

APLICAÇÃO WEB: ScruManager

Elvis Raul Estevan¹ – Faculdade de Tecnologia de Carapicuíba

Matheus Muzy Talpe² – Faculdade de Tecnologia de Carapicuíba

Willy Henrique da Silva³ – Faculdade de Tecnologia de Carapicuíba

Prof. Jadir Custodio Mendonça Junior – Faculdade de Tecnologia de Carapicuíba – *e-mail*:

jadir.junior@fatec.sp.gov.br

Profa. Silvia M^a. Farani Costa – Faculdade de Tecnologia de Carapicuíba – *e-mail*:

silvia.costa01@fatec.sp.gov.br

RESUMO

O mundo do desenvolvimento de *software* está vivendo o caos. Inúmeros projetos são interrompidos, atrasam ou são entregues contendo diversos problemas que muitas vezes, impedem que o produto final tenha algum valor palpável para o cliente. Este caos gera inúmeros prejuízos, tanto às empresas que desenvolvem *softwares*, quanto aos clientes que muitas vezes gastam grandes quantias de dinheiro para ter um sistema que não os atende. O desenvolvimento de sistemas realizado internamente em uma empresa, não é exceção a estes problemas. A utilização dos métodos ágeis surge como meio para aumentar a eficiência no desenvolvimento de *softwares*, buscando trazer maior adaptabilidade, levando em conta as numerosas mudanças que ocorrem nos requisitos no decorrer de um projeto. Este é o caso da aplicação web ScruManager desenvolvido para auxiliar a empresa CNTLog, que atualmente enfrenta diversos desafios em seu processo interno de desenvolvimento de *software*. O foco para desenvolvimento da aplicação foi utilizar-se das vantagens do método ágil *Scrum* ao mesmo tempo que atende à necessidade específica da empresa de realizar um controle de documentação de *softwares*.

Palavras-chave: *Software. Projeto. Métodos Ágeis. Aplicação Web.*

ABSTRACT

The world of software's development is in chaos. Countless projects are interrupted, delayed or delivered having many problems that many times prevents the final product from having any significant value to the client. The internal system development made by a company is no exception to these problems. The utilization of agile arose as a mean to increase efficiency in software development, searching to bring better adaptability, considering the many changes that occur in the requirements during a project. This is the case of web application ScruManager, develop to assist the CNTLog company, that nowadays faces many challenges in its internal software development process. The development focus of the application was to use the advantages of Scrum at the same time that meets the company's specific needs to control the software documentation.

Keywords: *Software. Project. Agile. Web Application.*

¹ - Analise e Desenvolvimento de Sistemas – *e-mail*: elvis.estevan65@gmail.com

² - Analise e Desenvolvimento de Sistemas – *e-mail*: matheusmtalpe@gmail.com

³ - Analise e Desenvolvimento de Sistemas – *e-mail*: willyy@hotmail.com.br

1 INTRODUÇÃO

Na atual sociedade, as empresas especializadas em desenvolvimento de *software* estão em uma constante busca por metodologias que permitam a melhor utilização possível do tempo e recursos disponíveis, viabilizando assim, a entrega de um sistema adequado à necessidade do cliente, que corresponda às suas expectativas e dentro do prazo. Infelizmente esta situação na maioria das vezes não ocorre atualmente.

De fato, qualquer produto entregue inadequadamente irá gerar prejuízos à empresa desenvolvedora, seja por tempo extra gasto com refação, cancelamento de projetos que não mais atendem às necessidades e custos de uma maneira geral.

De acordo com o *Chaos Report* elaborado pelo *Standish Group* (2015), que dispõe de um levantamento em mais de 50.000 projetos em 1.000 organizações, os projetos de *software*, em sua maior parte, fracassam ou sofrem de inúmeros problemas (atrasos, *bugs*, não atendem às necessidades do cliente). Na figura 1, observa-se que em 2015, dentre os projetos levantados durante o *Chaos Report*:

- Apenas 29% foram considerados um sucesso;
- A maior parte, 52%, foram entregues contendo problemas como *bugs* ou não atendiam completamente as necessidades do cliente;
- Uma fatia considerável, 19%, foram interrompidos em razão de atrasos que estouraram o *budget* do projeto ou por não mais atenderem às necessidades dinâmicas dos clientes.

Figura 1 – resolução moderna para todos os projetos

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.

Fonte: Standish Group, 2015 *Chaos Report*.

