



UNIVERSIDADE DO SUL DE SANTA CATARINA

THIAGO ALVES CAVALCANTE

**GESTÃO DE PROJETOS:
O GITLAB COMO FERRAMENTA DE GESTÃO ÁGIL EM UMA FÁBRICA DE
SOFTWARE**

Palhoça

2019

THIAGO ALVES CAVALCANTE

**GESTÃO DE PROJETOS:
O GITLAB COMO FERRAMENTA DE GESTÃO ÁGIL EM UMA FÁBRICA DE
SOFTWARE**

Relatório apresentado ao Curso **Tecnólogo em Gestão da Tecnologia da Informação**, da Universidade do Sul de Santa Catarina, como requisito parcial à aprovação na unidade de aprendizagem de Estudo de Caso.

Orientador: Prof. Roberto Fabiano Fernandes

Palhoça
2019

THIAGO ALVES CAVALCANTE

**GESTÃO DE PROJETOS:
O GITLAB COMO FERRAMENTA DE GESTÃO ÁGIL EM UMA FÁBRICA DE
SOFTWARE**

Este trabalho de pesquisa na modalidade de Estudo de Caso foi julgado adequado à obtenção do grau de Tecnólogo em Gestão da Tecnologia da Informação e aprovado, em sua forma final, pelo Curso Superior de Tecnologia em Gestão da Tecnologia da Informação, da Universidade do Sul de Santa Catarina.

Palhoça, 10 de Junho de 2019.

Prof. e orientador Roberto Fabiano Fernandes
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

A minha mãe, a quem devo todas as minhas conquistas nesta existência.

RESUMO

Este estudo de caso apresenta uma avaliação do software Gitlab como ferramenta de gestão de projetos e relaciona os recursos disponíveis no mesmo com a necessidade de usuários quanto as opções mais desejadas em um sistema de gestão de projetos.

A justificativa para o estudo se dá pela grande quantidade de opções de software disponíveis no mercado se propondo a apoiar as tarefas de gestão de projetos de forma que torna-se muito difícil decidir por qual ferramenta utilizar sem ter que recorrer a uma instalação para testes. Desta forma, este trabalho se apresenta como uma opção de apresentação específica de uma ferramenta visando enriquecer a literatura acadêmica com avaliações do gênero e se beneficiando da oportunidade de testar um sistema de gestão atrelado ao controle de versões mais utilizado no mundo atualmente.

A análise do sistema é baseada nas necessidades de uma fábrica de softwares atuante no setor público e privado do Brasil, utilizando a observação direta e entrevistas específicas com os envolvidos na instituição e se propõe a avaliar os recursos disponíveis no sistema chegando a uma conclusão sobre a viabilidade da utilização do Gitlab como ferramenta de gerenciamento de projetos e documentando uma sugestão de utilização do mesmo para este fim.

Palavras-chave: Tecnologia. Project Management. Gestão de Projetos. Scrum. Software.

SUMÁRIO

1 INTRODUÇÃO	6
2 TEMA	7
3 OBJETIVOS	8
3.1 OBJETIVO GERAL	8
1.1 OBJETIVOS ESPECÍFICOS	8
4 PROCEDIMENTOS METODOLÓGICOS	9
4.1 CAMPO DE ESTUDO	9
4.2 INSTRUMENTOS DE COLETA DE DADOS	9
5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA	11
6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA	15
6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA	15
6.2 RESULTADOS ESPERADOS	18
6.3 VIABILIDADE DA PROPOSTA	20
7 CONSIDERAÇÕES FINAIS	21
REFERÊNCIAS	22

1 INTRODUÇÃO

Realizando uma rápida pesquisa na internet é possível encontrar uma grande variedade de sistemas de gestão de projetos. Existem soluções em abundância, entretanto a pluralidade de opções, neste caso, acaba por ser um fator complicador em vez de uma solução, exigindo inúmeras horas de testes e validações de recursos até encontrar uma ferramenta que apresente aderência aos requisitos de sua escolha, sendo muitas vezes essa aderência somente representada por adaptações ou customizações nos sistemas em questão.

De acordo com (HELDMAN, 2005), ferramentas tendem a ser tão boas quanto o esforço aplicado em seu uso e foi esta a motivação para a realização desta pesquisa, que se propôs a realizar a instalação de uma versão específica do sistema Gitlab, utilizando-o de forma prática para realizar o levantamento dos recursos disponíveis e avaliar as possibilidades de utilização do mesmo com o foco em gerenciamento de projetos.

