

Universidade Federal de Minas Gerais  
Departamento de Ciência da Computação  
Especialização em Informática  
Ênfase em Engenharia de Software

CLAUDSON ELONE EVANGELISTA DE SOUZA

**APLICAÇÃO DO MÉTODO ÁGIL SCRUM NO DESENVOLVIMENTO DE  
PRODUTOS DE SOFTWARE EM TELEMEDICINA NO CENTRO DE  
TELESSAÚDE DO HOSPITAL DAS CLÍNICAS DA UFMG**

Belo Horizonte  
2012

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciências da Computação  
Especialização em Informática: Ênfase: Engenharia de Software

**APLICAÇÃO DO MÉTODO ÁGIL SCRUM NO  
DESENVOLVIMENTO DE PRODUTOS DE SOFTWARE EM  
TELEMEDICINA NO CENTRO DE TELESSAÚDE DO HOSPITAL  
DAS CLÍNICAS DA UFMG**

por

Claudson Elone Evangelista de Souza

Monografia de Final de Curso

Prof. Clarindo Isaias Pereira da Silva e Pádua

Orientador

Belo Horizonte  
2012

**CLAUDSON ELONE EVANGELISTA DE SOUZA**

**APLICAÇÃO DO MÉTODO ÁGIL SCRUM NO DESENVOLVIMENTO DE  
PRODUTOS DE SOFTWARE EM TELEMEDICINA NO CENTRO DE  
TELESSAÚDE DO HOSPITAL DAS CLÍNICAS DA UFMG.**

Monografia apresentada ao Curso de Especialização em Informática do Departamento de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Especialista em Informática.

Área de concentração: Engenharia de Software

Orientador(a): Clarindo Isaias Pereira da Silva e Pádua

Belo Horizonte  
2012



## RESUMO

As metodologias ágeis focam na entrega do maior valor de negócio, no menor tempo possível. Um tipo de processo ágil que se encaixa neste modelo é o Scrum, que foi estudado e adotado pelo Centro de Telessaúde do Hospital das Clínicas com o objetivo de organizar, padronizar e melhorar os processos utilizados para desenvolvimento de software. O principal objetivo deste trabalho é descrever a implantação da metodologia ágil em um pequeno ambiente de produção de software de telemedicina, bem como o resultante desta implantação para o trabalho do setor de desenvolvimento do Centro de Telessaúde do Hospital das Clínicas. Pretende-se, com esta implantação, obter melhor acompanhamento e organização no desenvolvimento de software do Centro de Telessaúde. O projeto piloto para a aplicação da metodologia ágil foi o da Teleconsultoria 2.0. Este sistema contempla a troca de informações à distância entre profissionais de saúde, utilizando textos, imagens, eletrocardiogramas, entre outros recursos para esclarecimento de dúvidas e discussões de caso. Neste trabalho é descrito as atividades realizadas para contemplar o Scrum, bem como levantado as dificuldades relatadas pelos envolvidos e também as facilidades proporcionada à equipe pelo Scrum.

Palavras-chave: Scrum, metodologias ágeis, Telessaúde, desenvolvimento, software.

## **ABSTRACT**

The agile methods focus on delivering the highest business value in the shortest time possible. One type of agile process that fits this model is Scrum, which has been studied and adopted by the Centre for Telehealth, Hospital das Clinicas in order to organize, standardize and improve the processes used for software development. The main objective of this paper is to describe the deployment of agile methodology in small production environment of software for telemedicine, as well as the result of this for the work of the development sector of the Center for Telehealth, Hospital das Clinicas. It is intended, with this deployment, a better monitoring and organization in software development at the Center for Telehealth. The pilot project for the implementation of the Agile method was Teleconsultoria 2.0. This system includes distance exchange of information between health professionals, using text, images, electrocardiograms, and other resources promote discussions. This paper describes the activities performed to implement Scrum, as well as raises the difficulties reported by those involved, and also the benefits provided by the Scrum team method.

Keywords: Scrum, Agile, Telehealth, development, software.

## LISTA DE FIGURAS

FIG. 1	Quadro tarefas com todas as sprint's finalizadas .....	22
FIG. 2	Ferramenta utilizada pela equipe para acompanhar processo .....	23
FIG. 3	Gráfico Burndown da primeira Sprint .....	23
FIG. 4	Status estória .....	24
FIG. 5	Estrutura do sistema mostrada em árvore .....	25
FIG. 6	Estrutura do sistema mostrada em hierarquia .....	25
FIG. 7	Cartões com estórias apresentados pelo software Mingle .....	26

## LISTA DE SIGLAS

CTHC	Centro de Tessaúde do Hospital das Clínicas
HTTP	Protocolo de Transferência de Hipertexto
IDE	Ambiente Desenvolvimento Integrado
MVC	Modelo, Visão e Controle
UFMG	Universidade Federal de Minas Gerais



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>10</b>
<b>1.1</b>	<b>Motivação .....</b>	<b>10</b>
<b>1.2</b>	<b>Objetivos .....</b>	<b>12</b>
1.2.1	Geral.....	12
1.2.2	Específicos .....	12
<b>1.3</b>	<b>Metodologia .....</b>	<b>13</b>
<b>1.4</b>	<b>Justificativa.....</b>	<b>14</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>16</b>
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>20</b>
<b>3.1</b>	<b>Análise dos dados obtidos no Scrum .....</b>	<b>22</b>
3.1.1	Reunião Sprint.....	22
3.1.2	Quadro de Tarefas .....	23
<b>4</b>	<b>RESULTADOS.....</b>	<b>29</b>
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>32</b>
	<b>CRONOGRAMA.....</b>	<b>33</b>
	<b>REFERÊNCIAS.....</b>	<b>34</b>
	<b>ANEXO 1.....</b>	<b>35</b>

## **1 INTRODUÇÃO**

O trabalho desenvolvido baseia-se na necessidade de organizar o desenvolvimento de produtos de software do Centro de Telessaúde do Hospital das Clínicas (CTHC). A metodologia ágil foca na entrega do maior valor de negócio, no menor tempo possível. Um tipo de processo ágil que se encaixa neste modelo é o Scrum, que foi estudado e adotado pelo CTHC com o objetivo de organizar, padronizar e melhorar os processos utilizados para desenvolvimento de software.

Através da implantação do Scrum, os processos de desenvolvimento de agentes de software do CTHC puderam ser geridos, evitando-se esforços desnecessários, erros oriundos de requisitos implícitos, falhas de comunicação entre setores e a contínua mudança de regras de negócios foi resolvida com maior facilidade pelos desenvolvedores.

### **1.1 Motivação**

O CTHC é responsável pelo desenvolvimento de ações de Telessaúde do Hospital das Clínicas da Universidade Federal de Minas Gerais, aproximando os centros universitários e profissionais de saúde fisicamente distantes, difundindo o conhecimento por meio de tecnologias de informação e comunicação.

A aproximação realizada entre profissionais de saúde é feita através de agentes de software de desenvolvimento interno, em ação conjunta com softwares de terceiros. Os softwares proprietários foram desenvolvidos de acordo com o surgimento de necessidade de melhorias tecnológicas de comunicação. A evolução e/ou desenvolvimento destes programas são realizados sem o auxílio de um arcabouço que possa orientar e coordenar o seu desenvolvimento.

