazendo com que ela seja cancelada, também existe a possibilidade de novas funcionalidades serem geradas a partir da visualização da Análise, e com isto a rejeição é necessária para que aquela mudança seja realizada de forma correta considerando os demais pontos, inclusive funcionalidades que impediriam o sistema de funcionar corretamente.

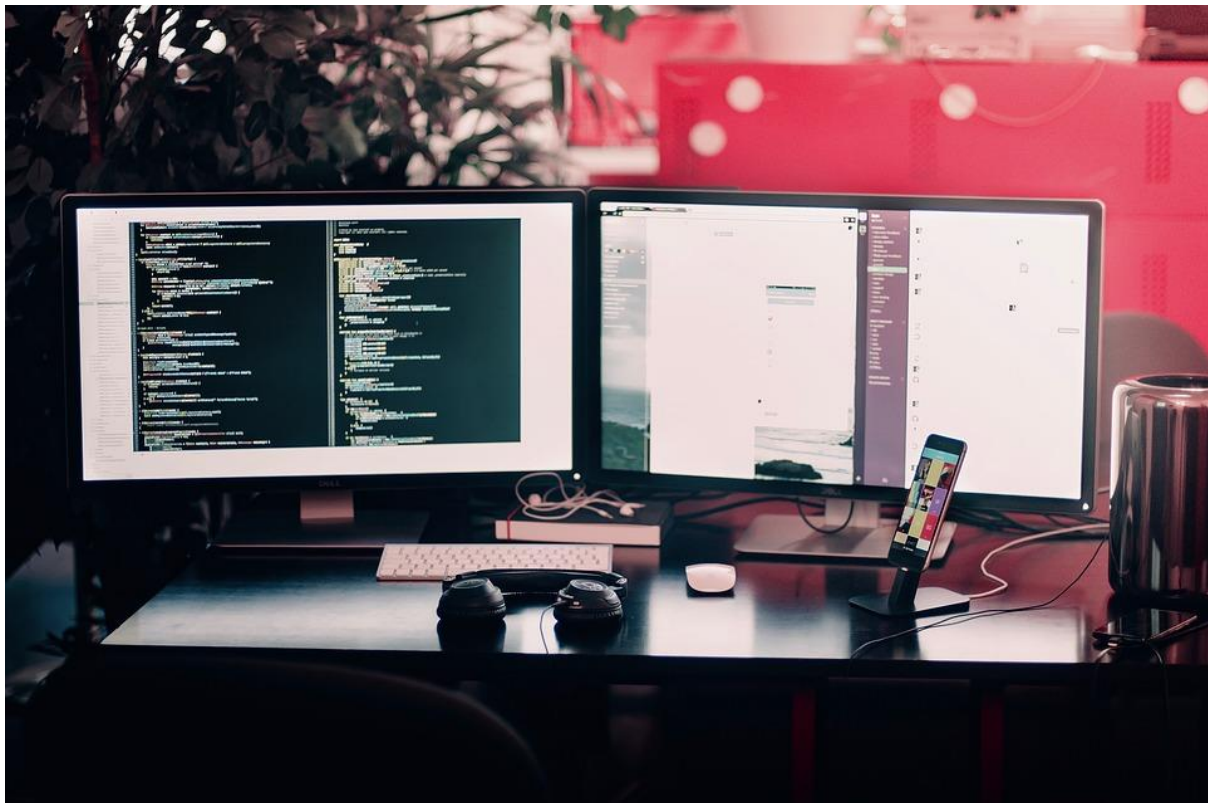


Com todos os pontos tratados, onde a equipe realizou a avaliação da Solicitação, e os motivos de rejeição foram tratados e passaram a atender a expectativa técnica, muitas vezes alinhados com o cliente, e após a aprovação do Escopo da mudança, o Custo que será necessário para executá-la e também o Prazo necessário para concluí-la esta Solicitação é direcionada para a área de desenvolvimento, onde será iniciada a execução das alterações solicitadas.

Desenvolvimento

Com a Análise aprovada pelo cliente a execução das mudanças apontadas é liberada, então é dado início a etapa do Desenvolvimento das Regras de Negócio e do Escopo Técnico, onde cada atividade detalhada será executada para que a necessidade do cliente seja atendida. Neste ponto o desenvolvedor inicia a implementação utilizando a descrição das funcionalidades, segundo o SWEBOK “Mudanças devem ser suportadas pelas ferramentas de controle de versão do código fonte.” ou seja, a ferramenta de Controle de Versão e os processos definidos para realizar alterações nos Itens de Configuração devem ser seguidos para que o desenvolvimento seja estruturado, além permitirem o controle do andamento das atividades relacionadas a alteração, e também o vínculo das mudanças realizadas nos Artefatos do sistema com estas tarefas. Durante a realização da implementação não é realizado somente a modificação dos objetos existentes no repositório do projeto, mas também a inclusão de novos Itens de Configuração que são gerado a partir das necessidades,

a cada nova funcionalidade o tamanho do projeto aumenta consideravelmente, e por isto é importante manter sempre o controle com base no relacionamento do Controle de Versão com o Controle de Mudanças para que o desenvolvimento seja realizado com qualidade e atenda a necessidade do cliente.



Um ponto importante referente a alteração e criação de Artefatos e a verificação de comportamentos e impactos não definidos na etapa de Análise, aqui vale lembrar a questão da assimilação das mudanças, é possível assimilar as alterações, porém é importante que isto esteja alinhado com as variantes do projeto pois aumento de escopo faz com que o custo e o prazo sejam afetados, nesta situação o ideal é que seja gerado em separado uma nova Solicitação de Mudança para que estes pontos sejam revisados e alinhados e aprovados com o cliente, para que seja possível a realização de forma que ambas as partes além de estarem cientes, estejam alinhadas quanto ao impacto gerado nos custos e prazos do projeto e ele seja realizado de forma que seja vantajoso para todos, e nenhuma parte seja lezada. Em alguns momentos estes pontos levantados não são tão críticos a ponto de afetar o prazo do projeto, normalmente colocamos um tempo adicional no cronograma para que seja possível assimilar estes imprevistos, porém é importante que o cliente esteja ciente para que isso seja utilizado como um fator adicional em uma negociação de outras Solicitações.

Verificação

Para garantir que as alterações solicitadas e analisadas sejam disponibilizadas para o cliente de acordo com o que foi definido e aprovado é necessário realizar a Verificação do que foi realizado, e neste caso não é necessária somente a validação funcional onde é avaliado o comportamento definido para o sistema, mas também a revisão técnica da estrutura de desenvolvimento e solução aplicada. Nesta etapa integrantes da equipe de projeto são selecionados para realizar verificações, segundo Figueiredo:

“Estes responsáveis devem verificar a descrição da alteração e o item de software alterado para serem capazes de fornecer um parecer aprovando a alteração, reprovando-a ou então informando ao responsável pela alteração a necessidade de rever algumas pendências na mesma para que esta seja aprovada.”

A primeira análise é realizada por um membro da equipe técnica, onde são avaliados cada um dos Artefatos que sofreram a alteração, e também os que foram gerados, para que seja evidenciada a aderência aos padrões de desenvolvimento definidos para o projeto, performance das estruturas de dados elaboradas e também a facilidade de leitura e entendimento deste código gerado, para que a manutenção destes itens seja facilitada em uma outra alteração. Caso sejam encontrados pontos de atenção nesta mudança aquela ela será reprovada, e será gerado um parecer referente aos Itens de Configuração com sugestões de melhorias, solicitações de revisão e adequações aos padrões definidos, com isto o responsável pelo desenvolvimento da atividade terá insumos para realizar os ajustes necessários e submeter novamente aquela mudança para validação, este processo é realizado até que a solução técnica seja aprovada para a integração, ou a tarefa de Enviar abordada no Controle de Versão, onde a mudança será disponibilizada para a segunda etapa, que é a verificação por parte do responsável pela Homologação e Regras de Negócio.