O Presente estudo não entra em questões específicas de instalação e nem apresenta modelos de documentos, focando apenas na utilização do sistema objeto deste trabalho. Nestes termos, espera-se contribuir com informações específicas extraídas dos testes em questão os quais podem facilitar a tomada de decisão na escolha de ferramentas do tipo.

2 TEMA

O área de gerenciamento de projetos tem enfrentado mudanças significativas nos últimos anos. Com a popularização das metodologias ágeis os modelos tradicionais de gerência, especificamente os que tratam do desenvolvimento de softwares, tem se tornado cada vez menos atrativos, por serem muitas vezes, compostos de milhares de páginas de documentos clichês que de forma geral nunca são lidos, forçando as partes que utilizam esses métodos a endossarem fantasias. (Sutherland, 2014).

Visões e métodos mais recentes, baseadas na identificação do valor para os clientes tem demonstrado ganhos significativos no tempo de entrega dos produtos sem comprometer a qualidade dos mesmos, além de nortear a adoção de ferramentas que tornem a sensação da tarefa de fazer software o mais natural possível. (Chacon, Straub 2014).

Com essas abordagens inovadoras, surgiram várias ferramentas se propondo a automatizar o processo de gestão de projetos, centralizando, agilizando as decisões e mantendo as informações estruturadas e disponibilizadas em interfaces com recursos que tornam prática a documentação, alocação de pessoas, materiais, facilitando revisões e aumentando o controle gerencial das fases implementadas, sendo estes itens, citados como uns dos principais benefícios do gerenciamento de projetos. (Vargas, 2009).

Conforme o apresentado, as seguintes perguntas são feitas: pode o Gitlab atender aos requisitos do gerenciamento de um projeto de software ágil? Quais seriam os recursos que poderiam ser utilizados e quais os benefícios de adotar a ferramenta?

Devido aos principais guias de gerenciamento de projetos não mencionarem ferramentas e/ou softwares específicos para controle e gerenciamento de projetos e devido ao crescente número de opções de sistema disponíveis para tal fim, foi escolhido este estudo de caso como forma de contribuir para a literatura do gênero em português.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Analisar os recursos do software Gitlab enquanto ferramenta de gestão de projetos.

1.1 OBJETIVOS ESPECÍFICOS

Avaliar os benefícios do uso do software em sua versão da comunidade e os ganhos relacionados a utilização da mesma;

Elencar os recursos da ferramenta, sua usabilidade e a aderência com práticas de gestão ágeis;

Documentar uma sugestão de utilização da ferramenta para gerenciar projetos de software alinhada com as ferramentas do framework Scrum.

4 PROCEDIMENTOS METODOLÓGICOS

4.1 CAMPO DE ESTUDO

O presente estudo de caso está fundamentado em um formato descritivo, sendo o universo da pesquisa, uma fábrica de software atuante no setor público do estado de Alagoas. A pesquisa apresentará visões dos gerentes de projetos sobre recursos desejáveis em uma ferramenta de gestão e analisará um software em específico quanto a sua afinidade e possibilidade de utilização no gerenciamento de projetos ágeis, tendo como referência o framework Scrum.

Por fim, o sistema será analisado enquanto ferramenta de gestão de projetos, destacando possibilidades, adequações e os ganhos que uma adoção possa vir a gerar.

4.2 INSTRUMENTOS DE COLETA DE DADOS

Os instrumentos de coleta de dados adotados neste trabalho são descritos no quadro a seguir.

Quadro 1 – Instrumento de coleta de dados

Instrumento de coleta de dados	Universo pesquisado	Finalidade do Instrumento
Entrevista	Gerentes e líderes de projetos, serão questionados quanto aos recursos desejados em um sistema de gerenciamento de projetos.	Identificar quais recursos são mais desejados em um sistema de gerenciamento de projetos e avaliar se o software objeto deste estudo atende a expectativa.
Observação direta	Análise dos recursos disponíveis no sistema Gitlab na versão da comunidade.	Elencar os recursos disponibilizados pela ferramenta e sua aderência com os desejos dos entrevistados.
Documentos	Livros e artigos relacionados ao tema.	Fundamentação teórica.
Dados arquivados	Não serão utilizados.	N/A.