Os serviços prestados pelo CTHC tiveram um aumento muito significativo desde o início de suas atividades (2001), principalmente nos dois últimos anos (2010, 2011),

chegando a atingir 754 pontos (unidades de saúde) no Estado de Minas Gerais, com a previsão de ultrapassar a marca de 4000 pontos até o final de 2014. Com este aumento contínuo na demanda de serviços, logo veio a evidência de que os sistemas de software utilizados na atualidade pelo CTHC não iriam ser eficientes para suportar este aumento. Com isto, foi decidido a utilização de novos aplicativos, com maior robustez. Para a criação destes novos softwares, o cenário de desenvolvimento atual seria totalmente impróprio, podendo levar ao caos o setor, devido à falta de um ambiente controlado para desenvolvimento de software.

O CTHC não contempla processo de desenvolvimento de software, sendo que o processo de trabalho adotado, em sua grande maioria, limita-se a conversas formais (reunião) ou mensagens eletrônicas (e-mail). O desenvolvimento do trabalho proposto fica por conta do desenvolvedor sem nenhuma formalização. Este cenário atual provoca diversos erros e insatisfação entre os envolvidos, pois o serviço a ser executado nunca é delimitado e o trabalho entregue, na maioria das vezes, não contempla por completo o que era esperado.

A utilização de um método prescritivo (métodos tradicionais) para o desenvolvimento de software foi o primeiro meio a ser analisado pela equipe de desenvolvimento, mas, devido às características de instabilidade de requisitos que o próprio CTHC impõe, são constantes as mudanças no serviço prestado; logo, houve um consenso que esta metodologia poderia se transformar em um problema, e o caos continuaria sendo o cenário. Com a detecção de vários obstáculos na utilização de um método tradicional para desenvolvimento de software, o estudo da possibilidade de se utilizar um método ágil foi efetivado, chegando-se à conclusão que este tipo de processo seria mais adequado ao contexto.

Pressman (2005) define a engenharia de software ágil de forma bem alinhada com o que busca o CTHC:

Engenharia de software ágil combina uma filosofia e um conjunto de diretrizes de desenvolvimento. A filosofia encoraja a satisfação do cliente e a entrega incremental do software logo de início, equipes de projeto pequenas, altamente motivadas, métodos informais, produtos de trabalho de engenharia de software mínimos e simplicidade global do desenvolvimento. As diretrizes de desenvolvimento enfatizam a

entrega em contraposição à análise e ao projeto (apesar dessas atividades não serem desencorajadas) e a comunicação ativa e contínua entre desenvolvedores e clientes. (PRESSMAN, ROGER, 2005, p. 62).

Diante de vários fatores favoráveis ao uso de uma metodologia ágil, a implantação do Scrum no setor de desenvolvimento de software do CTHC foi uma decisão tomada com a expectativa de melhora no fluxo de trabalho do setor. Com esta implantação, a verificação e o estudo das dificuldades encontradas serão uma forma de contribuir em mudança futura, tanto ao setor quanto a outras empresas.

## **1.2 Objetivos**

### **1.2.1 Geral**

O principal objetivo deste trabalho é descrever a implantação da metodologia ágil em um pequeno ambiente de produção de software de telemedicina, bem como o resultante desta implantação para o trabalho do setor de desenvolvimento do CTHC. Pretende-se, com esta implantação, obter melhor acompanhamento e organização no desenvolvimento de software do CTHC.

### **1.2.2 Específicos**

Os objetivos específicos deste trabalho irão relatar práticas do Scrum no CTHC, contemplando os seguintes itens:

- Descrever práticas de desenvolvimento ágil de software através do Scrum.

- Aplicar o Scrum em um pequeno ambiente de desenvolvimento de software, verificando as dificuldades e benefícios de sua implantação, bem como a sua aceitação em um local com uma cultura de desenvolvimento já estabelecida.
- Conseguir, através do Scrum, gerir projetos de desenvolvimento de softwares em ambiente com mudança contínua.
- Apresentar resultados do processo de implantação do Scrum.
- Apresentar resultado do Scrum implantado, analisando se realmente os projetos atingiram um patamar gerenciável.
- Avaliar se houve melhoria de condições de trabalho para a equipe de desenvolvimento.

### **1.3 Metodologia**

No desenvolvimento deste trabalho será realizado um estudo sobre o método ágil Scrum, utilizando-se de pesquisas bibliográficas, bem como metodologias utilizadas para sua implantação. Juntamente com a implantação do Scrum no CTHC, será feito o relato das experiências vivenciadas nesta implantação, informando acertos e erros cometidos. Com o desenvolvimento do trabalho, os resultados obtidos com o Scrum serão avaliados em relação ao produto fabricado (software), gerenciamento de projetos e produtividade da equipe de desenvolvimento.

A escolha de uma ferramenta de desenvolvimento foi baseada em publicações encontradas e material explicativo sobre o seu funcionamento. Analisando o material encontrado, chegou-se ao framework CakePHP que se mostrou bem difundido no mercado, e ainda houve a ajuda de um analista da equipe que possui experiência profissional com a ferramenta. O CakePHP possui padrão de arquitetura MVC, que proporciona uma arquitetura extensível, facilitando a manutenção e o desenvolvimento.

Como banco de dados foi utilizado o Mysql, com auxílio da ferramenta SQLyog. Neste caso é aproveitado o “know how” da equipe, que já possui experiência com o banco de dados e a sua IDE.

O ambiente integrado de desenvolvimento (IDE), o Netbeans, é utilizado para codificação do sistema. Esta IDE já é utilizada em projetos desenvolvidos na linguagem Java, facilitando, assim, a sua utilização pelos colaboradores.

Para acompanhamento do processo, será utilizada a ferramenta Mingle, que é uma solução de gerenciamento ágil de projetos. A utilização do Mingle, ficou acordada devido a experiência que um dos colaboradores possui, com isto o treinamento dos demais em sua utilização ficou facilitada, além do relato positivo sobre a mesma. No Mingle serão registrados, para a equipe, as tarefas realizadas, gráficos burndown da Sprint, tempo necessário para finalização das estórias, status das tarefas, priorização das funcionalidades solicitadas pelo solicitante do Produto.

#### **1.4 Justificativa**

A adoção do Scrum visa, principalmente, alcançar melhor qualidade nos softwares produzidos no CTHC, já que é sabido que, para se obter um padrão de qualidade em softwares, o processo de desenvolvimento deverá ser sistemático e bem definido. Segundo Pereira (2007, p.01) *“Qualidade é um conjunto de atributos de software que devem ser satisfeitos de modo que o software atenda às necessidades dos usuários”*.

Os métodos ágeis são uma proposta de desenvolvimento de software que visam adaptar-se às mudanças decorrentes de fatores externos (mudanças de requisitos). Estes tipos de métodos, tidos como ágeis, diferem dos métodos tradicionais, já que os tradicionais são centrados em planejamento e documentação, enquanto os ágeis são adaptativos e visam não gastar tempo escrevendo documentação. O Scrum é um método ágil de desenvolvimento de software, considerado um framework que pode ser utilizado para empregar diversos processos e técnicas de desenvolvimento de software. O seu papel é evidenciar a sua eficácia nas práticas de produção de software, para que estas sejam melhoradas, podendo até mesmo ser utilizadas em produtos complexos. O controle de processos do Scrum é realizado de forma

empírica, empregando-se abordagens iterativa e incremental, controlando risco e otimizando a previsibilidade.

