FATEC SÃO JOSÉ DOS CAMPOS JESSE VIDAL

TÓPICOS AVANÇADOS DE BANCOS DE DADOS

THAMYRIS DE CAMPOS

SETEMBRO/2013

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSÃO | AUTOR | DESCRIÇÃO |
| 19/09/2013 | 1.0 | THAMYRIS CAMPOS | Plano de gerenciamento de configuração |
|  |  |  |  |

# Histórico da revisão

# Descrição do Projeto

O objetivo é desenvolver um software para gerenciamento de uma vídeo locadora maior informações visualizar documento de visão.

|  |  |
| --- | --- |
| Lista de documentos do projeto | |
| Documento de visão 1.0 | <caminho do documento> |

# Processo

Após uma análise foi decidido que a equipe irá utilizar para o desenvolvimento desse projeto o processo de desenvolvimento cascata, devido ao baixo nível de complexidade do sistema a ser produzido, e por ele ter seus requisitos bem conhecidos.

# Equipe

|  |  |  |
| --- | --- | --- |
| Papéis e Pessoas | | |
| *Pessoa* | ***Ferramenta*** | ***Papel*** |
| Gerente de Projeto | MS-Project, Git, StarUML, Word, Excel | Realizar o primeiro contato com o cliente, levantando a necessidade do mesmo. |
| Analista de Sistema | MS-Project, Git, StarUml, Word, Excel | Desenvolver, documentar o sistema como um todo(Diagramas,Documentação,Requisito). |
| Desenvolvedor | Eclipse,Git, MS-Project | Desenvolver o a parte lógica do sistema |
| Design | MS-Project, Git, Dreamweaver, Photoshop | Desenvolver a interface do sistema |
| Equipe de Teste | MS-Project, Git, Eclipse | Realizar testes unitários no sistema |
| Equipe de Implantação | MS-Project, Git, Sistema Desenvolvido. | Implantar e treinar os usuários para utilizar o sistema desenvolvido. |
| Equipe de Treinadores | Manual do sistema | Treinar o usuário final para a correta utilização do sistema |

# Itens de Configuração

|  |  |
| --- | --- |
| Configuração de Hardware | |
| Adaptador de Rede | Intel® 82578DM Gigabit Network Connection |
| Adaptador de Vídeo | Intel® HD Graphics |
| Processador | Intel® Core™ I3 3.20 GHZ |
| Memória | DDR3 4 GB 1333 Mhz |
| Adaptador de Áudio | Realtek High Definition Áudio |
| Unidade de Vídeo | HD ST3500413AS ATA DEVICE 512 GB |

|  |  |
| --- | --- |
| Configuração de Software | |
| Sistema Operacional | Microsoft Windows 7 Professional 32 bits |
| IDE | Eclipse Juno Version Kepler Release Build id: 20130614-0229 |
|
| Linguagem | Java 7 |
| Gerenciamento de Processo | MS-Project |
| Servidor | Apache TomCat 8.0 |
| Ferramenta de Teste | Junit |
| Controle de Versão | Git - 1.8.4 - preview20130916 |
| Diagramas | StarUML |
| Banco de Dados | MongoDB 2.4.6 |

# Fluxo no Processo

## Acesso

Cada funcionário deverá se logar em sua máquina através do endereço da rede local http://seuservidor/nomedoservidor e acessar o MS-Project para que tenha acesso ao arquivo.

## Nova Tarefa

Funcionário quando for adquirir uma nova tarefa deverá atribuir seu nome a ela, para isso deverá digitar seu nome na coluna *NOMES DOS RECURSOS.*

É importante verificar durante a aquisição de novas tarefas as dependências das mesmas. Como seguimos o modelo cascata uma etapa depende da outra para que possa ser iniciada, por esse motivo deve-se tomar cuidado em pegar tarefas que estejam dentro da mesma etapa do processo.

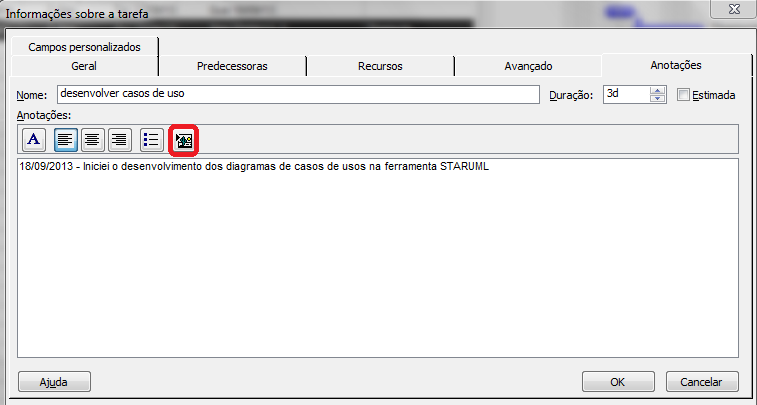
## Atividades

No decorrer do dia, as atividades que forem sendo desenvolvidas nas tarefas que foram pegas pelo funcionário devem ser formalizadas no MS- Project, dizendo:

* Como foi feito
* Com qual ferramenta
* Especificando a documentação que foi modificada

Essas informações devem estar contidas em *Anotações* do campo referente aquela tarefa.

Qualquer tipo de registro do tipo anexo deverá ser anexado através do bloco anotações, assim como mostra a imagem a seguir.



## Refatoração

Caso seja necessário realizar novamente o processo de desenvolvimento, será criada para cada tarefa uma nova sequência de tarefas e essas serão *vinculadas* as primeiras de forma a não perder o histórico contido nelas.