O departamento interno de desenvolvimento de *software* da CNTLog, encontra-se em situação similar há anos. Considerando este cenário, pretende-se reverter esta situação por meio da utilização de uma aplicação *web* que aplique os princípios do método ágil *Scrum*, ao

passo que permite realizar o controle de documentação do projeto que será exigido internamente na empresa.

1.1 Objetivo Geral

O objetivo geral deste projeto é desenvolver uma aplicação *web* para facilitar a utilização do *Scrum* e seus conceitos chave no desenvolvimento de *software* da empresa CNTLog, e assim auxiliar os desenvolvedores a cumprir os novos processos a serem aplicados na empresa.

Tais processos exigem que, para que um *software* seja considerado finalizado, haja disponível a especificação de *backlog* dos sistemas e evidências de testes a fim de garantir a qualidade das entregas.

1.2 Objetivos Específicos

A fim de atingir o objetivo geral, os seguintes objetivos específicos devem ser cumpridos:

- Aplicar alguns conceitos chaves do *Scrum*, a fim de aumentar a capacidade adaptativa da empresa às constantes mudanças de requisitos de *software* que ocorrem durante o ciclo de vida de um projeto.
- Disponibilizar este controle de entregas a fim de garantir certo nível de qualidade, deixando em aberto o tipo de documentação a ser entregue para não comprometer a flexibilidade da aplicação do método ágil e possíveis mudanças nos critérios de aceitação da empresa.
- Permitir a visualização de qual membro realizou a entrega e o *status* das *Sprints* do projeto.

1.3 Justificativa

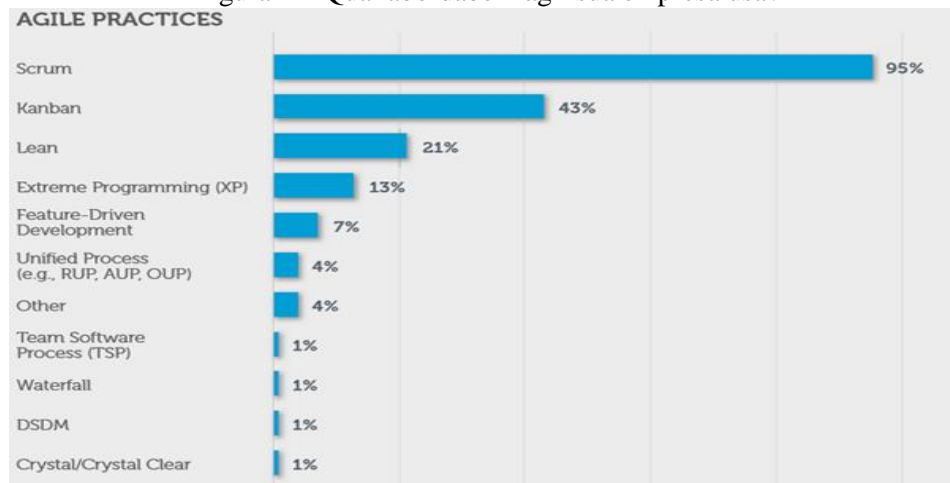
O desenvolvimento da aplicação *web* ScruManager é justificado por uma situação real de caos, no qual a área interna de desenvolvimento de *software*, que há na empresa CNTLog, enfrentava constantes atrasos, entrega de *softwares* que não correspondiam às necessidades e com alta incidência de *bugs*. Mediante diversas reuniões entre os membros da equipe de desenvolvimento de *software*, foi constatado que em grande parte estes problemas acontecem em razão de deficiências no controle interno de documentações do projeto e a falta de adequação às mudanças de escopo que ocorriam no decorrer do projeto.

A partir destas reuniões, por motivos de grande aceitação no mercado, o *Scrum* foi escolhido como método ágil a ser aplicado na empresa e que a partir de sua aplicação, seria exigida pela liderança da empresa uma documentação mínima para cada projeto realizado ser considerado como terminado.

Então, foi proposta a criação de um *software* gratuito que ajude a equipe de desenvolvimento aplicar conceitos do *Scrum*, e que deverá também garantir que a documentação que será tomada como padrão seja feita.

Entre os diversos métodos ágeis disponíveis no mercado, escolheu-se utilizar o *Scrum*, pois tem se mostrado como sendo uma forte tendência, conforme figura 2.

Figura 2 – Qual abordagem ágil sua empresa usa?



Fonte: ScrumAlliance, *The 2015 State Of Scrum Report*.

Este gráfico representa uma pesquisa realizada em 2015 com quase 5.000 indivíduos, representando 108 países e mais de 14 indústrias e aborda projetos de desenvolvimento de uma maneira geral. O mesmo demonstra a porcentagem destas pessoas que utilizam o *Scrum* diante de outras metodologias de desenvolvimento.