Com a alteração que foi implementada disponibilizada no repositório do sistema, o Analista de Qualidade irá realizar os testes daquela funcionalidade para validar se a mesma está aderente ao detalhamento da atividade, e também são realizados os testes com visão de usuário, onde normalmente são levantados pontos que não estavam detalhados na Análise da mudança porém são relevantes para que o sistema tenha a usabilidade ajustada para o seu perfil de utilizador. Esta validação é necessária para que a alteração seja disponibilizada para o cliente atendendo as expectativas de comportamento, com as funcionalidades realizando os processos de acordo com o esperado e resolvendo o problema que gerou a necessidade de mudança, e que a utilização esteja de acordo com o que já é realizado no dia a dia pelos usuários do sistema. Caso alguns destes pontos não esteja de acordo com o esperado, o parecer de homologação é gerado com o detalhamento dos cenários onde foram encontradas falhas, e também com os resultados esperados para aquelas situações, e com isto a mudança é retornada para o desenvolvedor realizar os tratamentos necessários, lembrando que as mudanças realizadas após a disponibilização dos Artefatos no repositório do sistema passarão pela revisão técnica antes de um novo envio para o repositório.



Após todas estas etapas serem cumpridas com a aprovação de todos os revisores, as funcionalidade entrarão no processo de inclusão na Baseline, já abordado no Controle de Versões, para que elas façam parte de uma versão estável do sistema a partir daí entrem no fluxo de envio de atualização do sistema para o cliente, na maioria dos casos existe uma etapa de validação por parte dos usuários do sistema, pois dependendo da alteração realizada existe risco de algum comportamento que não foi detalhado ou que está de acordo com a Análise porém gerou algum impacto não previsto afetarem o sistema em produção, para isto as alterações são disponibilizadas em um ambiente chamado de pré-produção onde o usuário realiza a validação, e caso sejam encontrados erros ou novas funcionalidades necessárias o processo de Solicitação é iniciado para que seja gerada uma nova versão para o cliente com os ajustes necessários.

Ferramentas

Como no Controle de Versão é possível realizar as etapas do Controle de Mudança de forma manual, e em alguns casos como é pregado na cultura do Scrum e Kanban é possível até realizar a utilização de post-its para controlar os estados de cada uma das atividades, porém nem sempre é a solução ideal para a realidade do seu projeto ou da sua empresa. Uma particularidade do Controle de Mudanças é que algumas empresas optam pela criação de um sistema próprio que será adaptado para sua necessidade, até mesmo pela particularidade de processo que pode ocorrer em cada etapa, porém ainda sim existe uma infinidade de ferramentas já criadas para atender a sua necessidade, com soluções gratuitas ou pagas. Pela criticidade desta Atividade dentro da Gerência de Configuração o interesse por resolver

ou facilitar a execução desta etapa com o objetivo de popularizar ou vender um sistema que atenda a estas necessidades é alto, e a cada momento novas metodologias de desenvolvimento surgem e novas ferramentas adequadas a elas para dar suporte a sua implementação no projeto, como responsável por cuidar das configurações do projeto você deve entender o processo que é realizado pelo time de desenvolvimento e até mesmo pela empresa como um todo e selecionar uma ferramenta ou sugerir a criação de uma que atenda a sua realidade, a seguir serão apresentados algumas versões gratuitas e pagas.



- Trello



O Trello é uma ferramenta de quadros, não é especificamente uma ferramenta de gerenciamento de projetos, ou de Controle de Mudanças, porém por ser construída para atender a diversas necessidades ela atende também ao controle das Solicitações de Mudança, segundo o Trello “Trello é a maneira gratuita, flexível e visual de organizar tudo, com quem quiser.”, é uma ferramenta que além de ser gratuita, permite que diversas pessoas interajam com os quadros de atividades, além de permitir a visualização rápida do status do projeto.

- Redmine



O Redmine é “uma aplicação web flexível para gerenciamento de projetos.” a partir desta informação é possível verificar que o objetivo dela é a gestão do projeto, porém as atividades estão inseridas neste contexto, e o controle delas é o objetivo final, o objetivo final do Redmine é ser prático, não possui uma interface visual muito elaborado, porém é uma das ferramentas de código aberto mais robustas e amplamente utilizada no mercado, por ser uma alternativa gratuita as ferramentas do mesmo porte.

- Github



“GitHub é como as pessoas constróem software” atualmente é a ferramenta mais utilizada pela comunidade de desenvolvimento para compartilhar seu código fonte, encontrar projetos e exemplos de soluções, e também criar suas próprias soluções privadas, é um repositório online que utiliza o Git com foco em compartilhamento, dentro da ferramenta além da solução para armazenar código, também é possível gerenciar atividades utilizando o benefício do compartilhamento e a colaboração para construir sistemas, atualmente conta com mais de 31 milhões de usuários.

- Azure DevOps (Visual Studio Team Server)



O Azure DevOps, é a ferramenta de gestão e controle de projetos online da Microsoft, segundo seu site “Planeje com mais inteligência, colabore melhor e envie mais rapidamente com um conjunto de modernos serviços de desenvolvimento” com foco direcionado na produtividade e agilidade na entrega do produto para o cliente, o antigo VSTS (Visual Studio Team Sever) é um serviço comercializado que disponibiliza um ambiente para a colaboração entre membros da equipe, onde as atividades são gerenciadas, e também estruturas para hospedagem do sistema em desenvolvimento, centralizando todas as necessidades de um projeto em um só lugar.

- Microsoft Project



O Microsoft Project é a principal ferramenta utilizada no planejamento e gerenciamento de projetos tradicional, e sua principal funcionalidade é a criação de cronogramas, possui o controle das atividades, porém é mais simplificado do que as demais com uma visão focada no nível gerencial para acompanhamento, por ser um das ferramentas mais robustas do mercado é paga e atualmente conta até com acesso em nuvem.

- Jira



Segundo seu site “A ferramenta número 1 de desenvolvimento de software usada por equipes ágeis” é uma ferramenta com foco no Desenvolvimento de Software Ágil, onde as atividades são controladas e acompanhadas por todos os envolvidos no projeto, conta com diversas funcionalidades como a criação do cronograma, e também na construção de quadros ágeis para a utilização da equipe. É uma ferramenta paga porém é uma das mais completas do mercado e também uma das mais utilizadas.

Considerações Finais

O desenvolvimento de um sistema é realizado para a solucionar algum problema, ou atender alguma necessidade de automatização, o projeto é iniciado com o escopo relacionado a estas demandas, porém com o desenrolar do projeto são encontrados outros pontos, como uma atividade que não foi sinalizada pelo usuário durante o levantamento de requisitos, uma mudança de lei que afete algum cálculo ou processo do sistema, ou a alteração do timing de mercado gerou uma nova funcionalidade, e os demais possíveis motivos para que seja necessária a mudança no que havia sido planejado. O maior problema destas alterações é que elas não estavam previstas no escopo inicial do projeto e geram diversos impactos no sistema como um todo, fazendo com que decisões tomadas em relação às necessidades levantadas inicialmente pelo cliente não sejam mais verdade. Caso estas alterações sejam realizadas sem que as definições já realizadas sejam revistas os seguintes impactos são possivelmente gerados:

1. Aumento de escopo funcional;
2. Ausência da análise de impacto técnico;
3. Ausência da análise do impacto de negócio;
4. Descumprimento de prazos;
5. Geração de erros;
6. Desenvolvimento de uma funcionalidade desnecessária;
7. Priorização de uma funcionalidade que não é de extrema necessidade;
8. Desenvolvimento de uma funcionalidade dependente de outra que não está funcionando.

