



**Faculdade de Informática e
Administração Paulista**

Curso de Sistemas de Informação – 2º SI-T

Engenharia de Software
Modelo de Desenvolvimento Ágil SCRUM

Hugo Cisneiros	RM 60900
Moyses Santana Jacob	RM 63484
Stelvio Mazza	RM 63117
Tiago Pereira	RM 63115

São Paulo

2009

Sumário

Introdução: Modelos de desenvolvimento de software.....	3
SCRUM: O que é?.....	4
1. Papéis do SCRUM (Roles).....	4
1.1. Proprietário do Produto (Product Owner).....	5
1.2. ScrumMaster.....	5
1.3. A Equipe de Desenvolvimento.....	6
2. Cerimônias SCRUM (Cerimonies).....	6
2.1. Reunião de Planejamento do Sprint.....	7
2.2. Reuniões Diárias SCRUM.....	7
2.3. Reunião de Revisão do Sprint.....	8
3. Artefatos SCRUM (Artifacts).....	8
3.1. Product Backlog.....	8
3.2. Sprint Backlog.....	8
3.3. Burndown Chart.....	9
O Processo SCRUM.....	10
Exemplo de Software: Sistema Web de streaming de vídeo.....	11
1. Papéis.....	11
2. Escopo do Projeto.....	11
3. Product Backlog.....	11
4. Reunião de Planejamento do Sprint.....	12
5. Início do Sprint.....	13
5.1. Reuniões diárias SCRUM.....	14
5.2. Burndown Chart.....	14
6. Revisão final do Sprint.....	14
Bibliografia.....	16

INTRODUÇÃO: MODELOS DE DESENVOLVIMENTO DE SOFTWARE

No decorrer de vários anos o desenvolvimento de software cresceu de forma muito acelerada, principalmente pelo fato de que as ferramentas da informática se tornaram presentes em grande parte das áreas de trabalho das pessoas. Quanto mais a informática penetrava no mercado, mais tipos de softwares tinham de ser feitos.

Esta crescente demanda fez surgir uma variedade de estratégias de desenvolvimento de software. Estas estratégias têm como objetivo organizar o projeto de software e o time de desenvolvimento, aumentando a eficácia e diminuindo os prazos e custos destes mesmos projetos de software. As metodologias de desenvolvimento trouxeram uma maior qualidade para o produto final: algo muito importante principalmente quando os softwares são mais complexos e precisam de muita organização para não se tornarem um "monstro" incontrolável.

Alguns dos principais problemas encontrados no processo de desenvolvimento de software:

- Caos pela mudança constante de requisitos;
- Estimativas de tempo, custo e qualidade do produto totalmente irreais;
- Os desenvolvedores não sabem exatamente como está indo o andamento do projeto e acabam *mentindo* ou se enganando sobre seus próprios resultados.

Na onda das metodologias de desenvolvimento de software, algumas técnicas foram criadas com o foco de terminar os projetos de software rapidamente e de forma eficaz. Este tipo de técnica foi categorizada com o nome de *Desenvolvimento Ágil*, onde a velocidade e o dinamismo são dois pontos importantes.

Nos modelos de *Desenvolvimento Ágil*, o principal objetivo é obter um produto rapidamente e com qualidade. Para isto, as metodologias que participam desta categoria de modelos se caracterizam por um gerenciamento de projeto em que um líder de time esteja frequentemente organizando, inspecionando, apoiando e garantindo que o time esteja bem, ao mesmo tempo que os resultados do projeto de software vão atendendo as necessidades do cliente.

O **SCRUM** é um modelo de *Desenvolvimento Ágil*. Ele lida totalmente com as pessoas e como elas vão desenvolver o projeto, sem se preocupar com que solução tecnológico o projeto irá utilizar: o próprio time de desenvolvimento já atuando no projeto faz isto. Talvez por este motivo, o SCRUM é um dos processos mais eficazes de gerenciamento de pessoas no desenvolvimento ágil e é bastante combinado com outras metodologias que tratam mais especificadamente da parte técnica.

Este documento traz uma introdução ao que é realmente o SCRUM e como ele funciona, para a aplicação da metodologia em projetos de software.

SCRUM: O QUE É?