Esta mesma pesquisa (*The 2015 State of Scrum Report*), apresenta uma taxa de 62% de sucesso na entrega dos projetos utilizando *Scrum*. Em um rápido comparativo com a taxa de sucesso de 29% na pesquisa (*Chaos Report*) representada pela Figura 1, que não está preocupada em especificar quais metodologias são utilizadas e sim o desenvolvimento de *software* no geral. Nota-se que *Scrum* possui uma boa incidência de sucesso (*software* entregue no prazo, com boa qualidade e que atenda às necessidades propostas).

Partindo do princípio que dentre as metodologias ágeis disponíveis no mercado, o *Scrum* possui ampla utilização, a elaboração do sistema ScruManager teve como propósito,

identificar as principais diretrizes do método ágil denominado *SCRUM* e aplicá-las à um *software* livre, ao passo que prioriza a entrega de documentação para atender às necessidades específicas da CNTLog, e por meio destes documentos mitigar problemas relacionados à qualidade de *software*.

1.3.1 Softwares Similares

Já existem *softwares* similares ao ScruManager disponíveis no mercado, mas além da grande maioria não ser gratuita, não possuem o diferencial desejado que é o foco em documentação.

A tabela 1 expõe um comparativo entre o ScrumManager e *softwares* similares bem conhecidos e aceitos no mercado.

Tabela 1 – Comparação de softwares similares

<i>Software</i>	<i>Web</i>	Acompanhamento de Sprints	Foco em Documentação	Atas de Reunião	Gratuito
Trello	X	-	-	-	X
ScrumDo	X	X	-	-	-
Jira	X	X	-	-	-
ScruManager	X	X	X	X	X

Fonte: Do próprio autor, 2017.

Dos *softwares* apresentados, apenas o Trello e o ScruManager são gratuitos. Porém o Trello não possui acompanhamento de *Sprints* conforme os demais, tal característica fez com que não fosse considerado para utilização na CNTLog.

Os demais *softwares* (ScrumDo e Jira) foram descartados por não serem gratuitos e não possuírem o foco desejado na documentação.

Em virtude das características apresentadas, e o fato da empresa desejar um *software* que seja gratuito e que possa ter seu código fonte alterado livremente para se adequar às mudanças e melhorias futuras, além de possuir o foco desejado no controle de documentação e atas de reunião que os demais não possuem, optou-se por desenvolver o ScruManager.

2 SCRUM

Nesta seção, busca-se demonstrar o que é o *Scrum*, suas origens e como utilizá-lo, a fim de deixar claro seus conceitos e a partir deles, entender o funcionamento do ScruManager que busca aplicá-los de modo que fiquem alinhados com a necessidade específica de documentação da CNTLog.

A origem do termo “ágil”:

...data de uma reunião de 2001, na qual eu e 16 outros líderes de desenvolvimento de software escrevemos o que se tornou conhecido como “Manifesto Ágil”. Nele declaramos os seguintes valores: pessoas em vez de processos; produtos que realmente funcionem em vez de documentação dizendo como o produto deveria funcionar; trabalhar com os clientes em vez de negociar com eles; e responder às mudanças em vez de seguir um plano. *Scrum* é a estrutura que eu construí para colocar esses valores em prática. Não existe uma metodologia (SUTHERLAND, 2014).

Durante uma reunião entre líderes de desenvolvimento de *software*, foi oficializado o “Manifesto Ágil” no qual se oficializou os valores a serem seguidos pelos métodos ágeis. O *Scrum* é um dos vários métodos criados para aplicar estes valores no trabalho do dia a dia das equipes de desenvolvimento.

O *Scrum* foi criado por Jeff Sutherland e Ken Schwaber no começo dos anos 1990, fortemente influenciados pelo Sistema Toyota de Produção, com a intenção de contornar os problemas que ocorriam com a utilização dos métodos em cascata/tradicionais.

O termo “*Scrum*” origina-se do *Rugby*, no qual é um método de reinício de jogada que ocorre após alguma infração ou quando a bola sai de campo, no qual os jogadores de ambos os times se reúnem e entram em formação, cada um com suas atribuições específicas, cooperando em prol do objetivo do time.