Estes são alguns dos possíveis impactos gerados pela ausência de um processo que avalie a necessidade é um risco muito grande para o projeto, segundo Boller “Em projetos de grande porte, fazer modificações sem um esquema formal de controle pode levar ao caos.” que seria a concretização dos riscos apresentados acima no sistema com frequência, e para evitar que o caos seja instaurado em nosso projeto utilizamos a Gerência de Configuração para que tenhamos um controle sobre a evolução da nossa aplicação, e para lidar com o problema dos surgimento de necessidades executamos a Atividade de Controle de Mudança, que se resume a gerenciar o surgimento melhorias e realiza a análise de cada uma delas, para que o desenvolvimento seja realizado de forma que todas os envolvidos estejam cientes do comportamento da modificação implementada, e que ela seja validada quando concluída para que seja possível evidenciar que a solicitação apresentada foi executada e concluída realizar dentro do esperado.

Segundo Hass “O Controle de Mudanças é iniciado por um evento”, este momento é definido como Solicitação, ou Solicitação de Mudança, onde o requerente da alteração descreve a sua necessidade, qual a motivação da solicitação, quais são os comportamentos realizados pelo sistema e seus respectivos resultados esperados, onde cada uma das mudanças será realizada, informando os produtos ou ferramentas afetados, e apresentar a data de conclusão esperada para a mudança. Quanto maior for o volume de informações apresentado nesta etapa, maior será a qualidade demais tarefas realizadas, por ser o ponto de partida do processo de alteração ele é utilizado como insumo das demais tarefas e também como base de comparação para que seja possível avaliar se o que foi definido e executado está de acordo com o que foi Solicitado, nesta etapa não só a quantidade mas também a qualidade das informações disponibilizadas de extrema importância para que a mudança seja realizada com sucesso.

Com a Solicitação preenchida com todos os dados possíveis para descrever os comportamentos desejados pelo cliente, é iniciada a Análise e Priorização destas informações, nesta tarefa o integrante da equipe responsável pelas regras de negócio do sistema irá realizar a avaliação da Solicitação como um todo, e executa a tradução dos comportamentos em funcionalidades, que por sua vez são quebradas em atividades independentes utilizadas como guia para a execução do desenvolvimento, além de definir qual será o esforço necessário para a execução desta implementação, que é utilizado para a definição do custo para realizar a mudança e também o cronograma de desenvolvimento, onde será apresentado os prazos para conclusão da Solicitação. Durante a realização deste detalhamento podem ser identificados alguns pontos de atenção ou impactos técnicos ou de negócio que gerem a necessidade de sugerir adaptações na solicitação do cliente, ou até mesmo inviabilizam a funcionalidade, nesta situação é necessário que sejam realizados alinhamentos para que a Análise seja concluída com sucesso, onde todos os pontos de impacto sejam contemplados e possuam um tratamento fazendo com que a funcionalidade seja adequada a estrutura do sistema.

Quando a Análise é concluída ela é enviada para Aprovação, tanto da equipe técnica quanto para o cliente, a primeira etapa de verificação é a revisão pela equipe de desenvolvimento onde será realizada a avaliação das possíveis soluções e se o que foi definido é viável tecnicamente, os impactos em comportamentos do sistema também são revisados, pois a alteração de uma funcionalidade pode gerar um impacto técnico alto que não foi visualizado na etapa de negócios, e existe a possibilidade dele inviabilizar uma mudança por inviabilidade técnica ou de custo e prazo, caso sejam encontrados pontos de atenção nesta avaliação ela será retornada para Análise e revisada até que seja aceita pela área técnica. Com o aceite do time de desenvolvimento precisamos que o cliente realize a validação para que o escopo

da funcionalidade seja aceito, ou seja o que será realizado pelo desenvolvimento, além da validação do custo e do prazo para a entrega, e também para que algum impacto de negócio que não contemplado na etapa de Análise seja revisto e não corremos o risco executar uma implementação que possa causar algum impacto no sistema ou gerar resultados diferentes dos esperados, caso alguns destes pontos sejam identificados a definição será retornada para o tratamento dos mesmos e com todos estes itens tratados o cliente pode dar o aceite de negócio. Vale lembrar que nesta etapa além dos fatores de prazo e custo que podem inviabilizar uma mudança, podem ser encontrados pontos não previstos no momento da Solicitação e que gerem um impacto alto no funcionamento ou na regra de negócio que impeçam a execução daquela modificação, neste caso ela é cancelada.

O Desenvolvimento é iniciado somente após a Aprovação da Análise, ou seja, com todos os pontos e de acordo com as alterações que serão realizadas no sistema, e também seus impactos, a execução será iniciada onde os desenvolvedores colocam em execução o Controle de Versão para que conforme as atividades são concluídas elas são preparadas para a integração no Repositório do sistema. Existe a possibilidade de serem encontrados pontos não previstos ou que necessitem de ajustes, mesmo com a revisão técnica e de negócio alguns casos específicos que não foram previstos podem ser levantados pelo time de desenvolvimento, neste caso o processo de Solicitação é iniciado para que todos os pontos sejam levantados, e dependendo do impacto gerado existe a possibilidade da execução ser realizada em um segundo momento ou quando for mais apropriado.

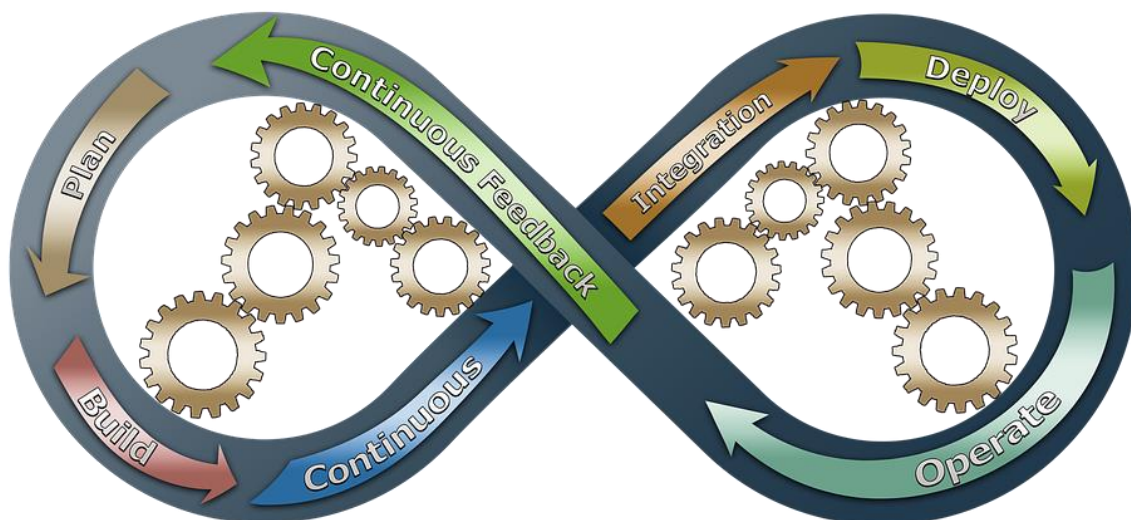
Com o desenvolvimento finalizado precisamos que as atividades que foram concluídas sejam revisadas, e com isto seja atestado que a implementação da mudança atende a necessidade apresentada pelo cliente, esta avaliação é realizada pela tarefa de Verificação, onde como na Aprovação, a equipe técnica e de negócio atuam para realizar suas respectivas avaliações, a primeira delas é a análise do código desenvolvido, onde um responsável irá comparar a evolução dos arquivos com base em suas versões anteriores, e avaliar se as alterações realizadas nele estão de acordo com os padrões de técnicos definidos para o projeto, caso não estejam a atividade retorna para Desenvolvimento com o parecer do responsável apresentando os motivos ou sugerindo alterações para melhorias de performance e facilitação de leitura do código, caso não existam pendências referente a esta verificação ela é direcionada ao Analista de Qualidade, que por sua vez irá realizar a avaliação funcional, onde as regras de negócio, usabilidade, estresse e testes com foco no usuário são realizados, caso sejam encontradas inconsistências o desenvolvimento será refeito, porém neste caso ao após concluir é realizada uma nova análise técnica antes de ser revalidada, quando todos os pontos foram ajustados a alteração estará validada para ser disponibilizada ao cliente.