O **SCRUM** é um modelo de desenvolvimento ágil de software que fornece métodos para se definir o *planejamento*, os *principais papéis de pessoas* e a *forma de trabalho do time*. O nome SCRUM é derivado de uma jogada de rúgbi, onde todo o mesmo time avança como apenas uma unidade, trabalhando com os mesmos jogadores e em conjunto, passando sempre a bola pra um e para outro.

A idéia do SCRUM é justamente definir papéis bem específicos para as pessoas envolvidas no projeto e como cada pessoa vai *jogar*, ou seja, o que cada uma vai ter que fazer para o *time* seguir em frente no *jogo*: que no nosso caso é o próprio desenvolvimento do software.

Em geral e de forma reduzida, esta metodologia funciona da seguinte forma:

1. O produto é definido: quais são os seus requisitos? O que realmente o cliente quer? O responsável por esta tarefa é o que chamamos de Proprietário do Produto (*Product Owner*, em inglês).
2. O Proprietário do Produto define quais são as funcionalidades do programa que mais importam, criando assim o que chamamos de *Product Backlog*.
3. Com as prioridades definidas, uma pessoa é definida para ser o *ScrumMaster*, uma espécie de coordenador do projeto. O *ScrumMaster*, junto com o Proprietário do Produto e o Time de desenvolvimento definem o que chamamos de *Sprints*.
4. Cada *Sprint* possui uma parte de todo o *Product Backlog*, e devem ser trabalhados de acordo com as prioridades definidas no *Product Backlog*. Os *Sprints* devem ser preparados de uma forma de que durem de 2 a 4 semanas, e que no final de cada período tenham um produto apresentável para o cliente.
5. Os *Sprints* vão sendo feitos até o *Product Backlog* acabar e o Proprietário do Produto definir que o projeto está pronto. Mas isso não quer dizer que novas funcionalidades não podem ser incluídas: basta ir adicionando no *Product Backlog* e realizando outros *Sprints*.

Durante os passos reais de desenvolvimento, os *Sprints*, a principal ferramenta de *medição* de desempenho é o *Burndown Chart*, que é uma das características mais especiais do SCRUM e que o torna um grande diferencial, no sentido positivo.

Falando mais detalhadamente, o SCRUM tem três *partes* principais em seu modelo: **Papéis (Roles)**, **Cerimônias (Cerimonies)** e **Artefatos (Artifacts)**. Cada um destes três pilares contém outros sub-itens. Todas estas três partes principais são utilizadas no que chamamos de ciclo de desenvolvimento, ou seja, o *Sprint*. Cada *Sprint* possui suas fases e utiliza destes *papéis*, *cerimônias* e *artefatos* para alcançar seu objetivo final.

1. Papéis do SCRUM (Roles)

Como a metodologia define como o time deve trabalhar, o primeiro passo para o desenvolvimento por SCRUM é definir **quem vai fazer o quê**. Por isso chegamos a três *entidades*

principais: o Proprietário do Produto (*Product Owner*), o *ScrumMaster* e o Time.

1.1. Proprietário do Produto (*Product Owner*)

O Proprietário do Produto representa os interesses do cliente. Ele tem que ser a *interface* entre o cliente e o time de desenvolvedores, ou seja, estar sempre em contato com o cliente e saber exatamente o que o projeto **tem que ser**.

Ele tem as seguintes responsabilidades:

- Definir as características e conteúdo do produto;
- Decidir sobre a data de término;
- Ser responsável pela rentabilidade do produto;
- Prorizar as funções de acordo com o valor de mercado e com o cliente;
- Ajustar recursos e priorizar tarefas a cada 30 dias, como necessário;
- Aceitar ou rejeitar o resultado do trabalho.

O trabalho mais árduo do Proprietário do Produto é definir o *Product Backlog*, um dos dois artefatos do SCRUM, que veremos posteriormente. Essa definição se dá durante o *Planejamento*, que também veremos adiante.