## 2 REFERENCIAL TEÓRICO

A busca por melhorias no processo de desenvolvimento de produtos de software é um ponto crucial em uma empresa do segmento no mundo atual. A melhoria no processo de desenvolvimento de produtos impactará diretamente na qualidade do produto final, que será apresentado e oferecido aos clientes. A engenharia de software apoia o seu desenvolvimento, sempre em busca de melhores resultados. *“A engenharia de software é um ramo da engenharia cujo foco é o desenvolvimento dentro de custos adequados de sistemas de software de alta qualidade”*. (SOMMERVILLE, 2007).

As empresas que desenvolvem software, em busca de um produto cada vez mais atraente para o cliente, e com custos de produção mais baixos, investem em qualidade de produção, já que este investimento pode ser garantia de um bom resultado. Nas empresas de software, o investimento na qualidade de seu desenvolvimento vem crescendo nos últimos anos. Sommerville (2007).

O processo de desenvolvimento de software é uma das principais atividades para que a qualidade do produto seja atingida. Segundo Sommerville (2007), um processo de software é um conjunto de atividades que levam à produção de um produto de software. Na atualidade, dois tipos de padrões são utilizados para esta finalidade, métodos tradicionais ou métodos ágeis. Os métodos tradicionais se demonstram mais sistemáticos, aparentemente mais estruturados do que os ágeis. Pressman fala sobre processo de forma bastante consistente para os processos tradicionais.

“Os métodos de engenharia de software fornecem a técnica de ‘como fazer’ para construir softwares. Eles abrangem um amplo conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção. Os métodos de engenharia de software repousam num conjunto de princípios que regem cada área da tecnologia e incluem atividades de modelagem e outras descritivas.” (PRESSMAN, ROGER, 2005, P. 18)

A metodologia ágil visa o desenvolvimento rápido de software, de forma incremental, onde o produto é desenvolvido por funcionalidades e as mudanças de requisitos são



facilmente tratadas. Sommerville (2007) diz que processos de desenvolvimento rápido de software são projetados para criar software útil rapidamente. Os processos ágeis são centrados nas pessoas. Possuir um “time” de desenvolvimento com talento, com capacidade de agir em prol do sucesso no seu trabalho é essencial para que todo o processo flua de forma correta e eficaz. Este talento que o desenvolvedor ágil deve possuir, segundo Pressman, está distribuído nas seguintes características:

- Competência: Em um contexto de desenvolvimento ágil (bem como em um contexto de engenharia de software convencional), “competência” inclui talento inato, habilidades específicas relacionadas a software e conhecimento global do processo que a equipe decidiu aplicar. Habilidade e conhecimento do processo podem e devem ser ensinados a todas as pessoas que servem como membros de equipes ágeis.

- Foco comum: Embora os membros da equipe ágil possam realizar diferentes tarefas e trazer diferentes habilidades ao projeto, todos deveriam estar focados em uma meta e entregar um incremento de software em funcionamento ao cliente dentro do prazo prometido. Para atingir essa meta, a equipe também vai concentrar-se em adaptações contínuas (pequenas e grandes) que farão o processo satisfazer à necessidades da equipe.

- Colaboração: Engenharia de software (independente do processo) diz respeito a avaliar, analisar e usar as informações que são comunicadas à equipe de software, criar informações que ajudarão o cliente e outros a entender o trabalho da equipe, e construir informações (software de computador e banco de dados relevantes) que forneçam valor de negócio para o cliente. Para realizar essas tarefas, os membros da equipe precisam colaborar uns com os outros, com o cliente e com os gerentes do negócio.

- Capacidade de tomada de decisão: Qualquer boa equipe de software (inclusive equipes ágeis) deve ter a liberdade de controlar o seu próprio destino. Isso implica que seja dada autonomia à equipe e autoridade para tomada de decisão sobre tópicos técnicos e de projeto.

- Habilidade de resolver problemas vagos: Os gerentes de software devem reconhecer que uma equipe ágil terá de lidar continuamente com ambigüidades e será continuamente confrontada por modificações. Em alguns casos a equipe precisa aceitar o fato de que o problema que está sendo resolvido hoje pode não ser o problema que precisará ser resolvido amanhã. No entanto, as lições aprendidas de qualquer atividade de solução de problema (inclusive aquelas que resolvem o problema errado) podem ser benéficas para a equipe mais adiante no projeto.

- Respeito e confiança mútua: A equipe ágil deve tornar-se uma equipe “consolidada”. Uma equipe consolidada exhibe a confiança e o respeito necessários para torná-la tão fortemente aglutinada que o todo é maior que a soma das partes.

- Auto-organização: No contexto de desenvolvimento ágil, a auto organização implica três coisas: (1) a equipe ágil organiza-se para o

trabalho a ser feito; (2) a equipe organiza o processo para melhor acomodar seu ambiente local; (3) a equipe organiza o cronograma de trabalho para conseguir melhor entrega do incremento de software. A auto-organização tem certo número de benefícios técnicos, porém, mais importante que isso, ela serve para aperfeiçoar a colaboração e aumentar o moral da equipe. Em essência, a equipe serve como sua própria gerência. (PRESSMAN, ROGER, 2005, p. 62)

Scrum é um método ágil, considerado um framework, que pode ser utilizado para se empregar diversos processos e técnicas de desenvolvimento de software. Schwaber (2011), compara o Scrum com um jogo de rúgbi. O Scrum ou formação ordenada é uma situação frequente no rúgbi, geralmente é usado após uma jogada irregular ou em alguma penalização. Os 8 Avançados das duas equipes formam uns contra os outros. O Scrum-half (Médio-Formação) da equipe que não cometeu a infração insere a bola no meio do "túnel" formado pelas duas primeiras linhas de cada equipe com a finalidade de que os jogadores da sua equipe consigam ganhar (talonar) a bola. Schwaber (2002). O seu papel é promover eficácia nas práticas de produção de software, para que estas sejam melhoradas, podendo até mesmo ser utilizado em produtos complexos. Segundo Schwaber, Scrum não é um processo ou uma técnica para construir produtos; em vez disso, é um framework dentro do qual você pode empregar vários processos ou técnicas. O Scrum deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las. Schwaber (2011). O controle de processos do Scrum é realizado de forma empírica, empregando-se abordagens iterativa e incremental, controlando risco e otimizando a previsibilidade. Segundo Schwaber (2011), o Scrum é fundamentado na teoria de controle de processos empíricos, emprega uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar riscos. Schwaber (2011) ainda complementa que, para que este controle de processos empíricos funcione, três fundamentos são essenciais, sendo eles:

1 – Transparência – os aspectos do processo que impactam nos resultados devem estar transparentes para as pessoas que os gerenciam, sendo que, além de sua transparência, a sua visibilidade deve ser conhecida.

2 – Inspeção - diversos aspectos do processo devem ser inspecionados com uma determinada frequência, afim de se encontrar varrições inaceitáveis durante o

processo; para isto a inspeção é utilizada com uma determinada frequência. Isto se faz para que seja inspecionado o andamento do planejamento Sprint em direção a meta estabelecida.

3 - Adaptação – Caso, na inspeção, seja detectado que aspectos do processo estão fora dos limites aceitáveis, o processo deverá ser ajustado, ou até mesmo o material que esteja sendo trabalhado sofrerá ajustes.