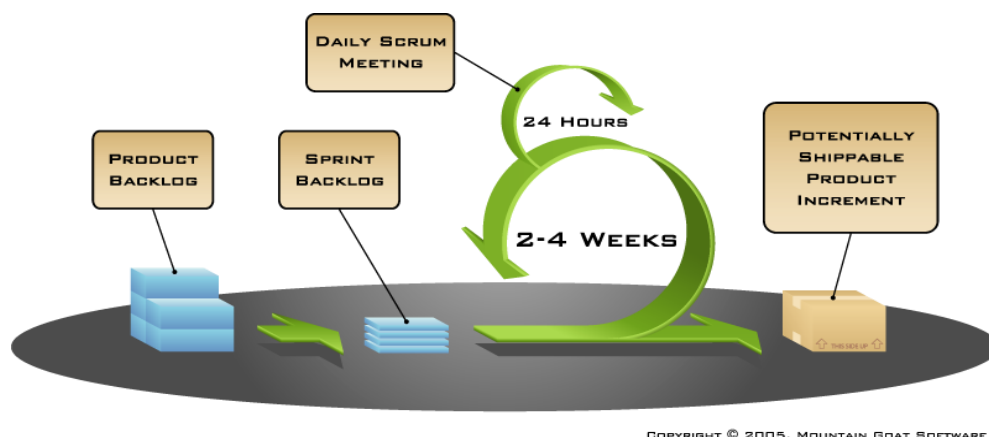
...eu chamei de “*Scrum*” essa estrutura de desempenho de equipe. O termo vem do jogo de rúgbi e se refere à maneira como um time trabalha junto para avançar com a bola no campo. Alinhamento cuidadoso, unidade de propósito, clareza de objetivo, tudo se unindo. Trata-se de uma metáfora perfeita para o que uma equipe deseja fazer (SUTHERLAND, 2014).

2.1 Ciclo de vida do *Scrum*

O entendimento do ciclo de vida do *Scrum* é muito importante para sua utilização e para compreender o ciclo de vida de um projeto dentro da aplicação *ScruManager*, que aplicará estes conceitos.

A figura 3 demonstra o ciclo de vida do desenvolvimento por intermédio do *Scrum*. No início de cada projeto, desenvolvedores, clientes e demais interessados se reúnem para definir o *Backlog* do projeto. O *Backlog* consiste em uma lista que contenha todos os requisitos que o produto deve apresentar, bem como definição de algumas datas de entrega de acordo com as prioridades do cliente.

Figura 3 – Ciclo de vida do Scrum⁴



Fonte: Mountain Goat *Software*, 2005.

Após a definição do *Backlog* do produto, a equipe deve realizar a definição do *Sprint Backlog* que consiste em uma lista de atividades que serão realizadas na próxima *Sprint*. A meta final de cada *Sprint* é originada a partir do próprio *Product Backlog*.

Neste momento também é realizado o planejamento da *Sprint*, na qual são definidas as atribuições de cada membro e padrões a serem adotados.

Ao final de cada *Sprint* deverá ser realizada uma entrega que tenha valor, funcionalidade agregada ao cliente, o qual pode acatar ou rejeitar os resultados. Também deverá ser realizada uma reunião com o objetivo de avaliar as dificuldades e lições aprendidas da *Sprint* concluída.

No decorrer da reunião de entrega com o cliente, pode ser observada a necessidade de alterações no *Backlog* do produto que deverão ser realizadas e reavaliadas as prioridades.

Estas características trazem grande adaptabilidade e capacidade de autocorreção ao *Scrum*, permitindo sempre manter o produto alinhado às expectativas do cliente, aumentando o nível da satisfação e qualidade do *software*.

Esse ciclo se repete até que todos os requisitos do *Backlog* do produto sejam atendidos.

2.2 Equipe

Nesta seção serão apresentados todos os componentes que são necessários para a realização do *Scrum*, de acordo com os parâmetros definidos por Ken Schwaber e Jeff Sutherland em “**Guia do Scrum**”.

⁴ Disponível em: <<https://www.mountaingoatsoftware.com/agile/scrum/resources/images>>. Acesso em 02 de Junho de 2017.

2.3 *Product Owner*

Geralmente é o próprio cliente ou representante do mesmo. Responsável pela criação e gerenciamento do *Backlog* de Produto, ou seja, definem as funcionalidades do produto, prioridades, datas e valida as entregas.

2.4 *Scrum Master*

Deve garantir que o *Scrum* seja totalmente compreendido por todos os envolvidos e está sendo devidamente aplicado.

- Responsável pela disseminação de conhecimento em relação ao *Scrum* a todos os envolvidos (*Product Owner*, *Team* e Organização).
- Promove a colaboração entre os envolvidos.
- Protege a equipe de interferências externas.
- Colabora na remoção de obstáculos.
- Estimula a comunicação entre os envolvidos, agindo como um facilitador.