1.2. ScrumMaster

O ScrumMaster é o líder da equipe de desenvolvimento e durante o trabalho, fica mais em contato com o Proprietário do Produto. Ele gerencia e repassa o trabalho que foi decidido durante o planejamento pelo Proprietário do Produto. Ele deve:

- Assegurar que a equipe de desenvolvimento funcione plenamente e seja produtiva;
- Ajudar na cooperação entre todas as funções e papéis do time;
- Remover barreiras;
- Proteger a equipe de interferências externas;
- Assegurar-se de que a metodologia está sendo seguida, incluindo chamadas para reuniões diárias (*Daily Scrum Meetings*), revisões de atividade (*Sprint Reviews*) e reuniões de planejamento das atividades (*Sprint Planning*).

Devido a todas estas responsabilidades, o ScrumMaster é o que podemos chamar de Coordenador do Projeto. Ele facilita a comunicação entre as pessoas e faz o projeto andar de verdade.

Além de comandar as reuniões diárias, o ScrumMaster têm três principais responsabilidades:

1. Precisa saber quais atividade foram concluídas, quais foram iniciadas, quaisquer novas

tarefas que foram descobertas e qualquer estimativa que possa ter mudado. Isto permite a ele atualizar sempre o *Burndown Chart*, que permite mostrar quanto trabalho falta para um Sprint acabar, dia por dia. Ele também tem que sempre olhar cuidadosamente para o número de tarefas em aberto. Estes trabalhos em aberto devem ser minimizados o máximo possível para garantir um trabalho sempre limpo e eficiente.

2. Deve avaliar as dependências superficiais e bloqueios que causam prejuízos ao andamento do Projeto. Estes itens devem receber prioridades e serem acompanhados. Um plano de solução deve ser implementado de acordo com a ordem de prioridade destes impedimentos. Alguns podem ser resolvidos pelo time, outros podem ser resolvidos através de vários times, e outros podem precisar de envolvimento da gerência, pois podem ser problemas relacionados à empresa que bloqueiam qualquer membro do time a fazer qualquer coisa. Como exemplo deste tipo de impedimento externo, temos as questões judiciais.

3. O ScrumMaster deve perceber e resolver problemas pessoais ou conflitos entre os integrantes do time de desenvolvimento SCRUM. Este tipo de problema deve ser clarificado pelo ScrumMaster e resolvido por diálogo com o time, ou então o ScrumMaster terá que ter ajuda da gerência ou do departamento de Recursos Humanos. Alguns estudos apontam que 50% dos problemas de desenvolvimento acontecem por razões pessoais. O ScrumMaster precisa estar sempre atento ao time para fazer dele totalmente funcional e produtivo.

1.3. A Equipe de Desenvolvimento

A equipe de desenvolvimento é quem vai colocar a mão na massa para que o software comece a ter cara e funcionamento. Pode haver uma ou mais equipes de desenvolvimentos, dependendo da complexidade do software.

Algumas características das equipes de desenvolvimento:

- São pequenas e multidisciplinares, com em média 7 participantes;
- Definem metas de cada *Sprint*, junto ao ScrumMaster, e especificam seus resultados de trabalho;
- Têm o direito de fazer tudo dentro dos limites das diretrizes do projeto para atingir a meta de cada *Sprint*.
- Organiza os trabalhos para atingir os objetivos dos *Sprints*.
- Trabalha para atingir todos os resultados definidos pelo Proprietário do Produto.

2. Cerimônias SCRUM (Cerimonies)

As cerimônias SCRUM são eventos que acontecem dentro de um ciclo de desenvolvimento utilizando esta metodologia. Existem três tipos de cerimônias SCRUM: a reunião de Planejamento do Sprint, as reuniões diárias SCRUM e a reunião de revisão do Sprint. Estes três tipos de evento caracterizam bem o ciclo de vida de cada *Sprint*: início, meio e fim.

2.1. Reunião de Planejamento do Sprint

Como dito anteriormente em resumo, o primeiro passo de um projeto SCRUM é quando o Proprietário do Produto desenvolve um plano para o projeto de software. Neste plano, o Proprietário do Produto trabalha bem próximo ao cliente para definir todas as funcionalidades que o cliente quer no seu produto. A partir desta lista, ele também tem que definir as prioridades de cada funcionalidade de acordo com várias variáveis: valor de mercado, importância de base, importância para o cliente, entre outras. Esta lista final é o que chamamos de *Product Backlog* e é a base para a reunião de planejamento do *Sprint*.