As Ferramentas utilizadas para o Controle de Mudança são responsáveis por gerenciar cada uma das Solicitações e acompanhar seus status, além de permitir a visualização do estado de cada uma das suas atividades, possibilitando assim a validação do cronograma e gerando insumo para a tomada de decisão. Existem muitas ferramentas que servem para diferentes propósitos, em grande parte elas são direcionadas a metodologias e processos específicos, e cabe ao Gerente de Configuração entender quais as necessidades da empresa e qual a melhor ferramenta para ela, e caso ainda sim não seja possível encontrar uma adequada, é possível a construção de uma para que ela seja adaptada a sua realidade.

Integração Contínua

Introdução

Como citado por Hass em relação ao Controle de Mudanças, a o processo de alteração é um projeto em miniatura, de acordo com a maior referência em Gerenciamento de Projetos o PMBOK, que apresenta todas as etapas necessárias para que seja possível realizar um projeto adequado e que ele seja concluído com sucesso, os pontos que aparecem em todos os tópicos abordados, quando não em seus próprios, são Comunicação e Integração, que são fatores também considerados no Manifesto Ágil, que dentre seus tópicos apresenta características como entrega ao cliente, sistema funcionando como documentação, e aceitação das mudanças como parte do processo. Mesmo em realidades distintas, uma mais tradicional e outra inovadora, elas prezam pelo mesmo princípio que é a aproximação com o cliente, e na Gerência de Configuração mantemos este valor, onde valorizamos esta Integração e também a Comunicação, o sucesso de cada uma das Atividades apresentadas anteriormente está ligado diretamente ao contato com o cliente, inclusive temos etapas onde somente com o aval dele é possível prosseguir com o processo, e mesmo que a Atividade seja mais técnica é necessário que todo o time esteja ciente de todos as tarefas em andamento, além de ter acesso às informações das demais atividades para que seja possível avaliar qualquer impacto em sua tarefa atual.



A Comunicação é essencial em cada uma das etapas da Gerência de Configuração, pois ela começa na Solicitação do cliente, onde precisamos que ele apresente o qual a sua necessidade, e alinhar possíveis dúvidas para que tenhamos o entendimento completo da demanda, e a cada uma das demais etapas precisamos que ele esteja ciente e engajado com a evolução da mudança, e neste ponto precisamos da Integração dele com o time para que este contato seja facilitado, e com isto agilizamos o alinhamento e tratamento de pontos que ficaram pendentes. A Integração Contínua é a Atividade responsável por manter o cliente em constante contato com o projeto, e com livre acesso às informações necessárias para acompanhar o andamento da sua Solicitação, além de facilitar as entregas do sistema, fazendo com que ele se sinta parte do processo e garantindo seu engajamento, as Atividades de Auditoria de Configuração, Status Report, Construção de Versão Estável do Sistema e Homologação são realizadas com o objetivo de manter o cliente integrado e facilitar a comunicação com ele.



Na Auditoria de Configuração, realizamos uma avaliação de cada uma das Atividades e tarefas realizadas durante a execução do processo da Gerência de Configuração, onde é revisado o cumprimento dos padrões estabelecidos e também a conformidade com outros critérios que podem ser definidos por contrato ou situações do projeto, normalmente estão relacionadas a algum risco ou algum impacto que foi gerado e precisa ser resolvido, com a realização da Auditoria é possível avaliar a qualidade de execução das etapas e caso não

esteja de acordo com o desejado, realizar planos de ação para que elas sejam concluídas de acordo com o esperado. Com os resultados gerados pela conclusão da Auditoria de Configuração, precisamos que os demais integrantes do projeto e também as partes interessadas nas mudanças recebam estas informações para que seja gerada a visibilidade do andamento das etapas e também como elas estão sendo concluídas, o Status Report é a Atividade responsável pelo compartilhamento do estado de tarefa que está sendo realizada e também a geração de um relatório com as informações relacionadas a elas, com isto garantimos que todos os dados necessários estão sendo entregues ao integrantes do projeto, e com isto geramos insumos para as avaliações de cada um deles com impactos entre atividades ou ajustes no percurso do projeto.

Após concluir todas as etapas de acordo com o padrão de qualidade definido para o projeto atestado pelas Auditorias e compartilhado via Status Report, iniciamos a seleção das funcionalidades finalizadas para a disponibilização ao cliente, onde realizamos a Construção de Versão Estável do Sistema, esta tarefa é iniciada com a verificação de quais mudanças já foram concluídas e estão prontas para serem enviadas, após é realizado a junção dos Artefatos referentes as Baselines e com gerado um pacote com as últimas versões dos Itens e entregue ao cliente. Ao receber esta nova versão do sistema, antes que ela seja atualizada no ambiente real, ou de produção, é necessário que seja realizada uma avaliação da configuração disponibilizada, esta tarefa é chamada de Homologação, onde o pacote é liberado para que o usuário final atue na simulação de situações reais, e com isto sejam validadas todas as funcionalidades e alterações inseridas no pacote, caso sejam encontrados erros ou novas mudanças será realizado o processo de Controle de Mudança novamente, caso contrário o sistema está pronto para entrar em produção.

Auditoria de Configuração

A definição de um processo de Gerência de Configuração, e a seleção das Atividades e tarefas necessárias para controlar as mudanças do projeto, não garante que elas estejam realmente sendo executadas, e também que as definições estão sendo seguidas e ocorrendo de acordo com o esperado. Para que seja possível manter a Gerência de Configuração em andamento é necessária a adoção de mecanismos para a verificação das Atividades, a Auditoria de Configuração é responsável por realizar este acompanhamento e revisar cada etapa, para que seja possível validar que estão ocorrendo dentro do esperado e cumprindo com padronizações definidas, segundo o SWEBOK “A auditoria de software é um exame independente de um produto ou grupo de produtos gerados pelo desenvolvimento para avaliar a conformidade com a especificação, padrões, acordos contratuais e outros critérios.”

esta Auditoria visa garantir que todas etapas necessárias para a realização de uma mudança foram devidamente executadas, e que contaram com os pontos necessários para que a implementação seja entregue ao cliente de acordo com o esperado e com o máximo de qualidade possível. Uma das tarefas que estão relacionadas a Auditoria e tem como objetivo validar a implementação é a Revisão Técnica, abordada na tarefa de Verificação do Controle de Mudanças, onde o código desenvolvido é avaliado para que seja atestada a aderência aos padrões e também que a solução entregue atende às premissas de performance, legibilidade, reutilização e demais itens acordados com o cliente e time de desenvolvimento.



Outra atividade que visa esta garantia é a Auditoria de Configuração, ou Auditoria de Configuração de Software, que tem como objetivo complementar as validações realizadas pela Revisão Técnica tratando dos demais aspectos do desenvolvimento de uma mudança, onde além dos requisitos técnicos as validações de negócio e cumprimento dos pré-requisitos do sistema e processos são avaliados, segundo Pressman ela trata dos seguintes aspectos:

- Verifica se a modificação técnica especificada foi realizada e se foi incorporada alguma modificação adicional;
- Verifica se uma revisão técnica formal foi feita para avaliar a correção técnica;
- Verifica se o processo de software foi seguido;

- Verifica se constam anotados no Item de Configuração modificado, dados de rastreabilidade tais como data da modificação, autor, nº de requisição etc.;
- Verifica se os procedimentos de Gerência de Configuração para registrar e relatar a modificação, foram seguidos;
- Verifica se todos os Itens de Configuração relacionados foram atualizados de forma adequada.