Fonte: CAVALCANTI e MOREIRA (2008).

5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA

O presente estudo analisará o software Gitlab como ferramenta de gerenciamento de projetos em um ambiente de fábrica de software, sendo a empresa objeto do estudo uma prestadora de serviços de desenvolvimento de sistemas atuante no setor público e privado. A empresa em questão, utiliza, como balizador do fluxo de desenvolvimento de seus sistemas, o método RUP – Rational Unified Process, que segundo (Kroll e Kruchten 2003) é um processo de engenharia de software estruturado que define claramente quem, é responsável pelo que, como as coisas devem ser feitas e quando fazê-las.

Não existe em uso na instituição objeto deste estudo, um software específico de gerenciamento de projetos, sendo a documentação descritiva do processo de gerência, base, para todos os documentos e artefatos gerados durante o ciclo de vida do projeto. Estes documentos são mantidos em um servidor de repositórios de códigos, no caso em questão, o Visual SVN (<http://www.visualsvn.com>), o qual é o sistema central de controle das versões da documentação e códigos dos softwares da organização.

A instituição faz uso de práticas ágeis em busca de aumentar a qualidade e segurança dos sistemas desenvolvidos, sendo: uso de testes automatizados (TDD, BDD), técnicas de programação extrema (XP Programming). Na área de gerenciamento, práticas do Scrum vem sendo adotadas de forma não integrais e isoladas, contudo tem ajudado a fomentar os valores descritos no manifesto ágil, representados na tabela abaixo:

Quadro 2 – Valores do Manifesto Ágil

Indivíduos e interações	mais que	processos e ferramentas
Software funcionando		documentação abrangente
Colaboração com o cliente		negociação de contratos
Responder a mudanças		seguir um plano

Fonte: Site Manifesto Ágil (2019).

A partir destas informações, alguns gerentes de projetos e analistas de requisitos foram entrevistados com o propósito de conhecer seu entendimento atual de metodologias ágeis e os recursos que os mesmos consideram mais importantes em um sistema de gestão de projetos. Para elencar tal conhecimento, as perguntas abaixo foram realizadas em forma de questionário, solicitando responder sim ou não:

Quadro 3 – Questionário sobre metodologias ágeis.

Pergunta	Sim	Não
Você conhece metodologias ágeis?	88,89	11,11
Já trabalhou com metodologias ágeis em algum projeto?	66,67	33,33
Você conhece Scrum?	77,78	22,22
Conhece os valores do manifesto ágil?	44,44	55,56

Fonte: Elaboração do autor (2019).

O segundo questionário foi construído embasado na afirmação: sistemas de controles inadequados são uma das causas de fracasso em projetos (Vargas, 2009) e teve como propósito identificar os recursos desejados pela instituição, em um software de gestão de projetos, sendo estas informações usadas para verificar a presença de tais recursos no sistema objeto deste estudo.

Como método de classificação, foi utilizada a escala de Likert que sugere a seguinte pontuação: 1) Discordo totalmente; 2) Discordo; 3) Indiferente; 4) Concordo e 5) Concordo totalmente. (Frankenthal, 2017). Sendo computados a pontuação total e a média, permitindo a classificação dos recursos apresentados por grau de importância.

Quadro 4 – Questionário levantamento de recursos desejados.

Os itens abaixo correspondem aos recursos mais desejados em uma ferramenta de gestão de projetos?	Total	Média
Permitir uma parametrização simples para início de utilização.	29	3,22
Possuir uma interface amigável e moderna.	30	3,33
Permitir o gerenciamento das tarefas relacionadas ao projeto.	40	4,44
Possuir uma visão de Kanban integrada.	36	4,00
Disponibilizar opções de colaboração entre os integrantes de um projeto.	31	3,44
Permitir a visualização do andamento de múltiplos projetos em simultâneo.	35	3,89
Permitir trabalhar com modelos de documentos e projetos.	27	3,00
Permitir integração com sistemas de versionamento de códigos.	34	3,78
Permitir integrações com outros sistemas.	33	3,67
Possibilidade de criar e manter cronogramas de forma simples.	42	4,67
Permitir criar e manter a documentação dos projetos de forma simples.	37	4,11

Fonte: Elaboração do autor (2019).