Nesta reunião, o ScrumMaster trabalha junto com o Proprietário do Produto e a Equipe de Desenvolvimento para definir qual a carga de tempo que cada funcionalidade do *Product Backlog* terá. Isto ajuda o Proprietário do Produto a definir prazos reais para o projeto e o habilita a poder verificar como está o andamento do projeto durante todo o período de desenvolvimento. Esta carga de tempo e esforço é definida de acordo com o tamanho do(s) time(s), horas disponíveis e o nível de produtividade do time.

Quando as prioridades e prazos das funcionalidades do software são definidas por completo, o Proprietário do Produto sai de cena e o ScrumMaster começa a trabalhar juntamente com a Equipe de Desenvolvimento para a *quebra* destas tarefas grandes em pequenas tarefas, divididas por todos os integrantes da equipe de desenvolvimento de acordo com suas especialidades. Esta reunião de planejamento geralmente dura até 4 horas e é ela quem define o *Sprint Backlog*, um dos artefatos SCRUM.

Uma vez definido o *Sprint Backlog*, com a listagem de todas as tarefas, seus responsáveis e seus prazos, o processo de desenvolvimento começa.

2.2. Reuniões Diárias SCRUM

Uma das principais características deste modelo de desenvolvimento ágil são as reuniões diárias SCRUM, onde o ScrumMaster se reúne com a equipe de desenvolvimento para saber como anda o projeto. Esta reunião acontece todo dia durante o ciclo de desenvolvimento (*Sprint*) e tem uma duração curta de aproximadamente 15 minutos.

Durante a reunião, cada um dos membros responde as seguintes três perguntas:

1. O que fiz ontem?
2. O que fiz hoje?
3. Quais impedimentos e dificuldades apareceram no caminho?

O ScrumMaster tem um papel muito importante nestas reuniões: ele terá que identificar todos os problemas ou novas tarefas que surgirem e planejar como estas aparições serão resolvidas. Além do mais, ele terá assim uma visão completa do projeto e poderá gerenciar todas as sub-tarefas de cada *Sprint*, preenchendo assim o *Burndown Chart*.

2.3. Reunião de Revisão do Sprint

No final do período do Sprint, a reunião de revisão do Sprint é feita. Nesta reunião, a equipe de desenvolvimento, junto ao ScrumMaster, se reúne com o Proprietário do Produto (e convidados, caso ele achar adequado, como por exemplo o cliente ou acionistas do projeto). Esta reunião dura aproximadamente 4 horas.

Na primeira parte da reunião, o resultado do *Sprint* é apresentado para o Proprietário do Produto, ou seja, tudo que foi desenvolvido durante o ciclo de desenvolvimento. As funcionalidades são revisadas e o valor do produto é definido. Depois de revisar todo este resultado, o Proprietário do Produto define quais itens do *Product Backlog* foram completados no *Sprint*, discutindo então com os membros da equipe e o cliente quais serão as novas prioridades. Este é o primeiro passo para criar um novo *Sprint*, caso seja necessário, pois dessa primeira parte da reunião, um novo *Product Backlog* é gerado.

Na segunda parte da reunião, o ScrumMaster se reúne com a equipe de desenvolvimento e revisa os altos e baixos do ciclo. O time conversa para saber o que foi bom e como se pode melhorar ainda mais o ambiente, as ferramentas e o convívio entre o time e suas funções. Esta parte é basicamente um aprimoramento interno.

Depois que esta reunião é finalizada, um novo ciclo *Sprint* pode começar até que todo o produto seja implementado/finalizado e o *Product Backlog* esteja limpo de pendências.

3. Artefatos SCRUM (Artifacts)

Os artefatos SCRUM são as ferramentas básicas para se trabalhar com este modelo de desenvolvimento. Estes artefatos servem como guias e indicadores durante o processo. São divididos em três: *Product Backlog*, *Sprint Backlog* e *Burndown Chart*.

3.1. Product Backlog

Como vimos anteriormente, no início do projeto, o Proprietário do Produto prepara uma lista de funcionalidades desenvolvida em conjunto com o cliente, que é organizada por prioridade de entrega. Essa lista chama-se *Product Backlog*. A equipe SCRUM contribui para o *Product Backlog* estimando o custo do desenvolvimento das funcionalidades.