2.5 *Team*

É o time multifuncional que de fato desenvolve o sistema. As responsabilidades do mesmo, nunca recaem sobre um indivíduo, mas sempre sobre o time todo. O tamanho recomendado de um time é compreendido entre 3 e 9 integrantes.

- Devem ser auto gerenciados.
- Realizam as estimativas.
- Garantem a qualidade do produto.
- Implementam a aplicação.

2.6 *Artefatos*

O *Scrum* utiliza um escopo obrigatório de documentos bem reduzido, compreendido apenas entre os dois *backlogs* (*Product* e *Sprint*) e o *Task Board*.

2.7 *Product Backlog*

Documento que descreve tudo que o *Product Owner* e o *Team* compreendem que deve fazer parte do *software* a ser desenvolvido.

Pode ser escrito de diversas maneiras, desde uma simples lista enumerada com tudo que o cliente deseja ser possível fazer, até algo mais elaborado como uma lista contendo o código de requisitos funcionais, porém deve sempre conter o nível de prioridade, estimativa e

valor. Quanto maior o nível de prioridade, maior deve ser o detalhamento da funcionalidade a fim de garantir maior precisão nas estimativas realizadas pelo *Team*.

Lembre que o *Backlog* de Produto é dinâmico, e está sempre evoluindo.

2.8 *Sprint Backlog*

Este artefato contém todas as tarefas a serem realizadas durante a *Sprint*. Apesar de permitido, não é recomendável que o mesmo sofra alterações até o fim da *Sprint* para que haja coerência nos prazos. Deve conter detalhes o suficiente para que o senso de progresso a cada atividade concluída possa ser mensurado.

2.9 *Task board*

Quadro no qual os membros do time registram suas atividades realizadas, que estão desenvolvendo, e que estão pendentes. Esta técnica facilita o acompanhamento no decorrer da *Sprint* e avalia os pontos de atenção, no qual outro membro da equipe pode atuar. Geralmente os registros são realizados por *post-it* ou escritos diretamente no quadro.

2.10 *Sprint*

A *Sprint* é o cerne do *Scrum*. Trata-se de cada iteração planejada que resulta na entrega de uma funcionalidade, de acordo com as prioridades definidas pelo *Product Owner*.

- Cada *Sprint* não deve durar mais do que um mês a fim de não prejudicar a previsibilidade e capacidade de adaptação do projeto.
- Devem-se evitar mudanças que coloquem em risco o objetivo da *Sprint*.
- A *Sprint* pode ser cancelada caso o objetivo se torne incoerente com a realidade.
- Somente o *Product Owner* pode cancelar uma *Sprint*.

2.11 Planejamento da *Sprint*

É realizado antes do início da *Sprint*, durante o planejamento é gerado o *Sprint Backlog* que contém todas as atividades a serem realizadas.

- Todas as partes devem participar.
- O *Product Owner* define a meta da *Sprint* de acordo com as prioridades contidas no *backlog* do produto.
- A partir da meta definida pelo *Product Owner*, o time gera o *Sprint Backlog*.
- Não deve durar mais que 8 horas.

2.12 Reunião diária

Realizada diariamente no decorrer da *Sprint*, deve ser curta e objetiva a fim de sincronizar as atividades dos membros da equipe.

- Deve durar no máximo, 15 minutos.
- Cada membro deve elencar o que fez ontem, o que fará hoje e possíveis obstáculos ao cumprimento da meta.

2.13 Revisão da *Sprint*

Reunião informal realizada ao fim da *Sprint*, que busca fazer com que os integrantes da equipe corroborem tudo que foi feito, realizando uma apresentação ao *Product Owner*.

- Deve durar, no máximo 4 horas.
- A equipe deve discutir as dificuldades e o que funcionou bem durante a *Sprint*
- Realiza-se uma discussão acerca do *Backlog* do Produto, orientada pelo *Product Owner*, afim de avaliar a situação atual e possíveis alterações ao *backlog*.
- O *Product Owner* pode aceitar ou rejeitar a *Sprint*.
- O resultado final desta reunião deve ser um *backlog* atualizado e pronto para o planejamento das próximas *Sprints*.

2.14 Retrospectiva da *Sprint*

Esta reunião ocorre entre a reunião de revisão da *Sprint* e o planejamento da próxima. Possui o intuito de elencar melhorias à serem aplicadas na próxima *Sprint*.