Durante a entrevista, foi detectado que houveram tentativas de utilizar outros softwares de gerenciamento, entretanto, os experimentos resultaram na descontinuação das iniciativas pelos motivos abaixo relacionados, conforme as necessidades da instituição, ordenadas por número de citações:

- Impossibilidade de visualizar múltiplos projetos em boards diferentes ao mesmo tempo. (4 citações)
- Limitações quanto a geração e manutenção de cronogramas. (4 citações)
- Complexidade de instalação e manutenção. (3 citações)
- Complexidade de parametrização e utilização. (3 citações)
- Limitações relacionadas a documentação dos projetos (2 citações)
- Interface obsoleta.

Os itens acima relacionados, foram destacados como causa da não adesão entre os participantes dos projetos, evidenciando a ausência da realização de um estudo de viabilidade prévio que norteasse tais esforços. Contudo, tais experimentações contribuíram para consolidar a visão dos envolvidos sobre os recursos mais importantes em um software desta natureza.

Seguindo com a análise, de acordo com o questionário referente ao quadro 3, evidenciou-se que 88,89% dos entrevistados possuem conhecimentos em metodologias ágeis; 66,67% já trabalhou com metodologias ágeis em projetos na instituição e 77,78% dos gerentes e líderes entrevistados tem conhecimento do Scrum e suas regras. Quanto aos valores do manifesto ágil embora não fosse de conhecimento de todos, apenas 44,44% conheciam, evidenciou-se nos demais a inclinação a aderir aos mesmos, ação esta, que habilita uma revisão e redução de artefatos baseando-se na agregação de valor nos projetos de software.

Evidenciou-se ainda, que embora os analistas de requisitos possuíssem conhecimento no Scrum, desconheciam os valores do manifesto ágil e apresentaram certa resistência, mais especificamente quanto ao redesenho de artefatos de documentação. Tal resistência relaciona-se a impressão de que não haveria demanda suficiente para os mesmos caso uma redução fosse aplicada indiscriminadamente. Contudo, esta impressão se demonstra equivocada e inteiramente fundamentada no desconhecimento dos processos e eventos ágeis.

Tomando a metodologia RUP como exemplo, a fase de elaboração, onde se realiza a maior parte das tarefas de levantamento de requisitos, necessita que toda documentação que compõe as funcionalidades a serem construídas estejam completas para que em seguida a fase de construção seja iniciada, geralmente deixando desenvolvedores sem atividades.

Ferramentas como as do Scrum forçam a prática de preparar pequenos de incrementos a serem implementados e com uma documentação mínima focando apenas no que é

de valor para a construção. Esta reorganização do fluxo de trabalho permite aplicar o conceito de geração de demanda contínua, que ao liberar estes incrementos correspondentes a pequenas funcionalidades do projeto para construção, leva imediatamente os analistas a continuar o processo de levantamento de requisitos para as próximas funcionalidades até o término do projeto. Não havendo desta forma, uma redução de atividades.

As respostas aos itens apresentados no Quadro 4 – Questionário levantamento de recursos desejados, permitiram classificar por ordem de importância os recursos mais desejados, pela instituição, em um software de gestão de projetos, classificação e recursos estes, que servirão como base para a análise de aderência do sistema a ser realizada a seguir. São eles:

1. Possibilidade de criar e manter cronogramas de forma simples.
2. Permitir o gerenciamento das tarefas relacionadas ao projeto.
3. Permitir criar e manter a documentação dos projetos de forma simples.
4. Possuir uma visão de Kanban integrada.
5. Permitir a visualização do andamento de múltiplos projetos em simultâneo.
6. Permitir integração com sistemas de versionamento de códigos.
7. Permitir integrações com outros sistemas.
8. Disponibilizar opções de colaboração entre os integrantes de um projeto.
9. Possuir uma interface amigável e moderna.
10. Permitir uma parametrização simples para início de utilização.
11. Permitir trabalhar com modelos de documentos e projetos.

Dando seguimento aos levantamentos desta pesquisa, cabe descrever de forma sucinta as principais informações sobre o framework Scrum visando compor base contextual para a proposta de implementação a ser realizada na próxima fase. O Scrum é um conjunto de ferramentas desenvolvidas por Jeff Sutherland e Ken Schwaber com o objetivo de conduzir o desenvolvimento de produtos complexos, podendo ser aplicado em projetos de qualquer tamanho e natureza.