O *Product Backlog* deve incluir todas as funcionalidades visíveis ao cliente, mas também os requisitos técnicos para poder desenvolver o produto, e em torno de dez dias de desenvolvimento esses itens deverão estar prontamente definidos para o seu desenvolvimento começar. As tarefas que tem mais prioridade e complexidade são quebradas em sub-tarefas para poderem ser estimadas e testadas.

Pense no *Product Backlog* como uma *WishList*: uma lista de itens que queremos ter.

3.2. Sprint Backlog

Assim que a equipe de Scrum escolher e definir a lista de requisitos e a prioridade de seus itens do *Product Backlog*, cada um desses itens agora será dividido em partes menores para o *Sprint Backlog*. Ou seja, uma lista especificando os passos necessários para implementar uma funcionalidade

do sistema.

Logo após o *Sprint Backlog* estar concluído, o total de horas trabalhadas é comparado com o total previsto anteriormente no *Product Backlog*. Caso haja uma diferença significativa, a equipe SCRUM deve negociar novamente com o cliente o número correto de horas, para que o *Sprint* seja realizado com maior probabilidade de sucesso.

3.3. Burndown Chart

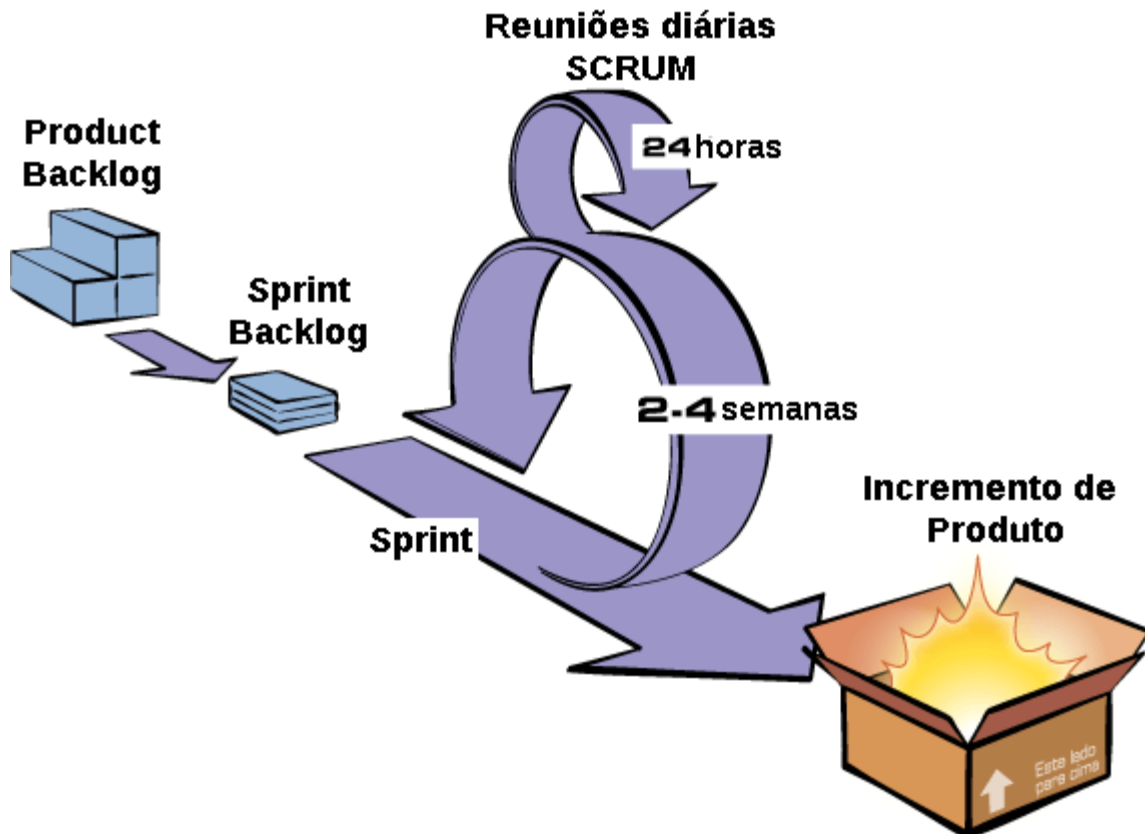
O *Burndown Chart* é um gráfico que mostra a quantidade de trabalho cumulativo restante de um *Sprint*, dia por dia. Neste gráfico, a altura indica a quantidade de tarefas do *Sprint Backlog* não completadas, e o comprimento são os dias. Com isto, podemos visualizar facilmente se um trabalho está tendo progresso, completando as tarefas, enquanto vemos que as colunas do gráfico vão caindo em sua altura. Se um ciclo de desenvolvimento, ou *Sprint*, tem a duração de 30 dias, haverão 30 colunas juntas. As colunas têm que diminuir ao longo do comprimento e antes ou até o 30o. dia não poderá haver colunas: indicando que todos as tarefas foram completadas e o *Sprint* foi um sucesso.