Por ser uma Atividade que pode ocorrer em diferentes momentos do ciclo de vida Desenvolvimento de Software e da Gerência de Configuração, a necessidade desta atividade está vinculada a complexidade do projeto e também com seu nível de maturidade, pois caso ele possua um grau atestado pelo CMMI ou pelo MPSBR que exigem auditorias e o armazenamento destes relatórios, esta atividade é essencial para que a mudança seja finalizada com sucesso, bem como em um projeto grande com complexidade alta, caso as Atividades e Tarefas não sejam auditadas e certificado que elas atendem aos requisitos mínimos de cumprimento, a chance daquela alteração gerar um impacto que pode levar o projeto ao fracasso é alta. Porém nos casos onde as Auditorias são periódicas e são utilizadas somente como medição da qualidade, ou em situações onde não existe a necessidade pois é possível acompanhar o progresso das etapas sem muito esforço, a aplicação desta prática pode fazer com que os responsáveis pelas atividades o façam com mais atenção, com a possibilidade de gerar feedbacks ou lições aprendidas que contribuam com a evolução do time como um todo, além de contribuir para o resultado final.

Um ponto importante ao processo de Auditoria de Configuração é que pode gerar algum desconforto dos integrantes da equipe, principalmente quando é realizado por um time especializado nisto, por alguma diretriz da empresa ou para o cumprimento de algum processo relacionado a certificação ou modelo de maturidade, é sempre importante discutir e apresentar os benefícios desta avaliação, pois além da possibilidade de aprender mais sobre o que é avaliado e quais os fatores que levam uma etapa a obter o máximo de qualidade, aprimorando o resultado de todo o time, também existe a possibilidade de gerar uma visão geral sobre o entendimento da equipe em relação a demanda em questão e o engajamento com o resultado final. É importante que os feedbacks gerados sejam críticas construtivas embasadas em processos e definições que estão compartilhadas e de fácil acesso a todos da equipe, e com isso gerar outros fatores que aumente o engajamento com o mudança entregue ao cliente.

Status Report

Em cada uma das etapas da Gerência de Configuração são geradas diversas informações, e a grande maioria delas precisa ser compartilhada com as partes interessadas do projeto, em determinados momentos estes dados serão cruciais para que sejam tomadas de decisões, o Status Report é a forma como chamamos a centralização desta comunicação é também o processo de compartilhamento destes dados para que eles sejam revisados posteriormente a fim de validar as motivações de uma determinada decisão. Segunda Hass “O Status report deixa disponível, de forma útil e legível, informações necessárias para gerenciar com eficácia o desenvolvimento e a manutenção de um produto.” existe a necessidade de legibilidade da informação pois diferentes perfis de pessoas são envolvidas no projeto, sejam elas usuários básicos que entendem somente da operação do dia a dia, e até o desenvolvedor sênior especialista em diversas linguagens e padrões, e para que a mensagem seja transmitida a todos com eficiência é necessário que ela seja simples e objetiva utilizando o mínimo de termos técnicos possível.

A função do Status Report, ou relatório de estado, como seu nome já diz é reportar uma determinada visão de algum estado ou situação dentro do projeto de desenvolvimento de software, é utilizado para o acompanhamento do projeto onde são apresentadas diversas perspectivas e sumarização de informações, porém para que ele seja efetivo precisamos definir a mensagem ou os dados que precisam ser transmitidos e quais são os interlocutores, ou seja, os receptores desta informação. O compartilhamento de dados é necessário durante todo o ciclo de vida do sistema, e dentro da Gerência de Configuração é utilizado em cada uma das Atividades, onde as informações são repassadas para viabilizar o gerenciamento de cada uma delas, tendo em vista que não existe uma sequência cada uma ocorre independente da outra, porém a execução de uma delas pode exigir que outra também seja realizada, nesta situação ter visibilidade de quais dados serão contemplados nestas etapas faz com que o controle seja efetivo.



A seguir serão apresentados cenários onde o Status Report é normalmente utilizado:

- Listar quais são os artefatos do sistema e seus respectivos agrupamentos;
- Acompanhamento do crescimento do projeto pela comparação da quantidade de artefatos;
- Listar as mudanças realizadas em um artefato para a revisão técnica;
- Apresentar o volume de envio de alterações para o repositório do sistema;
- Listar as Solicitações realizadas;
- Apresentar o status de cada uma das Solicitações;
- Apresentar os Artefatos alterados em uma Atividade da Solicitação;
- Listar os bugs encontrados no desenvolvimento das mudanças;
- Listar Análises aprovadas e pendentes de desenvolvimento;
- Apresentar o status de cada atividade da Análise em desenvolvimento.

De acordo com Hass, cada empresa precisa definir quais são os relatórios devem ser possíveis de gerar, o que irá variar de acordo com o projeto e suas particularidades, podem ser utilizados dados, gráficos, tabelas, imagens, fluxogramas, entre outras possibilidades, é importante focar em formas que facilitam o entendimento dos envolvidos, fazendo com que as informações sejam repassadas e todos estejam com o mesmo nível de compreensão em relação aos aspectos do projeto e das Solicitações de Mudança.

Construção de Versão Estável do Sistema

O ciclo de vida do sistema é marcado por mudanças, uma das etapas de finalização final deste processo é quando uma nova versão, ou release, é disponibilizada para o cliente, este é o momento pelo qual todo o time trabalha, e é neste ponto que o nosso projeto pode ser um sucesso ou um fracasso, pois como vimos no CHAOS Report, caso não seja entregue dentro do prazo, custo ou com as funcionalidades definidas, na grande maioria dos casos o cliente não estará satisfeito, por este motivo temos que sempre manter nosso cliente informado sobre o andamento e caso tenhamos divergências entre as expectativas elas sejam alinhadas e caso necessário geradas outras Solicitações de Mudança para que as novas alterações sejam consideradas nos três pontos citados acima. Com a aplicação dos conceitos apresentados na Gerência de Configuração, todos os riscos e impactos foram controlados e tratados e o cliente estará de acordo e com a expectativa alinhada com o resultado, precisamos garantir que no momento em que a aplicação seja entregue ela contenha os Artefatos necessários para que as necessidades sejam atendidas, e é neste momento que realizamos a Construção da Versão Estável do Sistema, de acordo com o SWEBOK:

“Quando versões diferentes de um item de configuração estão disponíveis para entrega (como versões para plataformas diferentes ou versões com capacidades variadas), frequentemente é necessário recriar versões específicas e empacotar os materiais corretos para a entrega da versão.”

A tarefa responsável por realizar o agrupamento de todos os Artefatos do projeto em um único pacote, além de inserir as configurações e scripts necessários para que o sistema funcione corretamente é chamada de Build, ou construção, que é literalmente responsável por utilizar todos os Itens de Configuração selecionados para compor uma determinada configuração do sistema e gerar uma versão da aplicação, além de incluir os scripts e configurações necessárias, este processo pode consistir na atividades necessárias para que seja gerado um artefato único que deve ser entregue ao cliente, ou na execução de uma automatização deste processo que evita a sua realização manual. A seleção dos Artefatos e suas respectivas versões, e até mesmo os demais itens que se relacionam com eles é realizada via Baseline, que foi construída na etapa do Controle de Versão, aliado a seleção das Atividades do Controle de Mudança que foram concluídas, por este motivo uma versão do sistema conta com diferentes versões dos artefatos e também com diferentes baselines, esta análise é importante pois caso existam os mesmos artefatos é necessária a seleção da versão que contenha todas as funcionalidades desejadas, e não possua dependência de um outro item ou contenha alguma funcionalidade que ainda não foi validada.



