



CENTRO UNIVERSITÁRIO

INSTITUTO DE EDUCAÇÃO SUPERIOR DE BRASÍLIA - IESB

Gerência de Configuração

Engenharia de Software e Métodos Ágeis

**Aluno(s): Brendo William de Souza Faria/Paulo Henrique Alves
Peixoto**

Matrícula(s): 20114290065/20113290051

Professor: Dr. Max E. Vizcarra Melgar

**Brasília – DF
2021**

I. Introdução

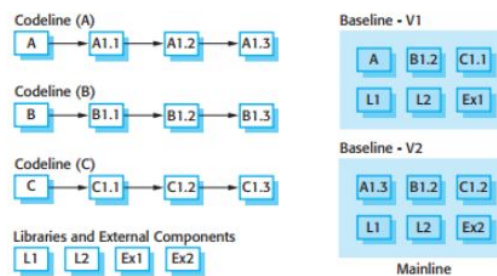
A Gerência é organização são sempre muito importantes , é o que fazem uma empresa desenvolver e seguir um caminho produtivo e rápido na construção de um sistema. E na **Gerência de Configuração de Software (GCS)** trabalha especificamente isso, pois os projetos de software estão em constante mudança alterando os requisitos em adaptação a o gosto do cliente, verificando se está conforme o combinado. Nesse trabalho vamos entender sobre como lidar com os artefatos no trabalho de criação de Software, e como eles são avaliados, separados e como eles mudam de acordo com o que é mostrado a o cliente.



(tirada do google imagens)

II. Desenvolvimento

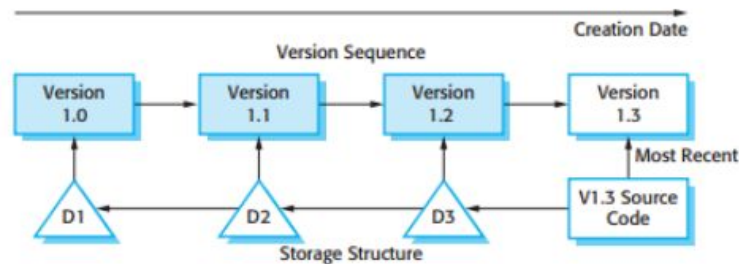
Durante o ciclo de vida do projeto, tem que ser levantado todos os pontos do requisitos, como será feito a arquitetura, backlog das interações, bibliotecas, códigos, ferramentas de teste, entre outras. Tendo como objetivo controlar a evolução e deixar sincronizado os Produtos de Trabalho. E parte disso é feito analisando os **gerenciamentos de versões** de componentes que trabalha com essas partes dos itens, chamados de **Artefatos**, podendo ser dentro do projeto alguns tipos de banco de dados, controle de APIs, linguagem de programação, tudo o que for relacionado com o software criado, variando muito com o que tem que ser feito, versões são combinações de atributos que a identifica. Essas partes são o corpo ainda não montado do projeto, peças não encaixadas que montaram o projeto, virando uma **release** para ser mostrada a o cliente. Como já diz o nome, são versões que sempre podem ser alteradas, montando **Codeline** que são linhas de versões que as configurações pode ter, alguns podem ver mais tipos de configurações do q outras mas os produtores sempre pegamos a versão mais estável. E com essas versões mais estáveis construímos as **Baseline**, que é a junção de codelines formando um grupo de atributos estáveis para montar a melhor versão do sistema, fazendo mais de uma delas entregando a mais atualizada para ser aprovado. Todas essas baseline formadas formaram outro grupo chamado de **Mainline** que é a junção de todas as baselines.



link:(

https://www.cin.ufpe.br/~kiev/IF682/09_Gerenciamento_de_configuracao_e_mudanca.pdf) pagina 33

Essas mudanças e alterações são arquivadas e documentadas para que ter uma visão mais ampla do projeto podendo voltar a versões anteriores, mas nem todas são salvas, normalmente escolhendo as versões mais otimizadas, chamadas de **Delta**, isso é feito para que não sobrecarregue o espaço de armazenamento.

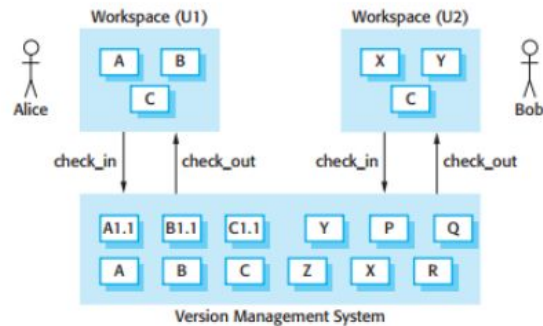


link:(

https://www.cin.ufpe.br/~kiev/IF682/09_Gerenciamento_de_configuracao_e_mudanca.pdf) pagina 52