O *Sprint* é uma iteração entre a equipe, que consiste em uma reunião de planejamento, trabalho de desenvolvimento, revisão da Sprint e a Retrospectiva da *Sprint*. Segundo Schwaber, (2011), o coração do Scrum é a Sprint, que é uma iteração de um mês ou menos de duração consistente com o esforço de desenvolvimento. Todas as Sprints utilizam o mesmo modelo de Scrum e todas as Sprints têm como resultado um incremento do produto final que é potencialmente entregável. Cada Sprint começa imediatamente após a anterior.

A adaptação e a inspeção no Scrum são realizadas basicamente em três momentos distintos do processo: reunião diária Scrum, reunião de revisão do Sprint, e a de retrospectiva do Sprint. Segundo Schwaber (2011), a reunião diária Scrum é uma reunião utilizada para inspecionar o progresso em direção à Meta do Sprint e para realizar adaptações que otimizem o valor do próximo dia de trabalho. Schwaber, (2011), ainda fala que:

“As reuniões de Revisão da Sprint e de Planejamento da Sprint são utilizadas para inspecionar o progresso em direção à Meta da Release e para fazer as adaptações que otimizem o valor da próxima Sprint. A Retrospectiva da Sprint é utilizada para revisar a Sprint passada e definir que adaptações tornarão a próxima Sprint mais produtiva, recompensadora e gratificante.” ( SCHWABER, 2011, P. 14)

O Scrum é formado por times de Scrum, com seus papéis vinculados, artefatos, regras e time-boxes. Os times de Scrum são auto-organizáveis, multidisciplinares e trabalham com iterações. A divisão do time é realizada por papéis, cada papel é definido com os seguintes nomes e atribuições, Schwaber (2011):

- Scrum Master: responsável por garantir que o processo seja compreendido e seguido.
- Product Owner: o product owner é responsável por maximizar o valor do trabalho que o time Scrum faz.
- Time boxes: definido como o que executa o trabalho propriamente dito. O Time consiste em desenvolvedores com todas as habilidades necessárias para transformar os requisitos do Product Owner em um pedaço potencialmente entregável do produto ao final do Sprint.

No Scrum, quatro artefatos são tidos como principais. O produto *Backlog* é uma lista priorizada de requisitos a serem desenvolvidos. Deve ser único durante toda a vida do projeto, e deve constar tudo que deverá ser feito. Os objetivos de mais prioridade são transferidos para um *Sprint Backlog*, que é a lista de tarefas que define o trabalho da equipe; isto ocorre anteriormente a cada Sprint. O *Burndown* de Release mede o trabalho restante dentro do tempo do projeto, refletindo a intuição da equipe, servindo-lhe de parâmetro. As Regras realizam o elo entre os time-boxes, os papéis e os artefatos do Scrum.

### **3 DESENVOLVIMENTO**

O projeto piloto para a aplicação da metodologia ágil foi o da Teleconsultoria 2.0. Este sistema contempla a troca de informações à distância entre profissionais de saúde, utilizando textos, imagens, eletrocardiogramas, entre outros recursos para esclarecimento de dúvidas e discussões de caso.

Em um sistema de teleconsultoria, um caso clínico real ou uma dúvida genérica é enviado pela internet para análise de profissionais especializados que analisam e respondem à solicitação, podendo também encaminhá-la para uma terceira opinião. Isso é feito através de uma interface web que restringe o acesso apenas aos profissionais envolvidos no processo, em um tempo relativamente curto, já que geralmente as dúvidas vêm de profissionais que estão em pontos mais afastados e têm que atender a várias especialidades sem muitos recursos.

Este produto a ser desenvolvido (Teleconsultoria 2.0) já é de conhecimento da equipe de desenvolvimento, pois todos estão envolvidos na manutenção e/ou extensão da primeira versão do produto (Teleconsultoria 1.0), que deverá ser descontinuado devido a dificuldades que este programa vem oferecendo para incremento de novas funcionalidades, e também o risco de não atender a carga de trabalho prevista para os próximos anos.

Para iniciar o desenvolvimento do sistema, reuniões de planejamento da Sprint foram realizadas com toda a equipe para definir o que seria entregue e como seria realizado o trabalho necessário para entregar as funcionalidades. A coordenação do setor, que fez papel de Dono do Produto (cliente), apresentou os itens do Backlog para a equipe, que avaliou o que poderia ser feito em cada Sprint.

Foram realizadas reuniões diárias do Scrum para manter a equipe informada sobre o andamento do projeto e verificar se havia algum impedimento na realização das tarefas. Nessas reuniões, os membros da equipe relatavam o que conseguiram fazer desde o último encontro, o que ficaria pronto no próximo encontro e quais obstáculos estavam impedindo o andamento do projeto.

No final de cada Sprint, foram feitas reuniões de revisão para inspecionar o trabalho realizado e apresentar o que estava pronto, quais os problemas encontrados e como foram resolvidos.

Na Reunião de Retrospectiva, o time avaliou os pontos positivos e negativos que ocorreram durante a Sprint, identificando melhorias no trabalho para as próximas tarefas.

O acompanhamento do trabalho foi feito pela equipe através das reuniões diárias e também pela ferramenta Mingle, com a qual foi possível visualizar todo o processo do Scrum de forma intuitiva e rápida.

### **3.1 Análise dos dados obtidos no Scrum**

#### **3.1.1 Reunião Sprint**

**Sprint1:** Na primeira Sprint ficou definido que seriam feitas as tarefas diretamente ligadas à realização da teleconsultoria, como criar um formulário dinamicamente para cada especialidade médica escolhida pelo usuário, fazer perguntas aos profissionais de saúde e alguns cadastros necessários para preencher listas de opções. A maior parte da modelagem do banco de dados foi importada do sistema antigo; mesmo assim, nessa Sprint, foi necessário criar novas tabelas e fazer algumas alterações.

**Sprint2:** Alguns novos conceitos foram discutidos com os usuários como o status das teleconsultorias; sendo assim, nesta Sprint, ficaram definidas algumas mudanças nos requisitos. Foi desenvolvida a visualização de acordo com o perfil, regras de negócio, tarefas administrativas e algumas correções de bugs.

**Sprint3:** Um dos pontos discutidos nessa Sprint foi a quantidade de campos que seriam disponibilizados para o cadastro dos pacientes, ficando definido que seria criado um cadastro dinâmico que poderia ser alterado de acordo com as necessidades do usuário. Foi desenvolvido um controle para permitir associar moderadores a determinadas especialidades, opções de encaminhamento da pergunta a outro especialista, entre outras regras de negócio discutidas com o Product Owner.