O Scrum enquanto método de trabalho, dita sequências iterativas e incrementais divididas em fases denominadas Sprints, que por sua vez são “espaços de tempo” que tem seu fim caracterizado pela entrega de uma parte do produto. O framework possui um conjunto de regras divididas em papéis, artefatos e atividades, sendo respectivamente: as atribuições dos membros da equipe, os documentos relacionados ao produto em construção e as reuniões da equipe para alinhamento e feedback sobre o produto.

Uma iteração tem como propósito a geração de um incremento de produto, ou seja, uma parte do projeto que possa ser utilizado e/ou validado pelo cliente. As etapas que compõem uma iteração são, em ordem: a geração do backlog do produto; a reunião de plane-

jamento da sprint; a geração do backlog da sprint (que são as atividades que compõem o incremento a ser entregue) e a execução da sprint, com duração de duas a quatro semanas e contendo reuniões diárias de feedback e alinhamento. Quando a sprint chega ao final, é realizada uma reunião de retrospectiva da sprint, onde são revistos todos os itens planejados para entrega, o que foi bem executado e os itens que precisam ser corrigidos ou melhorados.

As informações levantadas até aqui concluem a análise da realidade observada consolidando os pontos a serem verificados na proposta de melhoria e avaliação do sistema objeto deste estudo de caso nas seções subsequentes.

6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA

O software objeto deste estudo, Gitlab se apresenta como uma ferramenta de apoio para todo o ciclo de DevOps, que é a prática de engenheiros de operações e desenvolvimento que participam juntos em todo ciclo de vida do serviço, desde o projeto, passando pelo processo de desenvolvimento até o suporte de produção (Mueller, 2010). Esta estratégia de unificar os departamentos de TI resulta em um ciclo de vida composto pelas atividades de: gerenciamento, planejamento, criação, verificação, empacotamento, entrega, configuração, monitoramento e segurança.

O sistema é distribuído em duas modalidades: SaaS (Software como Serviço), hospedada pelo Gitlab (gitlab.com) e Self-Managed (em tradução livre, auto gerenciada), hospedada em sua própria instância local. A avaliação foi realizada utilizando a versão 11 CE (Community Edition), com a licença básica (Core) que não possui custos iniciais e permite ser atualizada para versões pagas e com mais recursos, sem comprometer a os dados da instalação atual.

6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA

O Gitlab é estruturado sobre o repositório de códigos Git que é um sistema de controle de versões distribuído e de código aberto, desenvolvido para suportar modelos de desenvolvimento distribuídos. Oferece suporte a times remotos e práticas não lineares de desenvolvimento (Gitlab, 2019). A maioria dos cadastros do sistema podem ser realizados em texto simples ou utilizando a linguagem Markdown para formatação, a qual permite interligar qualquer documento ou informação no sistema como acontece em uma Wiki.

Além do sistema de versionamento, são disponibilizadas várias ferramentas integradas, as quais destacamos abaixo:

Autenticação e Autorização

O sistema possui um módulo de cadastro de usuários próprio além de permitir a integração com provedores de autenticação externos como: LDAP, OmniAuth, Smartcard entre outros. Habilitando desta forma, a possibilidade de integração com os usuários de redes internas das instituições ou redes sociais para controle dos acessos ao sistema.

Grupos e Subgrupos

Os repositórios de código são tratados pelo sistema como projetos e cada repositório pode ser organizado em grupos e subgrupos. Este tipo de organização facilita a distribuição e o controle dos usuários por projetos. Por padrão, um usuário com acesso a um grupo, possui acesso a todos os subgrupos e projetos relacionados.

Etiquetas (Labels):

O sistema disponibiliza um cadastro de etiquetas que podem ser associadas a cada grupo, subgrupo, projeto ou tarefa e respeita a hierarquia de cadastramento. Ou seja, uma etiqueta cadastrada em um grupo, fica disponível para todos os seus subgrupos e projetos de forma automática, sendo possível também, cada grupo ou projeto possuir suas próprias etiquetas. Este recurso dá flexibilidade de poder tratar projetos ou subgrupos com regras de classificação diferentes das de seus grupos pais. Existem ainda, um padrão de cadastro de etiquetas chamado de “etiquetas de escopo” (scope labels), definido pela separação de duas palavras pelos os caracteres “::”. Ex.: “gerencia::aprovado”, “gerencia::revisar” etc. Estas definições tem a finalidade de agrupar etiquetas e podem ser utilizadas para controle de fluxos de aprovações quando inseridas direto em uma tarefa.