Durante as reuniões diárias do *Sprint*, o ScrumMaster acompanha os membros da equipe para ver o que foi terminado. Assim, dia a dia, ele ajusta o *Burndown Chart* e a qualquer momento todo o time pode ter uma idéia de como o processo está andando. O mesmo gráfico pode ser mostrado para o Proprietário do Produto ver que está tudo indo bem.

Por essas razões, o *Burndown Chart* é um dos principais recursos de medição do processo de desenvolvimento e um diferencial para a metodologia SCRUM.

O PROCESSO SCRUM

Como visto anteriormente, o processo SCRUM se dá por três etapas: O início, marcado pela reunião de planejamento, o ciclo de desenvolvimento (chamado *Sprint*) e a conclusão, marcada pela reunião de revisão do *Sprint*. Neste tópico, vamos ver um exemplo do processo SCRUM.



EXEMPLO DE SOFTWARE: SISTEMA WEB DE STREAMING DE VÍDEO

Para exemplificar o processo, utilizaremos um exemplo: o desenvolvimento de um sistema web de streaming de vídeo para a Internet.

1. Papéis

- Proprietário do Produto: Marcelo
- ScrumMaster: Hugo
- Equipe de Desenvolvimento: Moyses, Stelvio, Tiago

2. Escopo do Projeto

Definidos os papéis da equipe, o Proprietário do Produto Marcelo, depois de algumas várias visitas ao cliente e pesquisas de requisitos, define o *escopo* do projeto:

"O software será um sistema Web de streaming de vídeo, que permitirá usuários na Internet mandem seus vídeos, que serão armazenados no sistema e poderão ser gerenciados por seus donos e vistos pelo resto do mundo através das visitas ao site.

O sistema terá como funcionalidades principais a conversão dos vídeos mandados para um codec leve que permite qualidade e rapidez na visualização pela Internet; cadastro de usuários; interface de gerenciamento de vídeos; comentários para os vídeos e usuários; notas para vídeos; sistema de busca; reconhecimento de sons dos vídeos; proteção contra SPAM; sistema de legendagem de vídeos feitos por usuários; canais (grupos) de vídeos e usuários."

3. Product Backlog

Com as funcionalidades levantadas, o Proprietário do Produto Marcelo monta o *Product Backlog*, com as prioridades definidas de acordo com o valor de mercado/importância para o cliente. Em nosso exemplo, usaremos a notação de quanto maior o número de prioridade, menor prioridade ele tem (1 - prioridade máxima, 2 - prioridade menor, ..., 99 - prioridade quase inexistente):

Funcionalidade	Prioridade
Modelagem de dados	1
Cadastro e gerenciamento de usuários	2
Conversão de vídeo para visualização na Internet (Codec)	3
Cadastro e gerenciamento de vídeos pelos usuários	4
Layout (Aparência) da página	5

Comentários para os vídeos e usuários	6
Notas para vídeos (<i>ranking</i>)	7
Proteção contra SPAM	8
Canais (grupos) de vídeos e usuários	9
Sistema de legendagem de vídeos	10
Reconhecimento de sons dos vídeos	11

4. Reunião de Planejamento do Sprint

Com o *Product Backlog* definido, a reunião de planejamento é feita, o Proprietário do Produto Marcelo apresenta o projeto aos demais membros da equipe SCRUM e toda a equipe define a quantidade de horas que cada tarefa deverá ocupar. Os aspectos técnicos são levados em consideração e todo o planejamento é feito deste modo.