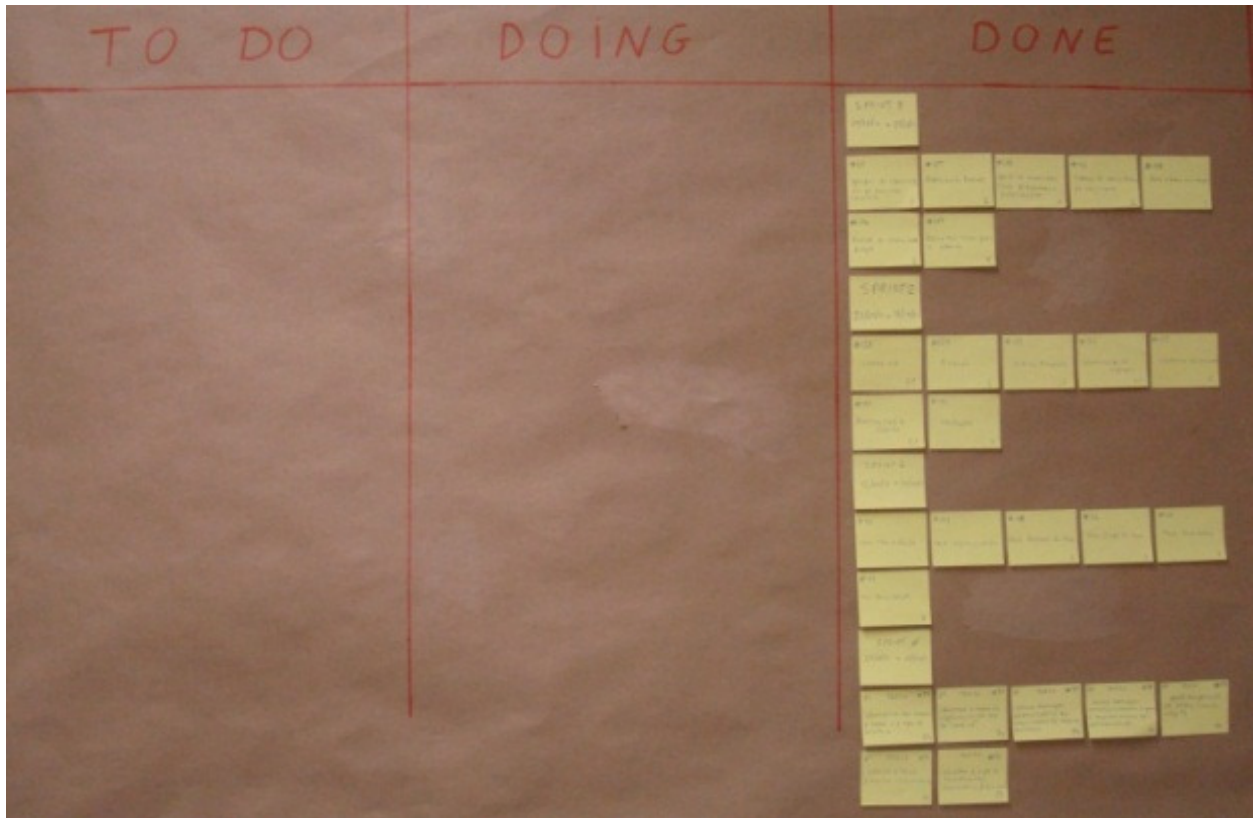
**Sprint4:** A última Sprint foi utilizada para finalizar algumas tarefas da Sprint anterior, realização de testes e correções. Ao término, foi apresentado para a diretoria o resultado final do projeto em que apareceram algumas modificações que serão implantadas em uma próxima versão do sistema.

Reuniões foram feitas com usuários do sistema e gerência que continuamente foi acrescentando mudanças à medida que o projeto ia sendo desenvolvido. Como forma de gerar um registro e acompanhar melhor o desenvolvimento, as informações foram lançadas na ferramenta Mingle, que gerou automaticamente gráficos para acompanhamento do projeto.

### 3.1.2 Quadro de Tarefas

Depois das reuniões de planejamento da Sprint, as tarefas foram anotadas em post-it e fixadas no quadro, no espaço denominado “To do” (fazer). No decorrer do trabalho, os desenvolvedores pegavam as tarefas e moviam para o espaço “Doing” (fazendo), colocando o nome de quem as estava fazendo no cartão. Após o término da tarefa, o desenvolvedor movia novamente a tarefa para o espaço “Done” (feito), concluindo o processo de seu desenvolvimento.

Figura 1: Quadro de tarefas com todas as Sprint's finalizadas.



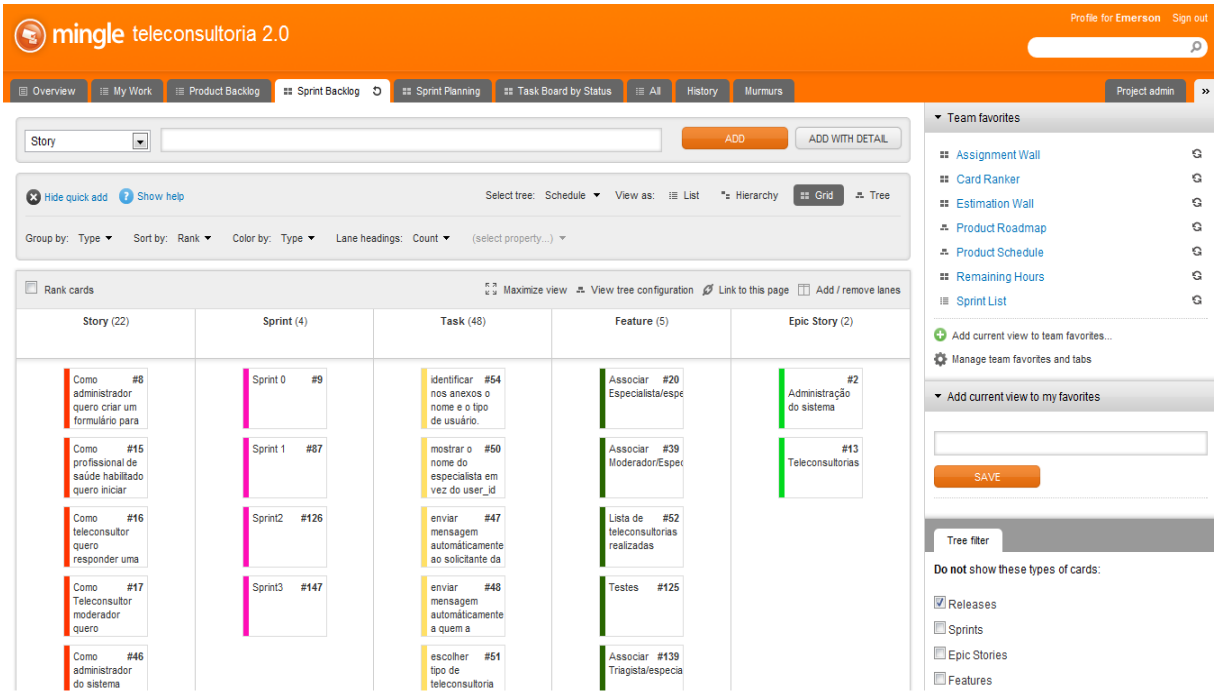
Fonte: CTHC.

Através do quadro, a equipe conseguiu visualizar o andamento da realização das tarefas, quais destas ainda não tinham sido feitas e quais estavam finalizadas. Como os próprios desenvolvedores retiravam os post-it e os movimentavam no quadro, as tarefas foram melhor divididas, de acordo com o conhecimento técnico de cada um.

Outra forma de acompanhamento do projeto também foi testada através do software Mingle. Além das tarefas, o software gerencia todos os processos do Scrum, gerando gráfico de *burndown* da Sprint automaticamente, à medida que o status das tarefas é modificado no sistema.