Marco de Entrega (Milestone):

O cadastro de marcos é composto por uma descrição e datas de início e fim e tem a finalidade de agrupar uma ou mais tarefas (issues). As atividades podem ser filtradas pelo marco em questão sendo uma forma de controlar prazos de entrega e base para a montagem e visualização de um cronograma.

Tarefas (Issues):

Este módulo disponibiliza um cadastro de atividades, podendo ser casos, problemas, erros etc. Permite utilizar modelos predefinidos que são carregados no momento do cadastro ou em sua edição. As tarefas são compostas pelos atributos: título, descrição (*permitindo formatação em Markdown referenciando outras áreas do sistema*), responsável (*atribuição a um usuário do sistema*), marco de entrega, etiquetas (*uma ou mais*), data de entrega e registro de tempo/esforço despendido na mesma (*time tracker*). Os atributos acima listados podem ser utilizados para filtragem das atividades no sistema.

As tarefas também são compostas por um registro de atividades que pode ser inserido por cada usuário que tiver acesso a tarefa, gerando uma linha do tempo das iterações com a mesma (semelhante aos comentários dos cartões do software Trello). Nesta linha do tempo, também são inseridas de forma automática pelo sistema, informações de commits que referenciem a tarefa e utilizadas etiquetas para controle de fluxos de aprovações e validações.

Quadro Kanban (Board):

O Board do sistema permite a visualização das tarefas (issues) formatadas como um quadro Kanban. São permitidas filtragens para as tarefas por vários critérios como: etiquetas, usuários associado, autor, marco de entrega, emojis de reações etc.

Existem duas colunas/listas fixas no board, são elas Aberto (Open) e Fechado (Closed), representando um fluxo mínimo de controle das atividades. Fluxos intermediários podem ser incluídos conforme for a necessidade do projeto. O sistema utiliza o próprio cadastro de etiquetas para incluir colunas/listas adicionais que representem fluxos de transição de estados das atividades.

Wiki:

Cada projeto possui uma Wiki própria integrada ao mesmo. A Wiki pode ser utilizada para documentação geral do projeto, possuindo ciclo de vida e histórico de alterações próprio, vinculado ao repositório principal (o projeto em si). É possível referenciar arquivos do repositório principal e em outros repositórios estendendo a característica principal de uma wiki a todo o sistema e seus projetos.

API:

O sistema conta ainda com uma API aberta podendo ser utilizada para integrações, consultas etc. No perfil de cada usuário podem ser criados tokens de acesso que permitem o uso das APIs do sistema. Este recurso permite estender as funcionalidades do sistema

com integrações a soluções de terceiros sem a necessidade de intervenções no código do Gitlab.

O Gitlab possui muitos outros recursos aderentes ao ciclo de vida de DevOps. Os listados acima, correspondem a uma parte destes, que durante a análise permitiu avaliar a ferramenta como uma solução de gerenciamento de projetos.

6.2 RESULTADOS ESPERADOS

A implantação de um sistema como o Gitlab pode representar uma base sólida para se estruturar todo um ciclo de controle de projetos em uma fábrica de software. Todas as ferramentas do sistema foram desenvolvidas de forma independente, interligando-se caso seja necessário e/ou configurado para o tal. Isso permite o uso do sistema em vários níveis, seja como simples repositório de códigos até como uma solução robusta que controle fluxos de projetos, documentações, de controle de códigos e até automatizações de deploys.

Abaixo apresentamos os passos que nortearam os testes no sistema como sugestão de utilização com o foco no gerenciamento de projetos, em seguida descrevemos os resultados obtidos com o modelo proposto.