Com o resultado do Build em mãos é necessária a disponibilização desta configuração do sistema para o cliente, o processo de entrega ao cliente é chamado de Release, em sua tradução literal é o lançamento de uma nova versão do software, normalmente quando atualizamos algum sistema é possível verificar as Releases Notes, ou notas de publicação, onde são apresentadas quais foram as novas funcionalidades disponibilizadas e também quais são as correções concluídas. O Release é diferente da versão do sistema, pois nem sempre o que o cliente recebe é a última configuração gerada, e são disponibilizados somente os Builds que estão estáveis, ou seja, que não apresentam erros ou alguma falha de incompatibilidade entre Artefatos, pois o impacto de gerado por um release onde o cliente encontre algum problema é muito alto, afetando o relacionamento, gerando desconfiança no sistema, e demais consequências, por este motivo é necessário muita atenção nesta etapa.



O Build é a Construção de uma Versão do sistema e o Release é a disponibilização para o cliente da Versão Estável do Sistema, a realização em conjunto destas atividades geram como resultado a Construção de Versão Estável do Sistema, ao finalizar cada uma das mudanças ou com a necessidade da disponibilização de alguma correção ou ajuste emergencial para o cliente este processo será repetido, para que sempre sejam entregues as configurações estáveis do sistema construídas de acordo com a funcionalidade concluída. O processo de Release também pode contar com a publicação do sistema no ambiente de homologação ou produção do cliente, esta atividade está relacionada a uma tendência de mercado que é o DevOps, que é a automatização de processos manuais como o de entrega, e também a cultura de disponibilização das aplicações com a maior agilidade possível, esta integração permite que o cliente receba e já inicie a validação do que foi Solicitado para atestar que está dentro do que foi definido e aprovado.

Homologação

A Atividade de Controle de Versão é responsável pela geração de novas funcionalidades ou alteração dos comportamentos do sistema, ao realizar a Solicitação e aprovar a Análise realizada, com as demais informações de impactos e mudanças que serão realizadas pelo time de desenvolvimento, mesmo que sejam realizadas as validações por testes cruzados entre os desenvolvedores ou o teste pela equipe dos analistas de qualidade e analistas de testes, é necessário que o cliente realize a sua própria comprovação de que as modificações foram realizadas de acordo com o esperado e atendem as expectativas, a execução desta comprovação por parte do cliente é chamada de Homologação, normalmente é realizada a disponibilização da Versão Estável do Sistema em uma ambiente criado com este propósito

que simula o sistema em produção e contém todos os dados necessários para que as validações sejam realizadas de forma a garantir que quando a mudança entrar no ambiente real ela não gerará nenhum impacto ou irá apresentar alguma falha que não foi identificada durante todas as etapas anteriores.



Por ser uma simulação bem próxima a utilização real do sistema, ela contará com algumas situações que não foram previstas nas outras etapas, principalmente com a interação do usuário final com o sistema, onde ele irá realizar o manuseio de acordo com seu dia a dia, e com isto alguns pontos são encontrados, que não foram identificados realizando a Análise das Solicitações, ou o acompanhamento do usuário por um Analista. Segundo Pressman a Gerência de Configuração é um processo que faz parte da Gerência de Qualidade, e normalmente a Homologação está ligada diretamente a Qualidade de Software, porém é importante ressaltar que caso nesta etapa sejam encontradas falhas ou divergências, as tarefas de Solicitação ou Desenvolvimento do Controle de Mudança são realizadas com o intuito de tratamento dos pontos apresentados, e com isto conseguimos garantir que as demais atividades de Análise, Aprovação e Verificação serão realizadas com o objetivo de realizar a entrega de uma nova configuração do sistema que contemple o tratamento das mudanças solicitadas.

Considerações Finais

Como já abordamos nos capítulos anteriores o objetivo final de todos os sistemas, produtos e serviços é atender aos clientes, manter sua satisfação e fazer com que esteja engajado e contribua com a evolução do projeto é essencial para que ele ocorra com sucesso. A Atividade de Integração Contínua é responsável por manter o cliente e contato com o as etapas da Gerência de Configuração e também facilitar a Comunicação entre ele e o time de desenvolvimento, além de agilizar o acesso ao sistema em funcionamento, fazendo com que ele consiga assegurar que todas as Solicitações foram disponibilizadas de acordo com as necessidades descritas, e também todos os impactos e efeitos colaterais da mudança foram tratados e não irão afetar a utilização do sistema. Para que seja possível realizar esta aproximação com o cliente, a Integração Contínua apresenta algumas atividades que permitem a visualização do andamento do projeto e também validam se alguns padrões e critérios estabelecidos estão sendo cumpridos, são elas: Auditoria de Configuração, Status Report, Construção de Versão Estável do Sistema e Homologação.

Definir quais Atividades da Gerência de Configuração serão utilizadas durante o desenvolvimento das alterações não garante que elas serão executadas, e também não comprova que os critérios mínimos de qualidade estão sendo atendidos, a Auditoria de Configuração realiza esta validação, tanto de inspecionar quais etapas estão sendo realizadas e quando, além de verificar se todos os pontos necessários para que elas sejam concluídas completamente e dentro dos parâmetros definidos. A execução desta Atividade é baseada em alguns questionamentos, que fazem relação com as características de suas respectivas etapas do processo, além dos critérios de cumprimento das mesmas, de acordo com as respostas encontradas é verificado se a tarefa foi realizada e se atende aos requisitos esperados para ela, vale lembrar que em processos formais, tanto as perguntas quanto as respostas são transcritas em um repositório para que seja possível recuperar estas informações caso alguma dúvida ou tomada de decisão seja necessária.

Com todos os pontos referentes às Atividades realizadas e a verificação do cumprimento dos critérios de qualidade, mesmo em um projeto que não exige a formalização dos resultados das Auditorias, é necessário que os envolvidos sejam informados dos resultados destas inspeções, e o Status Report é responsável por esta e as demais comunicações do projeto, onde além do compartilhamento dos resultados também é realizada a comunicação do andamento das Atividades, disponibilização de novas versões, e outras informações que forem necessárias. A comunicação é essencial para que as Solicitações de Mudança sejam concluídas com sucesso, a realização do Status Report faz com que o cliente se aproxime da

realidade do projeto, e mantêm ele e os demais envolvidos informados sobre os demais aspectos do sistema que podem afetar as demais tarefas em execução.

Conforme as Atividades relacionadas às Solicitações são concluídas e verificadas, é necessário que os Artefatos relacionados a elas sejam disponibilizados para o cliente, e com isso ele possa contar com as mudanças no sistema que está utilizando, porém são necessárias algumas considerações antes que um novo Item de Configuração seja inserido em uma versão do sistema, a principal delas é a validação da compatibilidade com os demais Artefatos. Para que a entrada de funcionalidades e correções seja realizada de forma natural e não gere impactos no comportamentos pré-existentes dos sistema é necessária a realização da Atividade de Construção de Versão Estável do Sistema, que realiza a seleção das Baselines que contam com as Solicitações finalizadas, verifica se existe impacto entre as funcionalidades e versões que precisam ser disponibilizadas, com estes pontos revisados e ajustados é realizado o Build da versão do sistema, onde todas as últimas versões dos Artefatos são empacotadas, gerando um artefato único para a entrega, a disponibilização deste item para o cliente é realizada pelo Release, que é o processo responsável por realizar as pré-configurações, como carga de dados, configuração do servidor, executar script de correção, e outros processos necessários para que o pacote com as últimas configurações do sistema sejam atualizadas.

Antes de realizar a inserção da última versão no ambiente de produção do cliente, mesmo que as etapas necessárias para validar se as mudanças foram desenvolvidas corretamente e atendem aos resultados esperados, é necessário que o cliente realize a sua validação e esta Atividade é a Homologação, onde a Construção da Versão Estável do Sistema é realizada na mesma estrutura para a produção, porém o Release é disponibilizado em um ambiente exclusivo de teses, que é chamado de pré-produção ou até mesmo homologação, e neste momento serão realizados os testes referentes às mudanças disponibilizadas, em funcionalidades relacionadas, e também caso possível um repasse geral em todos os pontos do sistema, neste momento é avaliado se a versão do sistema irá entrar em produção, com ou sem ajustes pendentes, porém se caso seja encontrada alguma falha ou ajuste que impeça a entrada será necessário este tratamento, caso seja um erro gerado pela implementação o time seguirá o processo de desenvolvimento, porém na situação de um ajuste ou mudança será necessária a iniciação de um novo processo de Solicitação.