O resultado é um *Product Backlog* que agora tem suas estimativas de custo/hora:

Funcionalidade	Prioridade	Custo-horas
Modelagem de dados	1	32
Cadastro e gerenciamento de usuários	2	26
Conversão de vídeo para visualização na Internet (Codec)	3	80
Cadastro e gerenciamento de vídeos pelos usuários	4	48
Layout (Aparência) da página	5	28
Comentários para os vídeos e usuários	6	20
Notas para vídeos (<i>ranking</i>)	7	16
Proteção contra SPAM	8	20
Canais (grupos) de vídeos e usuários	9	26
Sistema de legendagem de vídeos	10	64
Reconhecimento de sons dos vídeos	11	92

Com o novo *Product Backlog*, define-se qual será a meta do primeiro *Sprint*:

- Modelagem de dados
- Cadastro e gerenciamento de usuários
- Conversão de vídeo para visualização na Internet (Codec)
- Cadastro e gerenciamento de vídeos pelos usuários

Sendo assim, o ScrumMaster Hugo, junto à equipe de desenvolvimento, define o *Sprint Backlog*, quebrando as tarefas grandes em pequenas tarefas:

Funcionalidade	Prioridade	Custo-horas
Modelagem de dados	1	32
Definição de dados	1.1	8
Organização de tabelas	1.2	12
Relacionamento	1.3	8
Implementação em SGBD	1.4	4
Cadastro e gerenciamento de usuários	2	26
Formulários	2.1	6
Interação com cadastro na base de dados	2.2	6
Visualização de perfil	2.3	6
Mudança de dados	2.4	4
Relacionamento entre usuários	2.5	4
Conversão de vídeo para visualização na Internet (Codec)	3	80
Definição e Implementação de Codec	3.1	56
Integração de Codec com sistema	3.2	24
Cadastro e gerenciamento de vídeos pelos usuários	4	48
Upload de vídeos	4.1	16
Remoção de vídeos	4.2	14
Perfis de vídeos	4.3	18

5. *Início do Sprint*

Com as metas preparadas e as tarefas bem definidas, chega a hora de começar o ciclo de desenvolvimento, o *Sprint*. O objetivo do primeiro *Sprint* será apresentar uma interface básica onde os usuários poderão se cadastrar, mandar e visualizar vídeos em uma interface crua. Consideramos o esqueleto do sistema.

No nosso exemplo, temos o total de **186 horas** de estimativa para acabar o *Sprint*. É importante lembrar que esta quantidade de horas deve ser ajustável para não ser menos que 3 dias e não mais que 30 dias de ciclo de desenvolvimento. Durante o ciclo de desenvolvimento, Moyses, Stelvio e Tiago irão trabalhar nas tarefas, de acordo com o seguinte sub-ciclo:

1. Desenvolver o produto: implementar, testar e documentar;
2. Empacotar: deixar o produto pronto para ser apresentado e integrado;
3. Revisar: o trabalho para se certificar do que foi feito;