Etapas realizadas na avaliação do software:

- **Cadastro de um grupo** permitindo a associação do repositório do projeto (*este grupo pode representar a empresa ou um setor específico da mesma*).
- **Cadastro do projeto** (*dentro do grupo*), informando a descrição e selecionando o nível de visibilidade como Privado, que só permitirá visualização aos membros do projeto;
 - *Neste momento é oferecida a opção de inicializar o repositório com um arquivo README. Este arquivo é um indexador padrão em repositórios Git que é carregado como visualização padrão do repositório e é utilizado para descrever informações sobre o projeto.*
- **Criada a estrutura de diretórios** no repositório:
 - atas - *Atas de reuniões*
 - documentos-de-projeto - *Documentos formais do projeto, como termos de aceitação, plano de gerenciamento do projeto, etc.*
 - outros - *Quaisquer outros documentos utilizados no projeto, como planilhas, referências externas, etc.*
- **Cadastro dos Marcos de Entregas (Milestones)** que serão associados as atividades do projeto.

- **Cadastro das Etiquetas (labels)** para controle dos fluxos intermediários:
 - “lista::Em andamento” - *Atividades em andamento.*
 - “lista::Impedimento” - *Atividades com alguma pendência ou impedimento.*
 - “lista::Homologação” - *Atividades em homologação pelo cliente.*
- **Adicionar as listas criadas no Quadro Kanban**, deixando o fluxo de status das atividades do projeto da seguinte forma:
 - | OPEN | Em andamento | Impedimento | Homologação | CLOSED |
 - As atividades transitarão entre os status acima citados até sua conclusão.
- **Cadastro das Atividades** do projeto:
 - Associamos as atividades aos Marcos de Entrega (Milestones);
 - Associamos aos usuários responsáveis;
 - Definimos a data de conclusão da atividade;

Concluídas as etapas acima descritas, o projeto encontra-se minimamente estruturado para ser gerido no sistema, permitindo o controle das atividades de forma transparente, integral e com rastreabilidade.

O repositório pode ser utilizado para arquivar documentos específicos relacionados ao ciclo de vida do projeto em formato Markdown, o qual permite por se tratar em sua fonte, de texto simples, um controle fino das alterações realizadas no mesmo, utilizando o próprio motor de versionamento do repositório. O formato Markdown possibilita relacionar por meio de links, outros arquivos armazenados no sistema ou fora dele, entregando assim um resultado semelhante ao funcionamento de uma Wiki.

Durante esta avaliação o módulo da Wiki foi utilizado apenas para indexação da documentação do projeto, estando a mesma armazenada no próprio repositório. Contudo, este recurso poderia ser utilizado para gerar toda a documentação do projeto, mantendo um bom nível de controle e manutenção.

Os eventos do framework Scrum podem ser documentados em vários locais do sistema, desde arquivos específicos armazenados e versionados no repositório, na própria Wiki ou no cadastro dos Marcos de Entregas (Milestones), os quais numa abordagem de controle de Sprints, podem armazenar as informações de Planejamento e Revisão de forma a permitir a consulta posterior das informações tratadas pelo time.

Durante os testes com a versão CE (Community Edition) do sistema, foi verificado que a combinação do uso de Marcos de Entregas, com suas respectivas datas de início e fim, devidamente associados a um conjunto de tarefas com datas de entregas definidas, embora seja simples de realizar manutenções, não disponibiliza uma forma simples de visualizar um

cronograma geral do projeto. Os cadastros de Roadmap e Épicos, que se relacionam com os Milestones e tarefas, só estão disponíveis em versões pagas do Gitlab.

Entretanto, numa rápida pesquisa realizada na internet, chegamos ao sistema GanttLab (<https://www.ganttlab.org/>), de código aberto, mantido na própria estrutura do gitlab (<https://gitlab.com/ganttlab/ganttlab-live>), que permite por meio de consultas a API do sistema, a geração automática de um gráfico de Gantt, lendo as tarefas cadastradas no sistema, representando assim, o cronograma do projeto. Tal visualização é conseguida com a simples inclusão de marcações específicas na descrição das tarefas (“*GanttStart: aaaa-mm-dd*” e “*GanttDue: aaaa-mm-dd*”).

É possível visualizar em um quadro kanban único, todas as atividades de vários projetos ao mesmo tempo, bastando selecionar a visão do board a partir de um grupo. Em implantações que os grupos representem os setores de uma instituição, este recurso permitirá visualizar o fluxo atual de todas as atividades da empresa em um painel único, facilitando prioridades e auxiliando a tomada de decisão.

Durante os testes o sistema se apresentou estável, sem erros, com uma interface amigável, responsiva (*permitindo o uso no navegador de um smartphone sem complicações*) e bastante promissor quanto as possibilidades de automação de tarefas por meio do módulo de Integração Contínua – CI.

