

RETROSPECTIVA DO SPRINT 1

# SISGHE - Sistema de Grade Horaria Estudantil

## **SCRUM MASTER**

*Carlos Filipe Lima Bezerra*

*Guilherme de Lima Bernardes*

*Marcos Ronaldo Pereira Júnior*

*Pedro Henrique Potiguara Carvalho*

Universidade de Brasília - UNB  
Faculdade Gama – FGA  
Grupo de Engenharia de Software

Brasília, DF - 2013

# Índice

1.Planejamento do Sprint .....	3
1.1.Objetivos do Sprint .....	3
1.2.Objetivos do Sprint Bakclog .....	4
2.SCRUM MASTER - Conhecimentos relacionados a área de gerenciamento de projeto .....	5
3.EQUIPE - Conhecimentos relacionados às demais áreas envolvidas no desenvolvimento de um software, considerando a visão da engenharia de software .....	5
4.Retrospectiva do Sprint .....	6
Referências Bibliográficas .....	7

# 1 Planejamento do Sprint

## 1.1. Objetivos do Sprint

Esta Sprint consistiu na implementação das seguintes funcionalidades:

- Gerar PDF;
- Manter Grade Horária
- Manter Aluno(Validação dos dados)
- Visualizar Grade Horária

A geração do PDF havia sido criado na fase passada do projeto, porém não estava feita de forma correta e completa. Eram jogadas informações no arquivo sem nenhum tratamento. As modelagem também não havia sido criada da forma correta. Então, nessa sprint, foi determinada a criação das classes de acordo com o modelo MVC (DAO e Controler) referentes a geração do PDF e também a maneira de como o PDF deveria ser criado, podendo conter as informações do aluno, as informações das disciplinas cursadas e disciplinas à cursar ou ambos.

A funcionalidade de Manter a grade horária foi dividida em 4 histórias referentes a Salvar, Visualizar, Excluir e alterar a grade horária. Com isso, as atividades consistem na criação das classes referentes a arquitetura MVC, criação da interface gráfica e as funcionalidades de alterar, excluir e visualizar. Porém, a grade não irá ser montada ainda. Essa funcionalidade ficará para a Sprint 2, por ser uma funcionalidade muito complexa. O manter grade consiste então em criar a persistência, sem a lógica implementada.

A funcionalidade de Manter aluno já havia sido implementada na fase passada do projeto. Porém, os dados de entrada não haviam sido validados. Nessa Sprint, é desejado implementar as validações dos dados de entrada assim como fazer a filtragem das disciplinas cursadas no momento da seleção destas no cadastro. Existe uma lista de disciplinas ofertadas ao qual o aluno irá selecionar e adicionar as disciplinas cursadas. Essa filtragem consiste em ofertar uma disciplina na lista de oferta somente se seu pré-requisito já for adicionado na lista de disciplinas cursadas. Por exemplo, ao adicionar a disciplina de calculo 1 na lista de disciplinas cursadas, pode-se então ofertar a disciplina de calculo 2, que possui como pré-requisito calculo

A funcionalidade de Visualizar Grade Horária consiste na criação da interface gráfica na qual será mostrada as informações referente a grade horária. Nesta interface deve ter o espaço onde estará a grade em si e também as matérias e seus horários.

## 1.2. Objetivos do Sprint Backlog

Com base nos objetivos descritos, foram definidos as seguintes Stories para o Sprint Backlog:

Nome da história	Descrição
US4 - Salvar Grade horária	Como aluno eu quero salvar minha grade horária assim que monta-la
US 8 - Salvar em PDF e m um local específico	Como aluno eu quero gerar um pdf com as informações de aluno e de grade montada.
US 7 - Visualizar grade salva	Como aluno eu quero visualizar a minha grade horária salva
Us1 - Visualizar disciplina	Como usuário, quero visualizar na tela a oferta de disciplinas que eu posso cursar no semestre
US 6 - Excluir grade horária	Como aluno eu quero deletar minha grade horária para removê-la do sistema
US 5 - Alterar grade horária	Como aluno eu quero alterar minha grade horária assim que remontá-la
Us2 - Cadastrar aluno no sistema	Como usuário, eu desejo criar o cadastro no sistema para geração da minha grade horária

Cada história possui seus critérios de aceitação que podem ser visto no documento “SIGHE Critérios de Aceitação” disponível no link:  
[https://www.assembla.com/code/sisghe/subversion/nodes/docs/SIGHE%20Crit%C3%A9rios%20de%20aceita%C3%A7%C3%A3o.pdf?\\_format=raw&rev=235](https://www.assembla.com/code/sisghe/subversion/nodes/docs/SIGHE%20Crit%C3%A9rios%20de%20aceita%C3%A7%C3%A3o.pdf?_format=raw&rev=235)

## **2. SCRUM MASTER - Conhecimentos relacionados a área de gerenciamento de projeto**

Para as atividades do Scrum Master, foi utilizada a ferramenta online icescrum, disponibilizada pelo professor da disciplina, para o controle do processo de desenvolvimento durante a primeira sprint do projeto. Nesta ferramenta, foram geradas oito histórias relacionadas a possíveis funcionalidades que o usuário utilizasse no sistema, quinze tarefas para serem desenvolvidas ao longo da semana em (relacionada a sete das oito histórias criadas), e cinco tarefas recorrentes da parte de gerência do projeto.