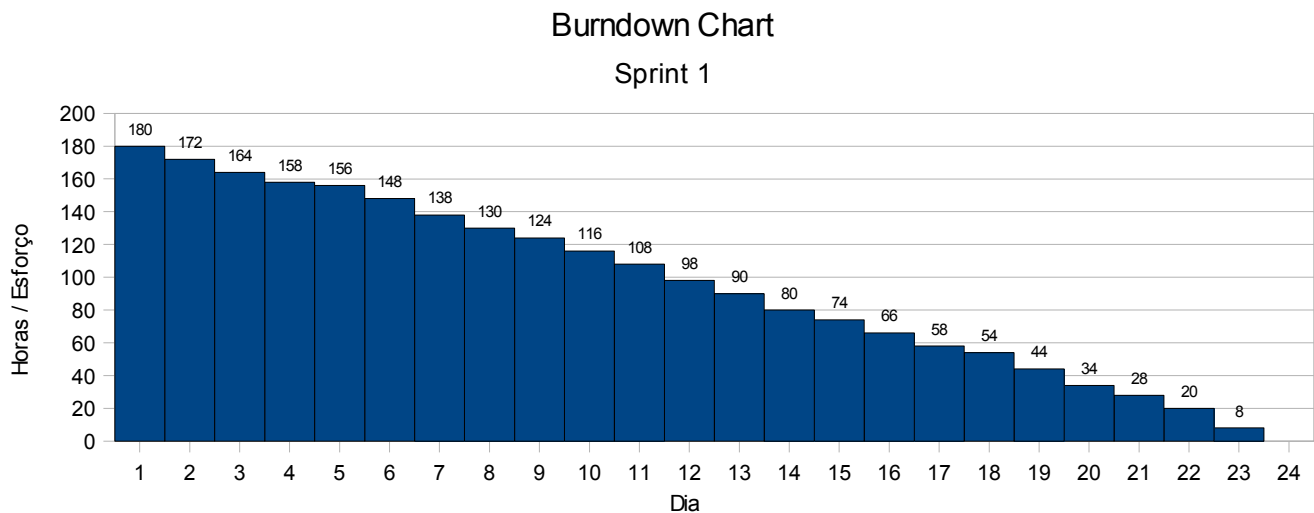
4. Ajustar: qualquer mudança nos requisitos ou planos.

5.1. Reuniões diárias SCRUM

O ScrumMaster Hugo irá acompanhar o desenvolvimento através de reuniões diárias para se certificar que os desenvolvedores Moyses, Stelvio e Tiago estejam completando suas tarefas, estejam bem de saúde, bem comprometidos com o projeto e se esforçando.

5.2. Burndown Chart

Com as reuniões diárias, o ScrumMaster Hugo poderá alimentar o *Burndown Chart*, como o exemplo a seguir:



Com o Burndown Chart, podemos ver claramente o andamento do projeto ao longo do seu ciclo de desenvolvimento (*Sprint*). Também, no meio do projeto, podemos calcular facilmente a velocidade com que o projeto está andando e assim estimar uma data para que o *Sprint* seja concluído.

Este dado estimativo pode ser comparado com o prazo que o Proprietário do Produto Marcelo nos deu para que possamos saber se o projeto vai acabar ou não no prazo.

Por exemplo, calculamos que a velocidade do projeto está como 8 horas por dia em média. Isso significa que o projeto irá terminar no dia 24, como o *Burndown Chart* nos mostrou. Se o prazo fosse, por exemplo, para o dia 19, logo nos 5 primeiros dias já poderíamos ter calculado essa velocidade e ver que estava pouco, tomando providencias para que essa velocidade se tornasse, por exemplo, 10 horas por dia em média, assim no dia 19 o projeto estaria pronto.

Este é um dos mais importantes trabalhos que o ScrumMaster Hugo terá que fazer e o *Burndown Chart* é o indicar perfeito para ele gerenciar o tempo de projeto e sua equipe de desenvolvimento.

6. Revisão final do Sprint

Ao final do ciclo de desenvolvimento (*Sprint*), toda a equipe se reúne e vê quais são os

resultados. O Proprietário do Produto Marcelo identifica todo o progresso e revisa o programa. Ele, junto com o cliente, concorda que os itens especificados para o Sprint foram completos e esta primeira versão do sistema web é satisfatória. Então eliminamos os seguintes itens:

- Modelagem de dados;
- Cadastro e gerenciamento de usuários;
- Conversão de vídeo para visualização na Internet (Codec);
- Cadastro e gerenciamento de vídeos pelos usuários.

Depois passamos a definir quais as próximas prioridades e então o que vai ser feito no próximo *Sprint*:

- Layout (Aparência) da página
- Comentários para os vídeos e usuários
- Notas para vídeos (*ranking*)
- Proteção contra SPAM

E o processo começa novamente, agora no que chamamos *Sprint 2*. Define-se os prazos e prioridades, e monta-se o plano de desenvolvimento para o próximo ciclo de desenvolvimento SCRUM.

BIBLIOGRAFIA

- <http://www.scrumalliance.org/>
- http://www.agilealliance.org/article/articles_by_category/17
- <http://www.codeproject.com/KB/architecture/scrum.aspx>
- <http://www.youtube.com/watch?v=Q5k7a9YEoUI>