- Deve durar no máximo, 3 horas.
- Devem participar o *Time Scrum* e o *Scrum Master*, enquanto a presença do *Product Owner* é opcional (geralmente ocorre a convite).
- Apesar deste ser o momento dedicado para tal, melhorias podem ser detectadas e aplicadas a qualquer momento no decorrer de um projeto.

3 DESENVOLVIMENTO

Esta seção irá descrever brevemente o caso da CNTLog e o processo de desenvolvimento da solução.

3.1 A CNTLog

A CNTLog é uma empresa que atua no mercado desde 2010. A empresa tem como foco as operações logísticas de empresas e-commerce. Atualmente, tem sua sede em Barueri, na grande São Paulo, e conta com uma base de aproximadamente 100 clientes.

Por ser especializada em operações de e-commerce, a CNT precisava de um *software* para gerenciamento e como não encontrou uma opção que atendesse suas necessidades de forma satisfatória partiu para o desenvolvimento de uma ferramenta própria. Assim nasceu a divisão de tecnologia do grupo: a CNT Tech.

O portfólio da divisão conta com dois sistemas: O WMS Thorpe e o TMS Scope. Ambos os sistemas possuem integrações com os principais ERPs do mercado e também uma API aberta para parceiros.

3.2 Situação Atual

Com o crescimento da empresa, a demanda por novas funcionalidades no sistema vem aumentando e a falta de documentação desses projetos de melhoria se torna um grande problema à medida que as novas funcionalidades devem ser entregues funcionando e atendendo aos requisitos especificados.

Hoje as solicitações chegam por meio de usuários internos e externos. Essas solicitações são controladas pelo líder da equipe de desenvolvimento. Esse controle é feito por meio de planilhas. Não existem documentos oficiais que explicitem os requisitos funcionais, a única documentação existente são os e-mails enviados pelos clientes com suas sugestões e trocados pela equipe. A partir daí a equipe desenvolve a solução e submete à validação do cliente.

Porém, como não são utilizadas técnicas de engenharia de *software*, o resultado acaba sendo, muitas vezes, o não atendimento ou atendimento parcial da necessidade do cliente.

3.3 Solução proposta

A solução proposta é implementar um sistema de informação, baseado na metodologia *Scrum*, para gerenciar o cronograma de desenvolvimento e entrega dos artefatos, assim como o armazenamento dos mesmos.

3.4 Características básicas

- Sistema *web*, facilitando o acesso.
- Cadastramento das tarefas e *upload* dos artefatos anexos.

- Criação e manutenção do histórico das tarefas, registrando quem as aplicou ações de alteração nas tarefas.

3.5 Regras de negócio

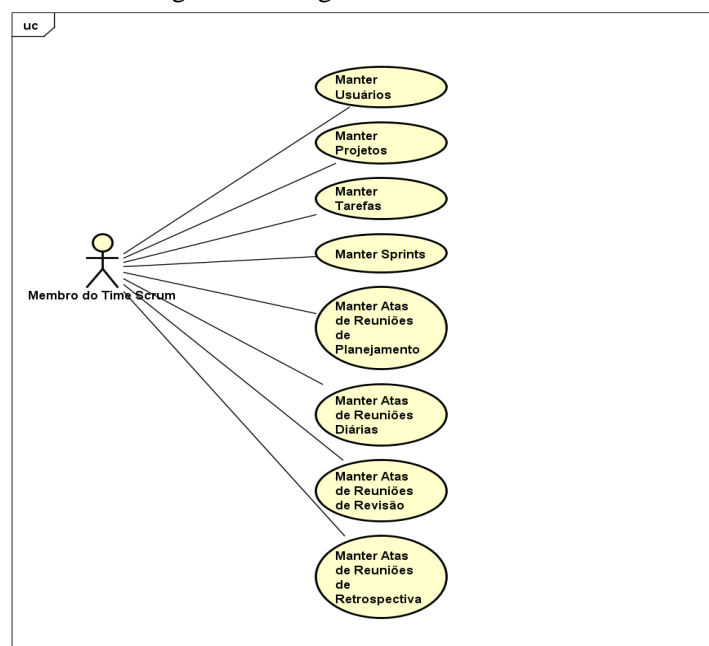
O Sistema ScruManager deve atender às seguintes regras de negócio:

- Todo projeto deverá ter membros para iniciar uma *Sprint*.
- As tarefas só devem ser alteradas de *status* de acordo com *workflow* pré-definido.
- Toda tarefa deve ter um nível de prioridade.
- Ao finalizar uma *Sprint* são obrigatórios os cadastros das atas de reuniões de retrospectiva e revisão da *Sprint* e os *status* das tarefas serão atualizados para “Resolvido”.
- Somente será permitido o cancelamento de uma *Sprint* em andamento, e todas as tarefas terão seus *status* atualizados para “Fechado”.