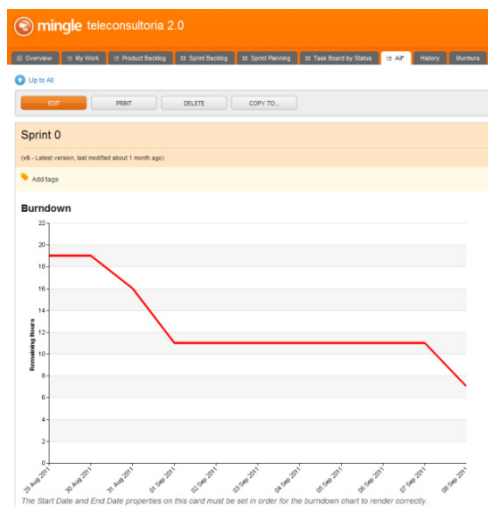


Figura 2: Ferramenta utilizada pela equipe para acompanhar o processo.



Fonte: CTHC.

Figura 3: Gráfico Burndown da primeira Sprint



Fonte: CTHC.

Os status das estórias disponíveis no sistema estão divididos em: pronto para análise, em análise, pronto para desenvolvimento, em desenvolvimento, pronto para teste, em teste, pronto para entrega e aceito.

Neste ponto também é preenchido o proprietário da tarefa, ponto por estória, se existe alguma dependência em relação à outra estória.

Figura 4: Status da estória

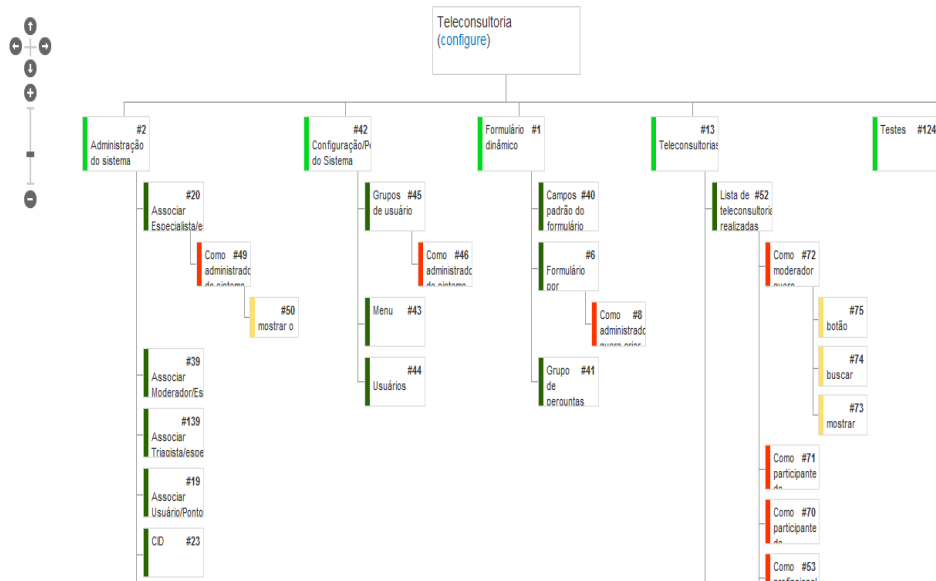
The screenshot shows a Jira story card with the following details:

- Type:** Story
- Status:** Accepted
- Owner:** Emerson
- Story Points:** 3
- Depend on:** (not set)
- Sprint Added to:** (not set) Scope
- Sprint Completed:** (not set)
- from Schedule tree:** (This card belongs to this tree.)
  - Release:#10 release 1
  - Schedule - Sprint:#9 Sprint 0
  - Schedule - Epic Story:(not set) 1
  - Schedule - Feature:(not set)
  - Estimated Task: (not set) Hours
- from Teleconsultoria tree:** (This card belongs to this tree.)
  - Epic Story:#13 Teleco...
  - Feature:#14 Realiz...

Fonte: CTHC.

Se preenchido corretamente, o sistema organiza todo o trabalho em forma de listagem, árvore ou hierarquicamente, o que facilita a visualização das tarefas de acordo com suas dependências.

Figura 5: Estrutura do sistema mostrada em árvore.



Fonte: CTHC.

Figura 6: Estrutura do sistema mostrada em hierarquia.

13	Teleconsultorias
10	▼ release 1
147	▼ Sprint3
146	revisão de idioma nas telas do plugin
145	busca por especialidade, paciente, etc para o admin
144	colocar data e hora na lista de teleconsultorias
142	criar termo de concordância do solicitante
138	botão finalizar teleconsultoria com opção do solicitante aceitar ou não
137	tornar um especialista padrão em cada especialidade
133	criar grupos de especialidades para facilitar a inserção
102	Como desenvolvedor do sistema gostaria de realizar alguns testes no sistema
139	Associar Triagista/especialidade
126	▼ Sprint2
141	visualizar teleconsultorias que ainda não tem resposta de acordo com a categoria permitida (triagista)
140	Como administrador do sistema quero associar um triagista a uma especialidade
136	retirar o campo tipo de dúvida
135	escolher quais os campos estarão disponíveis para cadastrar o paciente
134	resumir a visualização do enfoque
132	Como administrador do sistema quero associar um moderador a uma especialidade
131	colocar a categoria e sub-categoria do cid na resposta

Fonte: CTHC.

Figura 7: Cartões com histórias apresentados pelo software Mingle.

Story (22)				Task (49)			
Como administrador quero criar um formulário para #8	Como profissional de saúde habilitado quero iniciar uma #15	Como teleconsultor quero responder uma #16	Como Teleconsultor moderador quero encaminhar uma #17	identificar #54 nos anexos o nome e o tipo de usuário.	mostrar o #50 nome do especialista em vez do user_id	enviar #47 mensagem automaticamente ao solicitante da	enviar #48 mensagem automaticamente a quem a
Como #46 administrador do sistema quero escolher quais	Como #49 administrador do sistema quero associar um	Como #53 profissional de saúde habilitado quero visualizar	Como #56 profissional de saúde quero poder recuperar	escolher #51 tipo de teleconsultoria (assistencial /	opção de #57 recuperação de senha	botão #62 finalizar teleconsultoria.	corrigir o #76 botão finalizar teleconsultoria.
Como #58 profissional de saúde quero visualizar as	Como #78 profissional habilitado depois de escolher o	como #88 visitante do site gostaria de realizar um pré-	Como #94 administrador do sistema gostaria de habilitar,	criar #89 formulário de pré-cadastro	tela de login: #90 testes e botão voltar no recuperar senha.	revisão de #91 idioma nas telas do plug in.	revisão e #92 testes : usuário/ponto remoto
Como #98 profissional habilitado gostaria de	Como #102 desenvolvedor do sistema gostaria de	como #103 administrador gostaria de habilitar e	Como #108 administrador gostaria de habilitar/desabilitar	revisão e #93 testes : especialista/espec	botão #95 habilitar/desabilitar status	botão e tela #96 inserir tipo de dúvida	botão e tela #97 editar tipo de dúvida
como #112 administrador gostaria de habilitar e	como #113 administrador gostaria de habilitar e	como #114 administrador gostaria de habilitar e	Como #130 especialista quero responder uma	botão e tela #99 inserir paciente	botão e #100 tela visualizar paciente	botão e #101 tela editar paciente	botão #104 habilitar/desabilitar status
Como #132 administrador do sistema quero associar um	Como #140 administrador do sistema quero associar um			botão e #105 tela editar	botão e #106 tela inserir	botão e #109 tela editar	botão e #110 tela inserir
<b>Feature (5)</b>							
Associar #20 Especialista/espec	Associar #39 Moderador/Especi	Lista de #52 teleconsultorias realizadas	Testes #125	botão #111 habilitar/desabilitar status	botão e #115 tela editar	botão e #116 tela inserir	botão #117 habilitar/desabilitar status
Associar #139 Triagista/especiali				botão e #118 tela editar	botão e #119 tela inserir	botão #120 habilitar/desabilitar status	botão e #121 tela editar
				botão e #122 tela inserir	botão #123 habilitar/desabilitar status	retirar o #128 campo cid	dividir a #129 tela em 3: novas/em aberto/finalizadas
				colocar a #131 categoria e sub-categoria do cid na resposta	criar #133 grupos de especialidades para facilitar a	resumir a #134 visualização do enfoque	escolher #135 quais os campos estarão disponíveis para
<b>Release (1)</b>	<b>Sprint (4)</b>			retirar o #136 campo tipo de dúvida	tornar um #137 especialista padrão em cada especialidade	botão #138 finalizar teleconsultoria com opção do	visualizar #141 teleconsultorias que ainda não tem resposta de
release 1 #10	Sprint 0 #9	Sprint 1 #87		criar termo #142 de concordância do solicitante	escolher #143 por plantão (várias especialidades)	colocar #144 data e hora na lista de teleconsultorias	busca por #145 especialidade, paciente, etc para o admin
	Sprint2 #126	Sprint3 #147		revisão de #146 idioma nas telas do plugin			