# Controle Versão

Utiliza-se para o controle de versão o Git, todas as máquinas deverão ter o software instalado e um repositório local com todos os arquivos.

## Estruturação de Pastas

O sistema estará estruturado da seguinte maneira

* Desenvolvimento
  + Interface – Dentro desta pasta terá arquivos relativos à interface do sistema. Os desenvolvedores das interfaces e do sistema como um todo deverão ter acesso à pasta.
  + Implementação – conterá arquivos relacionados ao desenvolvimento do sistema(códigos). Desenvolvedores terão acesso ao diretório.
  + Teste – arquivos de teste, esses arquivos estarão disponíveis aos testadores e aos desenvolvedores.
* Documentação – Toda a documentação do sistema contendo os casos de usos, diagramas de classes, sequencia. Requisitos do sistema, comunicação com o cliente, fechamento de contrato, os engenheiros de software terão acesso a essa pasta.

## Quando dar Commit ?

Quando se tratar de tarefas curtas (até 3 dias), estas deverão ser commitadas na sua conclusão, tarefas com mais de 7 dias de duração deverá ser commitada quando houver mudanças significativas, ficando a critério do responsável o momento exato de realizar o commit.

Ao realizar o commit no campo descrição deverá conter o nome do funcionário e sua identificação na empresa.

## Em caso de Conflitos (Merge)

No caso de existir este tipo de conflitos, os desenvolvedores, deverão resolver de forma manual, ou seja, conversar e ver qual parte será utilizada.

## Baseline

Toda etapa do projeto terá uma baseline associada a ela, a tabela a seguir apresenta as definições as siglas de cada uma.

|  |  |
| --- | --- |
| *Baseline* | |
| *Definição* | ***Sigla*** |
| Definição de Requisitos | BR1 |
| Projeto de Sistema e de Software | BSS1 |
| Implementação e Teste de Unidades | BTU1 |
| Integração e Teste de Sistemas | BTS1 |
| Operação e Manutenção | BOM1 |

## Comitê de mudança

|  |  |  |  |
| --- | --- | --- | --- |
| Comitê de Mudança | | | |
| *Responsável* |  | ***Areá na Empresa*** |  |
| Financeiro | Financeiro | | |
| Gerente de Projeto | Desenvolvimento de Projetos | | |
| Tester | Equipe de Teste | | |
| Responsável Implantação | Equipe Implantação | | |

## Fluxo de mudança

O comitê de mudança será solicitado para qualquer mudança que seja necessária ocorrer no projeto, tanto da parte do cliente quanto da parte da equipe.

O comitê será responsável por analisar a viabilidade da mudança, apresentando seu *Motivo* para a aceitação ou não da mudança.

Caso a aceitação seja aprovada, o comitê designará um responsável para atuar durante a mudança, este responsável estará diretamente relacionado com sua área de atuação, irão determinar também uma data para a mudança, tudo isso deverá ser documentado.

*Solicitante da Mudança: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Motivo da Mudança: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Descrição do que irá mudar:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Data Estipulada para a Mudança:\_\_/\_\_/\_\_*

*Responsável:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Área na Empresa:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

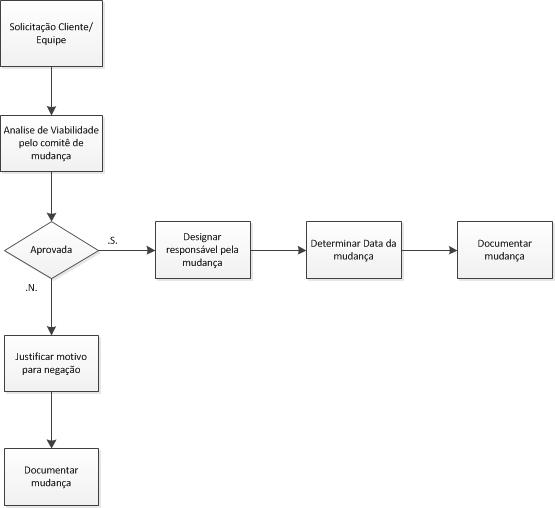
*Data da Reunião do Comitê: \_\_/\_\_/\_\_*

Caso a mudança não seja aceita também deverá existir um documento, contendo o motivo para a sua não aceitação:

*Solicitante da Mudança: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Motivo da Negação da Solicitação de Mudança: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Data da Reunião do Comitê: \_\_/\_\_/\_\_*



# Bibliografia

http://pt.wikipedia.org/wiki/Sistema\_de\_controle\_de\_vers%C3%A3o

http://pronus.eng.br/artigos\_tutoriais/gerencia\_configuracao/conceitos\_basicos\_controle\_versao\_centralizado\_e\_distribuido.php?pagNum=3

http://pt.wikipedia.org/wiki/Sistema\_de\_controle\_de\_vers%C3%A3o#Mesclagem

http://git-scm.com/book/pt-br/Ramifica%C3%A7%C3%A3o-%28Branching%29-no-Git-B%C3%A1sico-de-Branch-e-Merge

*Posso gerar uma versão do sistema no meio do fluxo do cascata ?*

- falar da estrutura de pastas

As TAGS UTILIZAS NO GIT SÃO UTILIZADAS PARA GERAR A VERSÃO !!

- dizer quando o programador vai poder dar commit na tarefa no git por exemplo no final do dia

- o mesmo a se dizer para os demais dentro da estrutura do cascata

- pensar na regra para o merger (pesquisar)

-regra de baselane (pesquisar)