Conclusão

As mudanças fazem parte do ciclo de vida de um sistema, independente da sua motivação temos a certeza de que elas irão ocorrer, mesmo em nosso dia a dia precisamos lidar com elas, seja quando o tempo está ensolarado e somos pegos de surpresa por uma chuva, ou acidente no caminho de um compromisso que gera um engarrafamento, ou seja tanto na nossa vida quanto nos projetos de desenvolvimento de software precisamos lidar com estas situações, para facilitar o controle destas situações a Engenharia de Software conta com a Gerência de Configuração que é responsável pelo tratamento das mudanças que ocorrem durante a evolução da nossa aplicação. Este processo que está inserido na Qualidade de Software define as Atividades necessárias para que os as alterações recebam o tratamento devido e sejam executadas de forma que no momento da disponibilização do sistema para o cliente ele recebe a solução dentro das expectativas e resolva os problemas do com a qualidade desejada, para isso precisamos realizar as Atividades que proporcionam a entrega atendendo a estas expectativas, são elas Identificação dos Artefatos, Controle de Versão, Controle de Mudanças e Integração Contínua. Segundo Pressman, “Identificação, controle de versão, e controle de mudanças te ajudam a manter a ordem onde o que em outra situação seria uma situação caótica e fluida.” a intenção da Gerência de Configuração é evitar que o caos se instaure no projeto e o levem ao fracasso, por este motivo alguns institutos padronizam e atestam a realização destas Atividades para que seja repassada a garantia ao cliente de que a empresa ou equipe que está realizando o desenvolvimento, o fará com qualidade e conseguirá reagir a estas situações adversas, concluindo e entregando com sucesso a solução desejada.

O primeiro passo para realizar este controle é definir onde ele será realizado, para isto precisamos selecionar dentre os objetos que compõem o nosso sistema quais serão gerenciados durante a execução da Gerência de Configuração, a Identificação dos Artefatos é a atividade responsável por esta distinção, para isto precisamos entender quais são os alvos desta seleção, dentro desta etapa eles são chamados de Artefatos ou Itens de Configuração, que são nada mais que os arquivos utilizados na execução do desenvolvimento, sejam eles de código fonte, requisitos, roteiros de teste, evidências de testes, configuração de servidores, solicitações do cliente, modelagem de dados, entre outros que são utilizados para este propósito. Estes objetos são divididos em duas categorias, Objetos Básicos e Agrupamento de Objetos, a primeira define a menor unidade existente dentro do repositório de dados do projeto, que varia de acordo com a granularidade e também pelo grau de controle desejado, ambos são afetados pela complexidade do projeto e também pela formalidade do projeto, o nível mais comum utilizado é o de arquivos individuais, porém

o código fonte como um todo, um conjunto de requisitos e modelagem de dados também são uma possibilidade, porém este segundo grupo também pode entrar na segunda classificação que a de Agrupamento de Objetos, onde um conjunto de Artefatos básicos que têm alguma relação ou objetivo em comum como o atender uma funcionalidade do sistema, ou representam um mesmo tipo de informação como todos os roteiros de testes, ou os scripts que criam a estrutura do banco de dados do sistema. Após categorizar os tipos de informações que serão trabalhadas na Gerência de Configuração precisamos das características de cada uma delas para que seja possível desempenhar as demais Atividades, os chamados metadados armazenam dados relacionados a estes itens, armazenam o seu nome, a data de criação, data da última alteração, os usuários relacionados a estas datas, descrição ou objetivo, e demais informações definidas de acordo com as necessidades do projeto, as outras etapas utilizam os metadados para identificar de forma única aquele objeto e também evidenciar a evolução daquele dado.

Com os Artefatos que serão gerenciados durante o ciclo de vida do sistema definidos, é necessária a definição de uma estrutura, seja um processo ou um sistema, que facilite o acompanhamento da evolução destes itens durante a realização das mudanças, o Controle de Versão é a Atividade da Gerência de Configuração responsável por monitorar as alterações realizadas nos Itens de Configuração e identificar cada uma delas a partir da definição de versões, que permitem que ao armazenar no histórico de mudanças cada uma seja identificada de forma distinta, com esta marcação é possível evidenciar quais foram as etapas realizadas durante a evolução daquele Artefato, com a possibilidade de identificar quais alterações foram realizadas em cada momento. O Controle de Versão tem como objetivo manter as configurações relevantes dos objetos em um banco de dados ou repositório do sistema, armazenar as versões para que seja possível reconstruir cada uma delas a partir das diferenças encontradas, e facilitar a seleção das configurações armazenadas para a construção de uma versão do software, para que seja possível a realização deste controle precisamos executar as tarefas responsáveis por este processo, onde cada uma das configurações ou versões destes Itens de Configuração são armazenados de forma única para que seja possível a identificação, são elas Armazenar, Extrair, Misturar, Guardar, e Enviar.

Iniciamos guardando cada um dos Artefatos em um repositório para que estes objetos sejam centralizado em um único local, e com isto seja possível realizar o gerenciamento das mudanças de forma única, sem precisar consultar diversos locais, além de manter somente os Itens estáveis armazenados, com isto o processo de desenvolvimento realiza um cópia local destes Itens para que seja possível executar as mudanças necessárias sem afetar que já foram enviados para o repositório, caso seja recuperado algum arquivo que contenha

alguma diferença entre a configuração local e a do repositório, será necessário que estes conflitos sejam misturados, para que as divergências co-existam ou que uma delas seja definida como a atual, este processo pode ser realizado em lote caso existam muitas ocorrências, nesta situação é possível assumir que a configurações recuperadas sejam válidas ou podemos manter as alterações locais para todos os itens, com isto precisamos guardar cada uma das mudanças realizadas antes que elas sejam enviadas para o repositório, para que caso ocorra algum imprevisto que faça as alterações serem perdidas é possível que elas sejam recuperadas a partir deste armazenamento local, além de centralizar todas as informações que serão misturadas com as versões estáveis quando forem finalizadas, isto ocorre quando enviamos as alterações realizadas para o repositório, este é o momento onde o ambiente local é sincronizado com as configurações armazenadas, e com isto são geradas novas versões para cada um dos Itens de Configuração.

Após realizar cada uma destas tarefas diversas vezes é necessária a geração de uma versão do sistema como um todo, e a seleção de cada uma das versões dos Artefatos para que façam parte desta configuração global, este agrupamento é chamado de Baseline, e é uma Atividade auxiliar a disponibilização da aplicação para o cliente, durante a evolução do software diversos Baselines são gerados, e também a cada disponibilização de funcionalidades para que seja possível agrupar os Itens de Configuração envolvidos nela, e existem ferramentas que tem o objetivo de facilitar e automatizar este controle, além de darem suporte na realização das tarefas citadas anteriormente. Ao estruturar todo o processo responsável por controlar cada uma das alteraçãoe realizadas e também garantir que elas sejam armazenadas, suas versões identificadas em seus respectivos Artefatos, além de agrupadas nas Baselines, precisamos controlar quais são as alterações a serem realizadas, quais os motivos que levaram sua execução, e quais as alterações necessárias para atender a necessidade do cliente, a Atividade responsável por realizar esta identificação é o Controle de Mudanças, que controla de maneira global as atividades relacionadas às alterações realizadas no sistema e também atesta que elas foram realizadas de acordo com a expectativa do cliente e com qualidade.