Fonte: CTHC.

## 4 RESULTADOS

O Scrum aplicado teve também como objetivo descrever o trabalho para implantar o framework, bem como analisar a dificuldade que a equipe teria em aceitar esta nova cultura. A implantação do Scrum foi, em seu todo, tranquila, ela se demonstrou muito simples e de fácil entendimento. O estudo da metodologia foi realizado por toda a equipe em conjunto. Todos os envolvidos procuraram ler sobre o assunto e trocaram informações, além da ajuda de um colaborador que já havia trabalhado com o Scrum.

As mudanças de cultura impostas pelo framework Scrum, foram sentidas por toda a equipe, que estava acostumada a realizar o trabalho quase que individualmente e colocando em prática aquilo que se achava mais viável para o momento. A divisão das tarefas que deveriam ser cumpridas e o seu registro no post-it fixado ao quadro de tarefas, e também no Mingle, foi algo bem inovador para a maioria, e nas primeiras atividades gerou conflito e algumas reclamações.

Depois do primeiro impacto negativo com o quadro de tarefas e do Mingle, os desenvolvedores começaram a se acostumar com os registros, e foram além, perceberam como os registros se tornaram aliados no desenvolvimento do seu trabalho. A concentração de esforços passou a ser maior, ou seja, o trabalho ficou mais focado, pois novas requisições de serviço não poderiam ser aceitas pelo setor de desenvolvimento do CTHC; de qualquer forma, as pessoas acostumadas com isto passaram a visualizar as tarefas que estavam sendo executadas, entendiam e respeitavam mais a negativa de se fazer um trabalho naquele momento, e tratavam estas requisições com o Scrum Master (gerente). Anteriormente ao Scrum, requisições de novos trabalhos e/ou mudanças no que já estava sendo executado eram constantes. Em um trabalho com um tempo de duração estimado em duas semanas, a média de pedido de alteração e/ou inclusão era de quatro (4) para o mesmo período. Nesta mesma quinzena, novas demandas surgiam numa média de um (1) novo serviço por semana para cada desenvolvedor. Com o Scrum, as requisições de novas demandas de trabalho continuaram com a mesma intensidade,

porém passaram a ser feita seguindo o Scrum em reuniões semanais e definidas por prioridades, gerando um filtro para novas tarefas, tornando assim o fluxo mais rápido e profissional entre os envolvidos.

Os desenvolvedores puderam, de uma forma mais clara e visual, mostrar que para novos trabalhos e/ou alterações em solicitações anteriores, teriam que seguir um procedimento e que este procedimento tornaria o fluxo de serviço mais rápido.

As reuniões Sprint foram outro ganho para o setor. Com estas reuniões, o contato com o solicitante do serviço passou a ser rotineiro, e com maior formalidade, além de ficar mais claro e melhor definido que tarefa iria ser realizada, em que prazo, e como estava o seu desenvolvimento. Com isto, a cobrança pelo trabalho realizado em um prazo ilusório foi quase que totalmente resolvido, além do total acompanhamento por parte do solicitante, causando menos atrito entre as partes. No cenário anterior, os atrasos nos prazos estabelecidos para cada trabalho era uma rotina, gerando cobranças e mal estar entre os envolvidos.

As constantes mudanças ou indefinições sobre o trabalho a ser realizado foram também contornadas pelas reuniões de Sprint, já que o próprio solicitante é que decidia quais funcionalidades eram de maior relevância e quais regras seriam aplicadas a estas. Além disso, a priorização de retrabalho de alguma atividade era decidida em conjunto, provocando assim visualização e conscientização por parte do solicitante. Mudanças não são algo simples de se fazer; geralmente isto acarreta um prazo maior de entrega do produto final. A aceitação por parte do Dono do Produto destas mudanças não foi fácil, com a argumentação que estava se burocratizando, ou que era perda de tempo. Essas foram as primeiras reclamações levantadas. Mas, no decorrer do projeto, verificou-se que os próprios solicitantes gostaram das mudanças, pois os produtos começaram a ser entregues mais rápido, e eles próprios começaram a visualizar e entender melhor o que estava sendo construído e/ou modificado, gerando assim maior tranquilidade e satisfação.

Na organização interna do setor, ganhou-se uma nova opção de acompanhamento e registro das tarefas, o Mingle. Com isso, conseguiu-se gerir os projetos que estavam sendo executados, e visualizar tarefas já realizadas e futuras. Com a ferramenta

Mingle, registrou-se o tempo de desenvolvimento das atividades; assim, conseguiu-se negociar melhor com o cliente os prazos de entrega dos produtos, além de permitir estimar esforços necessários em serviços que apresentam características parecidas.

No aspecto geral, os desenvolvedores relataram que o Scrum trouxe benefícios ao seu trabalho. As melhorias na comunicação e no relacionamento entre os envolvidos foram as mais citadas. A utilização do Mingle para registrar o status das atividades foi o item mais relatado como dificuldade de adaptação ao Scrum. A falta de uma pessoa, com qualificação em Scrum para sua implantação, foi o ponto negativo mais citado entre os colaboradores. Entre os desenvolvedores foi unanimidade a melhora de relacionamento entre o setor de desenvolvimento do CTHC com os demais setores. A sistemática do Scrum foi primordial para a melhora de relacionamento entre os envolvidos. Para se chegar aos dados relatados acima, foi aplicado um questionário para a equipe de desenvolvedores, tal questionário, com as respostas, se encontra no anexo 1.

## 5 CONCLUSÃO

O uso do Scrum no desenvolvimento de projetos é um tema bastante discutido na atualidade, sendo adotado por diversas empresas do setor de desenvolvimento de software.

Apesar de ter muitos pontos positivos que aceleram o desenvolvimento de um produto, a aplicação do Scrum no trabalho aqui registrado ainda foi limitada pela falta de conhecimento técnico da equipe, que precisa estar bem nivelada para que não ocorram falhas no processo.

O autogerenciamento é fundamental para o sucesso do time que precisa estar totalmente comprometido com o projeto, já que todos tomam decisões importantes que definem o andamento das tarefas.