3.6 Casos de uso

A Figura 4 demonstra os principais casos de uso, que foram essenciais para fundamentar o desenvolvimento do ScruManager, pois definem suas principais funcionalidades ao passo que demonstram o que o usuário poderá esperar realizar no sistema.

Figura 4 – Diagrama de Caso de Uso



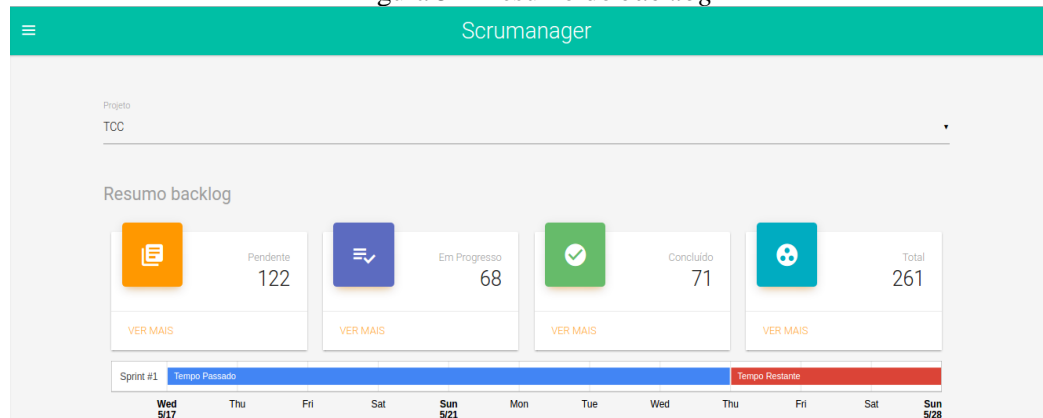
powered by Astah

Fonte: Do próprio autor, 2017.

4 RESULTADOS

O resultado obtido a partir da análise do caso da CNTLog e seu subsequente processo de desenvolvimento da solução, foi o *software* denominado ScruManager. A maioria das informações relevantes que o sistema apresenta ao usuário estão na tela principal de cada projeto. Tal tela é demonstrada na Figura 5.

Figura 5 – Resumo do *backlog*



Fonte: *Print screen* da aplicação ScruManager, 2017.

A Figura 5 demonstra a seção superior da tela principal da aplicação ScruManager, onde é apresentado um resumo da *sprint* em andamento. Neste resumo, há totalizadores que demonstram a quantidade de atividades da *sprint* em cada *status*. Tais *status* são divididos em:

- Pendente: atividades que ainda não foram iniciadas.
- Em progresso: atividades que já foram iniciadas e atribuídas, porém não foram concluídas pelo responsável.
- Concluído: atividades que já foram finalizadas por seus executores.
- Total: Total de atividades da *sprint* corrente.

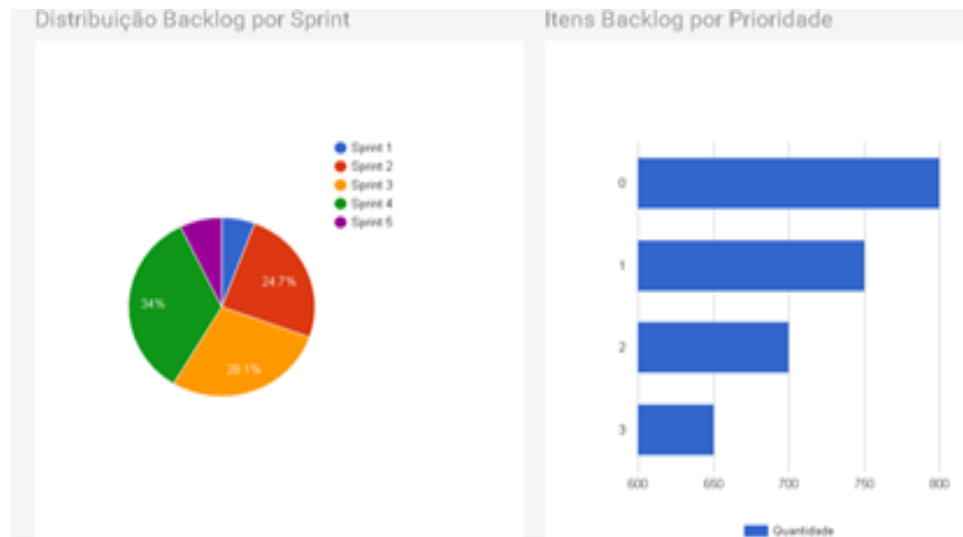
Abaixo dos totalizadores, encontra-se uma linha do tempo que contém as datas de início e fim (em negrito), tempo passado (azul) e o tempo restante (vermelho) da *sprint* atual.