6.3 VIABILIDADE DA PROPOSTA

A análise realizada demonstra que o sistema se apresenta viável para utilização como ferramenta de gerenciamento de projetos já em sua versão livre, que não apresenta custos iniciais relacionados a licenciamentos. Vale ressaltar que outras opções de licenciamento do mesmo sistema, entregam ainda mais recursos que somados disponibilizam um maior controle sobre o gerenciamento de projetos. Entretanto, fica claro que os recursos disponíveis na versão atual são suficientes para demonstrar a viabilidade de seu uso.

Quanto ao uso com metodologias ágeis, o sistema não apresenta recursos/módulos específicos de controle implementados para o tal. Ainda assim, a forma descentralizada com que o mesmo foi construído, permite que seja utilizado para controlar praticamente qualquer fluxo de trabalho. Seu conjunto de ferramentas habilita o sistema para utilização com Scrum, Kanban, RUP ou outra metodologia qualquer.

No âmbito de uma fábrica de software, exemplo deste estudo, os demais recursos do sistema agregam controle total ao ciclo de vida de aplicações, permitindo automatizações alinhadas com a cultura de DevOps.

7 CONSIDERAÇÕES FINAIS

Este estudo teve como finalidade de avaliar o sistema Gitlab enquanto ferramenta de gestão de projetos, elencando os recursos disponíveis e apresentando uma sugestão de utilização do sistema. Ficou demonstrado a aderência à lista dos recursos desejados pelas partes integrantes da instituição, bem como as possibilidades de agregar outros ganhos relacionados aos fluxos de trabalhos existentes em uma fábrica de software.

Vale ressaltar que, existem várias possibilidades de integração com o Gitlab, que por meio de sua API, permitiriam a construção de soluções que atendam a necessidades específicas de outras instituições.

Por fim, fica evidenciado neste trabalho que o Gitlab se demonstrou, de acordo com os testes realizados, uma ferramenta viável para gerenciar projetos de software sejam os modelos empregados ágeis ou tradicionais, bastando para isso que sejam definidos os fluxos de trabalho e os padrões de artefatos e armazenamento desejados.

REFERÊNCIAS

AGILE MANIFESTO, 2001. Disponível em <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 20 de Março de 2019.

CHACON Scott; STARUB Ben. **Pro Git: Everything you need to know about Git - Apress**; 2nd ed. 2014 edition (12 Nov. 2014) (1600).

FRANKENTHAL, Rafaela; 2017. **Entenda o que é a escala Likert e como aplicá-la**. Disponível em <<https://mindminers.com/blog/entenda-o-que-e-escala-likert/>>. Acesso em 05 de Junho de 2019.

GITLAB, 2019. Disponível em <<https://about.gitlab.com/>>. Acesso em 22 de Abril de 2019.

HELDMAN, Kim. **Gerência de Projetos : fundamentos : Um guia prático para quem quer certificação em gerência de projetos**; Kim Heldman; tradução de Luciana do Amaral Teixeira. - Rio de Janeiro: Elsevier, 2005 – 5ª reimpressão.

KROLL, P. ; KRUCHTEN P. **The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP**. Addison Wesley 2003.

MACHADO, Cristiane Salvan et al. **Trabalhos acadêmicos na Unisul**: apresentação gráfica. Palhoça: Ed. Unisul, 2012.

MUELLER, Ernest, 2010. **What is DevOps**. Disponível em <<https://theagileadmin.com/what-is-devops/>>. Acesso em 04 de Junho de 2019.

RAUEN, Fábio José. **Roteiros de investigação científica. Tubarão: Unisul, 2002.**

SCHWABER Ken; SUTHERLAND Jeff. **Guia do Scrum: Um guia definitivo para o Scrum: As regras do Jogo**. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 27 de Abril de 2019.

SCHWABER, Ken. **Agile Project Mangement with Scrum**. Microsoft Press, 2014.

SUTHERLAND, Jeff. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo** – Jeff Sutherland; tradução de Natalie Gerhardt. São Paulo: Leya, 2014. 240 p.

VARGAS, RICARDO VIANNA. **Manual prático do plano de projeto: utilizando o PM-BOK Guide / Ricardo Viana Vargas**. - 4. ed. - Rio de Janeiro: Brasport 2009.