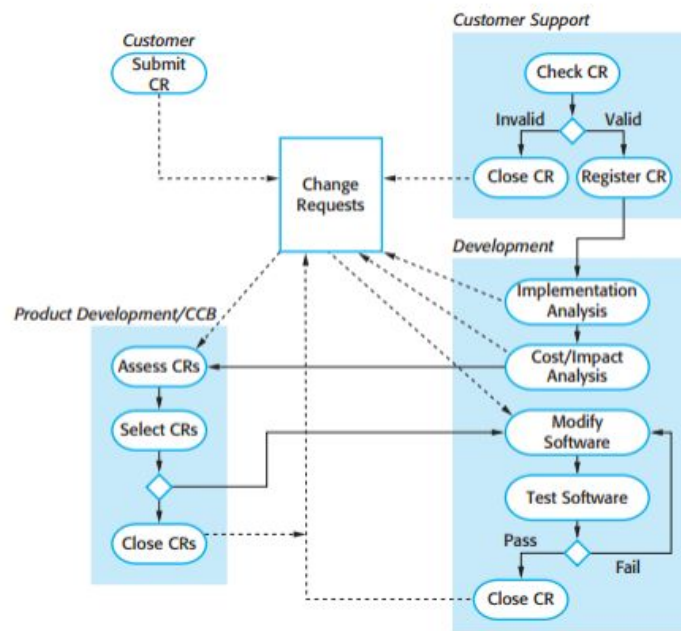
No espaço de trabalho temos a verificação do **check out** e do **check in**, onde tais acessam o sistema de gerenciamento de versões, com o check out fazendo uma cópia desses arquivos fazendo alterações, logo após a alteração eles fazem o check in, mandando os dados alterados para o gerenciamento de versões. Essa envio de versões podem vir a ser problemas, pois se duas pessoas alteram a mesma codeline, pode haver erros, a não ser que outros check out não possam ser feitos simultaneamente,

permitindo somente uma pessoa executá-lo.



link:(https://www.cin.ufpe.br/~kiev/IF682/09_Gerenciamento_de_configuracao_e_mudanca.pdf) página 51

Mas mesmo as vezes seguindo as instruções do requisito levantado algumas mudanças tem que ser feitas ,entrando no **gerenciamento de mudanças** que acontece se ocorrer algum impasse em alguma versão do projeto, acontecendo assim: um testador faz sua parte e identifica algum erro, uma falha no sistema, tendo que ser ajustado,outro tipo de mudança pode ocorrer se o cliente mudar de ideia, então alterações podem ser feitas para agradar na apresentação,então ele solicita uma **CR**. Essa solicitação vai para um banco de dados de solicitação de mudança sendo avaliada se ela é validada ou invalidada, indo para a equipe de desenvolvimento q faz uma análise, em seguida essas mudanças têm que serem avaliadas pelo **CCB(change control board)**,vendo o que tem que ser mudado,algumas vezes essas alterações não são feitas por questão de custo benefício, questões são levantadas para ver se vale mesmo a pena aquela alteração pois às vezes algumas mudanças alteram em outras partes podendo gerar mais erros, e isso pode custar tempo e não fazer tanta diferença, tendo pequeno impacto. As que passarem pelo comitê foram aprovadas, voltando para a mão dos desenvolvedores, ajustando e executando o software.



(link

:https://www.cin.ufpe.br/~kiev/IF682/09_Gerenciamento_de_configuracao_e_mudanca.pdf) pagina 15

Visto que, todo esse processo feito deu certo, e hora de reportar a o cliente fazendo uma **release**, será tomada a decisão de quando terão que ser mandadas aos clientes de acordo com o prazo ou por questão de marketing. Podendo gerar alterações como foi levantado anteriormente.

Tendo dentro de mercado de avaliação para empresas que querem ser certificadas para mostrar seu nível de maturação dentro do mercado de trabalho avaliando prazos e ferramentas, a CMMI e uma que já está no mercado a um tempo, ela é reconhecida mundialmente, fazendo a ter um valor um pouco auto, avaliando as empresas em 5 níveis de 1 a 5 tendo como 1 a empresa que não estão tão madura assim, e 5 a mais experiente, também temos a MPS.BR que possui o mesmo tipo de avaliação, sendo ela brasileira, com o preço mais baixo q o da CMMI, ela é avaliada de G a A sendo G o menos experiente é A o com mais experiência.

CMMI:



link:

(<https://upload.wikimedia.org/wikipedia/commons/thumb/4/47/CMMI.jpg/496px-CMMI.jpg>)

MPS.BR:



link: (<https://asrconsultoria.com.br/index.php/mps-br/>)

III. Exemplo de uso com sucesso

Vemos que existem vários exemplos de gestão no mercado, a ERP, que auxilia na empresa melhorando processos internos e integrados de muitos setores como os de venda, finanças e estoque, que já é muito utilizado no mercado de diversas áreas não só na criação de software. Alguns outros exemplos são os servidores de hospedagem também que ajudam em projetos BitBucket. O GitHub é um outro grande exemplo que ajuda na hospedagem de código-fonte e arquivo.

IV. Conclusão

Bom, vemos que a gerência de configurações é algo de extrema importância, e de amplo raciocínio, que se passa por mais de um desenvolvedor, e esse time com muito que constrói os requisitos, avalia tudo o que o cliente quer construindo com ele uma visão otimizada tanto de software quanto de verba investida, verificando ferramentas que serão usadas no projeto, avaliando isso tudo para aí então começar a desenvolver, produzir, as vezes reproduzir algo que já foi feito, sempre com o foco de otimizar esse projeto, para aí então construir as chamadas release e entregar partes do projeto para o cliente ver e aprovar. Pessoas que vêm de fora, acham que gerenciamento de software é algo que pode ocorrer depois que está tudo pronto, então você mexe nas configurações, etc, mas não, é algo bem mais complexo. Vemos que com funcionamento otimizado o Gerenciamento de configurações, tudo pode ser feito da melhor forma possível.

V. Referências

- https://www.cin.ufpe.br/~kiev/IF682/09_Gerenciamento_de_configuracao_e_mudanca.pdf
- <https://www.devmedia.com.br/gerencia-de-configuracao-de-software/9145>
- <https://www.youtube.com/watch?v=AZXFeJ92JvI&t=127s>
- <https://www.youtube.com/watch?v=eNYNv156YIY&t=973s>
- https://www.youtube.com/watch?v=Rflg8-dBd_c&t=1172s
- <https://asrconsultoria.com.br/index.php/mps-br/>
- algumas referências de imagens tiradas do google imagens