A Figura 6 demonstra a seção seguinte, onde o sistema apresenta os gráficos que demonstram a distribuição dos itens do *backlog* do projeto por *sprint* e sua distribuição de acordo com os níveis de prioridade.

A Figura 7 demonstra o gráfico *Sprint Burndown* que é apresentado ao fim da tela principal, com este gráfico é possível rastrear as tarefas pendentes e o progresso real da *sprint*. No gráfico são estabelecidas duas linhas, uma representando o progresso ideal (vermelho) da *sprint* em andamento e outra representando o real progresso (azul) até o momento.

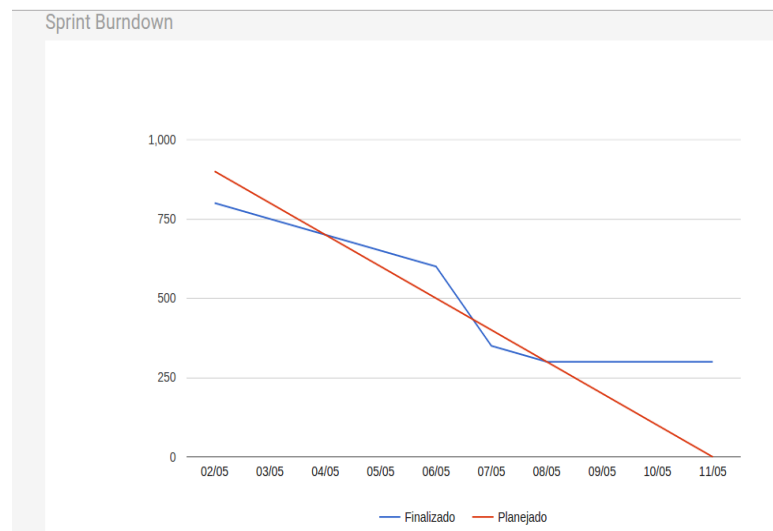
Possibilitando visualizar o ritmo do andamento da *sprint*, sabendo se está de acordo com o planejado, identificar possíveis problemas e traçar estratégias para finalizar a *sprint* dentro do prazo estipulado.

Figura 6 – Distribuições



Fonte: *Print screen* da aplicação ScruManager, 2017.

Figura 7 – *Sprint Burndown*



Fonte: *Print screen* da aplicação ScruManager, 2017.

5 CONSIDERAÇÕES FINAIS

É evidente que o desenvolvimento de um projeto de *software* de um tamanho razoável sem a utilização de qualquer metodologia, está fadado ao fracasso.

No caso analisado neste trabalho, os projetos falham não apenas por ausência de metodologia, mas também por pecar ao faltar com as documentações e padronizações relativas à disciplina de engenharia de *software* como diagramas, dicionários de dados, especificações UML e até mesmo casos de uso do sistema.

Com base nos conceitos de *Scrum* apresentados, no desenvolvimento e implementação da solução proposta, será atendida a necessidade da CNTLog de aumentar o controle, adaptabilidade e qualidade de seus projetos internos de desenvolvimento de *software*.

Como melhoria futura, há a implementação do controle visual de *taskboard* para facilitar a visualização por cada usuário, das tarefas pendentes e do responsável por quais tarefas.

REFERÊNCIAS

PRESSMAN, R. S. **Engenharia de *software***. São Paulo: Pearson Makron Books, 1995.

SBROCCO, J. H. T. C.; MACEDO, Paulo Cesar de. **Metodologias ágeis: engenharia de *software* sob medida**. 1. Ed. São Paulo: Érica, 2012.

SCHWABER, K., SUTHERLAND, J. *Guia do Scrum*. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 25 mar. 2017

SCRUM ALLIANCE. *The 2015 State of Scrum Report*. Disponível em: <<https://www.scrumalliance.org/why-scrum/state-of-scrum-report>>. Acesso em: 27 mar. 2017.

SUTHERLAND, J. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo**. São Paulo: LeYa, 2014.

THE STANDISH GROUP. *The Standish Group Report CHAOS*. Disponível em: <<https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>>. Acesso em: 27 mar. 2017.

“O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es).”