A experiência também mostrou pontos positivos em diversos aspectos, como nas reuniões diárias, que proporcionaram um maior entrosamento da equipe. O acompanhamento das tarefas através da ferramenta Mingle também foi bastante produtivo, dando uma visão em tempo real do andamento do projeto.



**CRONOGRAMA**

<b>Etapas</b>	<b>Atividades</b>	<b>Produto</b>	<b>Início</b>	<b>Fim</b>
Inicial	Definição do tema / Problema / Objetivo / Orientador	Definição da Monografia	11/04/2011	22/05/2011
Análise de informações	Estudo e organização do conhecimento obtido	Projeto da Monografia	23/05/2011	10/08/2011
Elaboração da monografia	Elaboração da Monografia	Versão inicial da monografia	11/08/2011	30/08/2011
Versão e Correção monografia	Primeira Correção	Primeira Versão	21/02/2012	02/05/2012
Versão e Correção monografia	Segunda Correção	Segunda Versão	17/05/2012	20/05/2012
Versão e Correção monografia	Entrega Versão Final	Versão Final	23/05/2012	23/05/2012
Apresentação	Apresentação Monografia	Monografia	30/05/2012	30/05/2012

## REFERÊNCIAS

AgileManifesto (2001) “Manifesto for Agile Software Development”,  
<http://agilemanifesto.org/>. Última visita: novembro 2011.

CakePHP, <http://cakephp.org/>. última visita: novembro 2011.

Maria Beatriz Moreira Alkmim, Mônica Pena de Abreu, Renato Minelli Figueira, Lemuel Rodrigues Cunha, Elizete da Silva Neme, Antônio Luiz Pinho Ribeiro, Teleassistência para municípios remotos do Estado de Minas Gerais: ampliando a atuação geográfica dos Hospitais Universitários

Mingle, <http://www.thoughtworks-studios.com/mingle-agile-project-management>  
Última visita: novembro 2011

PRESSMAN (2006), Roger. **Engenharia de Software**. 6 Edição. McGraw-Hill Editora, 2006.

SOMMERVILLE (2003), Ian. **Engenharia de Software**. 8 ed. São Paulo: Addison-Wesley, 2007.

Schwaber (2011), Ken Schwaber, Jeff Sutherland. **Scrum**, Disponível em: <  
<http://www.scrum.org/storage/scrumguides/Scrum%20Guide%20-%20PTBR.pdf#view=fit>> Acesso em 15/05/2012.

Telessaúde, <http://www.telessaude.hc.ufmg.br/>. Última visita: novembro 2011

## ANEXO 1

1) Quais benefícios foram proporcionados ao seu trabalho com a utilização do Scrum?

Colaborador 1 → Comunicação, e dedicação dos mais envolvidos nos projetos.

Colaborador 2 → Melhor organização, prazos melhores, maior compreensão no que esta sendo desenvolvido.

Colaborador 3 → Maior foco no que estava sendo desenvolvido, além de melhor relacionamento com os envolvidos no trabalho.

Colaborador 4 → Melhor distribuição das tarefas, comunicação mais objetiva, melhor relacionamento entre as pessoas.

Colaborador 5 → Organização das tarefas, trabalho mais focado, menor retrabalho, melhor relacionamento entre colaboradores, maior produtividade.

2) O que foi mais difícil na adaptação do seu trabalho com a introdução da metodologia Scrum?

Colaborador 1 → Prazos de entrega. Pois quando há um requisito novo, devemos ter o conhecimento da tecnologia a ele aplicada, e isso deve ser levado em consideração.

Colaborador 2 → Adaptar a sistemática do Scrum, apesar de participar na definição das tarefas, utilizar um sistema para registrar o trabalho é algo que sempre acaba ficando para trás, mas a adaptação esta acontecendo.

Colaborador 3 → A minha adaptação foi tranquila, o Scrum deixa os desenvolvedores bem à vontade, a sistemática a ser seguida não sacrifica a rotina,

por já ter trabalhado com Scrum, vejo como fácil a adaptação, trazendo na verdade benefícios.

Colaborador 4 → Registrar no Mingle as atividades desenvolvidas.

Colaborador 5 → No início achei que as reuniões seriam desnecessárias, achava que perderíamos tempo, mas logo percebi a objetividade das mesmas é logo me adaptei. Registrar o trabalho desenvolvido não é difícil, mas por diversas vezes acabo esquecendo, esta é a maior dificuldade encontrada.

3) Quais foram os aspectos positivos durante a implantação do Scrum? E os negativos?

Colaborador 1 → Iteração entre os desenvolvedores, onde todos se empenharam na construção de uma única demanda.

Colaborador 2 → A possibilidade de conhecer uma nova rotina de trabalho, fazendo um comparativo entre uma forma pessoal de se trabalhar com o uso de uma metodologia, que apesar de deixar o desenvolvedor bem à vontade, o Scrum impõe uma boa organização de fluxo de trabalho.

Colaborador 3 → Maior organização no trabalho, possibilidade de saber o que cada desenvolvedor está fazendo, juntamente com toda demanda do setor, poder mensurar com maior facilidade a produtividade pessoal e também da equipe. Negativamente ficou a falta de conhecimento da equipe sobre implantação do Scrum.

Colaborador 4 → De positivo foi o envolvimento da equipe, de negativo foi a falta de uma pessoa como referência para mostrar o que fazer, como fazer na implantação do Scrum, se tivesse esta referência teria sido mais tranquila a implantação.

Colaborador 5 → Conhecer um novo método de desenvolvimento de software foi o mais positivo. De negativo não ficou nada, pois esta implantação só agregou pontos positivos para a equipe é o setor.

4) Houve mudança de relacionamento dos demais setores do CTHC com o setor de Desenvolvimento? Quais mudanças ocorreram?

Colaborador 1 → Sim, sem dúvidas. Com a aplicação do Scrum em nosso ambiente, se tornou mais comunicativo e dinâmico. Os colaboradores conseguem acompanhar o desenvolvimento dos projetos, seus prazos e requisitos implementados.

Colaborador 2 → Sim. A demanda de serviço deixou de ser tratada diretamente com os desenvolvedores, o Scrum estabeleceu uma rotina onde as pessoas de outros setores passaram a fazer parte com isto o atrito entre desenvolvedores é o restante dos colaboradores deixou de existir, é o trabalho demandado passou a ser melhor alocado.

Colaborador 3 → Houve grande mudança de relacionamento dos demais setores com o setor de desenvolvimento. A continua alteração no que estava sendo desenvolvido passou a ser discutida semanalmente, em diversas vezes a alteração que seria proposta deixou de ser solicitada, com isto os prazos passaram a ser cumpridos, evitando atrito entre os colaboradores.

Colaborador 4 → Sim, as novas demandas passaram a ser tratadas em reuniões, não esta sendo mais aceito novos serviço ou alterações diretamente com o desenvolvedor, isto fez com que discussões sobre possibilidades de desenvolvimento e alterações juntamente com prazo de entrega seja mais elaborada é ampla, evitando assim atrito desnecessário.

Colaborador 5 → Sim. Além de deixar de existir cobranças desnecessárias de ambas as partes, o relacionamento mais formal imposto pelo Scrum, fez com que erros de entendimento de requisitos diminuísse, prazo de entrega fosse melhor

definido, deixando de existir um certo desconforto entre os desenvolvedores e os demais setores do centro de Telessaúde.