Este controle é composto por uma série de tarefas, que caso realizadas de acordo com o que é definido pela Gerência de Configuração garantem que as necessidades apresentadas sejam atendidas, ou que seja verificado que a mudança proposta não é viável para o sistema, o processo se inicia com a tarefa de Solicitação, onde o cliente envia um documento, ou conforme formato acordado, para apresentar e formalizar as alterações que são desejadas, com estes dados em mãos, é realizada a Análise e Priorização desta mudança, onde os comportamentos solicitados são transformados em requisitos e suas definições são detalhadas para que seja possível visualizar aquela necessidade como um todo, e também

quais as alterações que precisam ser realizadas no sistema, além de avaliar qual a sua prioridade sobre outras Solicitações em andamento ou enviadas, o tempo e o custo necessários para implementar os itens definidos, estas informações são repassadas para que a equipe de desenvolvimento validem se o que foi especificado é passível de desenvolvimento, se existe algum ponto que não foi considerado, e também revisar o esforço necessário contribuindo com o refinamento do custo e cronograma, além de elencar quais serão os responsáveis pela implementação, caso seja reprovado o analista responsável irá realizar a correção dos pontos apresentados até que o time técnico de o aval para prosseguir com a tarefa de Aprovação do cliente, onde ele irá avaliar os pontos e impactos apresentados, além de validar se o custo e o prazo definidos estão de acordo com a sua expectativa, neste momento esta alteração pode ser revisada para se aproximar das condições apresentadas, cancelada ou aprovada para o Desenvolvimento que é a próxima etapa, onde as funcionalidades, melhorias ou ajustes serão implementados no sistema com a utilização do Controle de Versão, e quando os mesmos forem disponibilizados no repositório, é realizada a Verificação do time de analistas de qualidade, onde será confirmado a conclusão das alterações atendendo aos padrões de qualidade do projeto e também as necessidades do cliente, caso sejam identificados pontos que precisam de revisão, estes serão direcionados para a etapa que atenda, caso um novo comportamento seja identificado para tratamento será encaminhado para análise, caso sejam encontrados erros, estes serão devolvidos para o time de desenvolvimento até que as mudanças esteja dentro dos padrões estabelecidos e atendam as expectativas do cliente.

A execução do conjunto de Atividades da Gerência de Configuração, e suas respectivas tarefas, pode ser demorado, massivo e estressante para o cliente, pois ele sempre quer receber todas as alterações solicitadas no menor tempo possível, e o maior desafio neste momento é manter ele engajado durante a execução do projeto, pois quanto maior for a importância dada por ele maior também será a chance do projeto ser concluído com sucesso, o objetivo como um todo é atender às expectativas do cliente e fazer com que ele continue usando o sistema e com isto nossa atuação continue sendo necessária, a Integração Contínua é a Atividade responsável por realizar o compartilhamento de informações relevantes e entregar o sistema para todos os interessados, além de realizar a inspeção de cada uma das demais etapas da Gerência de Configuração avaliando se cada uma foi cumprida de acordo com os critérios definidos, a Auditoria de Configuração é realizada em todos os momentos da mudança, e gera o parecer relacionado a cada ação que ocorre no projeto, o compartilhamento destas informações fica a cargo do Status Report que é a Atividade responsável por manter a comunicação e fazer com que estes dados cheguem aos interessados ou principais envolvidos naquela ação, fazendo com que todos sempre estejam

cientes do andamento das alterações. Com todos de acordo com as demais etapas e também com o resultados das mesmas, quando as mudanças chegam ao fim precisamos entregar o sistema ao cliente, e é neste momento que a Construção da Versão Estável do Sistema ocorre para realizar o agrupamento de todos os Baselines que foram gerados para entregar as alterações como um todo, ou como funcionalidades, e com estas Configurações do Sistema construídas é realizada a disponibilização de uma nova versão para que o cliente possa gerar seu parecer em relação ou que foi desenvolvido, neste momento é possível que diversas Atividades da Gerência de Configuração sejam refeitas, como Solicitações de novas mudanças, a sinalização de erros que precisam ser corrigidos e avaliados pelo time técnico e analistas de qualidade, e caso seja aprovada pelo cliente, então o processo é finalizado com sucesso e nosso papel como Gerentes de Configuração foi realizado com perfeição.

Referências

PRESSMAN, Roger S. **Software Engineering: A Practitioner's Approach**. 7.ed. New York: McGraw-Hill, 2010. 930p.

HASS, Anne M. J. **Configuration Management Principles and Practice**. 1.ed. Boston: Addison Wesley, 2003. 370p. (The Agile Software Development Series).

PROJECT MANAGEMENT INSTITUTE. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)**. Sexta edição. Newtown Square, PA: Project Management Institute, 2017.

THE STANDISH GROUP INTERNATIONAL. **CHAOS Report 2015**. {online}. Disponível via https://standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf. Arquivo capturado em 03/12/2018.

BOURQUE, Pierre; FAIRLEY, Richard E.; IEEE Computer Society. **SWEBOK. Guide to the Software Engineering Body of Knowledge, 2014**. {online}. Disponível via <https://www.computer.org/web/swebok/v3>. Arquivo capturado em 07/12/2018.

Oxford Dictionarie, <https://en.oxforddictionaries.com/>.

Subversion, <https://subversion.apache.org/>.

Git, <https://git-scm.com/>.

Mercurial, <https://www.mercurial-scm.org/>.

O'SULLIVAN, Brian. **Mercurial: The Definitive Guide** 2009. {online} Disponível via <http://hgbook.red-bean.com/>. Arquivo capturado em 17/12/2018.

COLLINS-SUSSMAN, Ben; FITZPATRICK, Brian W.; PILATO, Michael C. **Version Control with Subversion: For Subversion 1.7**. 11.ed. Stanford: O'Reilly Media, 2011.

CHACON, Scott; STRAUB Ben, **Pro Git**. Version 2.1.105. Mountain View: Apress, 2019.

Helix, <https://www.perforce.com/products/helix-core>.

What's New in Helix Core, 2018. {online} Disponível via <https://www.perforce.com/products/helix-core/whats-new-helix-core#tab-panel-2805>. Arquivo capturado dia 18/12/2018.

Helix Core Product Brief. {online} Disponível via <https://www.perforce.com/sites/default/files/pdfs/pf-core-product-brief-web.pdf> Arquivo capturado dia 18/12/2018.

AWS CodeCommit, <https://aws.amazon.com/pt/codecommit/>.

BARR, Jeff. **Now Available – AWS CodeCommit,** 2015. {online} Disponível via <https://aws.amazon.com/pt/blogs/aws/now-available-aws-codecommit/>. Arquivo capturado em 18/12/2018.

TFS. <https://visualstudio.microsoft.com/pt-br/tfs/>.

TFS Release Notes. {online} Disponível via <https://docs.microsoft.com/en-us/visualstudio/releases/tfs2015-rtm-vs>. Arquivo capturado em 18/12/2018.

What is Team Foundation Server? {online} Disponível via <https://docs.microsoft.com/en-us/azure/devops/user-guide/what-is-tfs?view=tfs-2018>

FIGUEIREDO, Sávio; SANTOS Gleison; ROCHA, Ana Regina, **Gerência de Configuração em Ambientes de Desenvolvimento de Software Orientados a Organização.** COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, RJ, 2004.

BOLLER, Alberto Filho; COSTA, Marcelo Nascimento; SPÍNOLA, Rodrigo Oliveira; KALINOWSKI, Marcos. **Gerência de Configuração: Definições Iniciais, Ferramentas e Processo.** Engenharia de Software Magazine, São Paulo. Edição 24.p14-30.

Trello. <https://trello.com/>.

Redmine. <https://www.redmine.org/>.

GitHub. <https://github.com/>.

Azure DevOps. <https://azure.microsoft.com/pt-br/services/devops/>.

Microsoft Project. <https://products.office.com/pt-br/project/project-and-portfolio-management-software>.

Jira. <https://br.atlassian.com/software/jira>.