Um novo planning poker foi gerado para a pontuação de cada história, dessa vez seguindo como referência as pontuações das histórias da sprint passada (sprint de teste).

## **3. EQUIPE - Conhecimentos relacionados às demais áreas envolvidas no desenvolvimento de um software, considerando a visão da engenharia de software**

Para esta Sprint foi necessário que a equipe também desenvolvesse testes unitários à medida que as classes fossem sendo criadas, embora isso não tenha sido feito na prática. Esta técnica visa obter uma cobertura de código próxima de 90%, na qual é desconsiderada apenas a camada de visão do projeto, ou seja, para a construção da interface gráfica não havia a obrigação de desenvolver testes unitários.

Esta foi uma recomendação imposta desde o começo da Sprint, na qual seria de extrema importância o desenvolvimento de um caso de teste para cada método que fosse criado. Entretanto muitas classes criadas anteriormente ficaram sem seus respectivos testes unitários.

Portanto para as próximas Sprints a equipe deverá se ajustar para manter o padrão de codificação de testes unitários para cada classe criada, na qual também será necessária a criação de novos casos de testes para os métodos já existentes no projeto. Desta maneira será possível alcançar o índice de 90% de cobertura de código, exigido pela disciplina.

Durante esta Sprint para realizar o monitoramento dos custos envolvidos no projeto foi aplicada a técnica de EVM voltada para metodologia ágil. Nesta abordagem o valor agregado foi calculado levando em consideração o número de story points finalizadas e não mais utilizando o percentual completado de atividades. Esta técnica possibilita realizar estimativa de custo em cima de tarefas que já foram concluídas, tornando esta métrica mais precisa. A partir desta Sprint este indicador será usado até o fim do projeto.

## 4. Retrospectiva do Sprint

Assim como na sprint passada, o grupo não usou a ferramenta como deveria, porém os erros foram menores, demonstrando relativa melhora em relação a utilização na sprint anterior. A adição de tarefas no meio da sprint voltou a ocorrer, porém não foi permitida tal ação, sendo então revertida.

Vários problemas foram encontrados em relação a aderência ao processo e suas atividades. Um deles foi a adição de uma tarefa para “done” na ferramenta icescrum enquanto a mesma não estava completa, ou mesmo não comitada. Isso gera uma consistência entre a ferramenta de gerência de projeto e o repositório de arquivos, fazendo com que não seja possível acompanhar o verdadeiro andamento do projeto. Ainda nessa linha, alguns commits que levavam o sistema para um estado de inconsistência (software deixando de funcionar) foram feitos ao longo da sprint.

Todas as histórias planejadas para serem implementadas ao longo da sprint foram implementadas, com exceção de alguns testes unitários. Com isso, vários testes unitários de várias partes da codificação já realizada terão que ser feitos na sprint seguinte.

Todos os problemas (tanto de código fonte quanto de utilização da ferramenta) foram devidamente relatados aos responsáveis para que a correção dos mesmos pudesse ocorrer da forma mais rápida possível.

Apesar desses problemas, a sprint correu de forma muito mais fluída que a sprint de teste, dividindo o trabalho a ser realizado durante sua duração de forma relativamente uniforme.

O grupo de gerência fez a coleta de algumas métricas ao longo da sprint. Em síntese, foi observado que houve um retrabalho de 8,5%, um treinamento de nota 3,8 numa escala de 1 a 5, e índices de desempenho de custos e prazos menores que 1, indicando ineficiência nesses planejamentos. Porém, em relação ao índice de prazo, podemos nos considerar que o projeto está no prazo devido ao fato que tivemos um total de 40 pontos de história sendo que uma dessas histórias (montar grade) tem 21 pontos. Dessa forma, as demais histórias foram alocadas para essa sprint, representando 19 pontos. Como foi executada todas as histórias dessa sprint e ela não representa exatamente 50% do total de pontos de todas as histórias, o SPI ficou um pouco a baixo (cerca 0,95). A ferramenta para análise de cobertura de código não estava devidamente instalada ao longo dessa sprint, permanecendo assim até o final da mesma.

## Referências Bibliográficas

- <http://www.infoq.com/articles/agile-evm>
- <http://www.planningpoker.com/>