

RETROSPECTIVA DO SPRINT 3

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

1 Planejamento do Sprint

1.1. *Objetivos do Sprint*

Essa foi a última Sprint do projeto. O objetivo macro é a conclusão do projeto, e como resultado, o software funcionando.

Essa Sprint consistiu na implementação da seguinte funcionalidade:

- Montar Grade Horaria;

A funcionalidade de Montar Grade Horaria consiste em um aluno cadastrado no sistema ter capacidade de montar sua grade de matérias para o semestre corrente. É disponibilizado uma tabela para o aluno que representa a grade horaria da universidade com os dias e seus respectivos horários. O aluno então pode editar essa grade, selecionando as matérias que lhe é permitido cursar (com base nas matérias já cursadas) e seus respectivos horários.

Também faz parte dos objetivos da sprint a implementação de alguns testes e tarefas que faltaram da sprint anterior. As tarefas recorrentes cabem a gerencia, como coletar métricas, realizar planejamento, analisar o custo e etc., para que no fim desta sprint todo o projeto e suas informações sejam concluídas.

Nesta etapa será finalizada toda a construção do software, ou seja, todo código e também a parte relativa aos testes devem estar prontos. Para esta Sprint restou apenas uma história de usuário, porém os testes unitários correspondentes às classes implementadas estavam incompletos. O objetivo do projeto é realizar uma cobertura de 90% de todo código (exceto pacotes da view e modelo), por isso também foram definidas tarefas urgentes que visam corrigir este atraso e também consertar alguns erros das funcionalidades já concluídas.

1.2. *Objetivos do Sprint Backlog*

Durante o projeto foram implementadas as funcionalidades de manter aluno, manter grade horária e também a geração de relatórios em formato PDF. Para esta sprint restou apenas uma história, porém ela apresenta um alto grau de complexidade, pois envolve validações na interface gráfica que não triviais, além disso ela representa a principal funcionalidade do software, pois engloba toda montagem da grade horária do aluno. Portanto esta última fase do projeto consiste em apenas uma história complexa que se divide em várias tarefas.

Com base nos objetivos descritos no tópico anterior, foi definida uma User Story para o Sprint Backlog: US 3 - Montar Grade. Esta história possui 21 pontos e foi dividida em 11 tarefas, que foram estimadas juntas para terem duração de 65,5 horas.

Cada história do projeto (inclusive a história de Montar Grade) 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/SISGHE%20Crit%C3%A9rios%20de%20aceita%C3%A7%C3%A3o.pdf?_format=raw&rev=235.

Para essa sprint, o escopo foi reduzido para se adequar as modificações feitas no prazo, e portanto foi removido o critério de aceitação seguinte relacionado a história de editar grade horária:

- Dado que a grade foi visualizada - Quando o usuário clicar em um campo da tabela - Então o sistema apresenta uma lista de disciplinas que podem ser cursadas naquele horário.

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. Para essa Sprint, relacionada à história de Montar grade, foram criadas onze tarefas para serem desenvolvidas ao longo da Sprint, cinco tarefas recorrentes da parte de gerência do projeto e 18 tarefas urgentes. Essas tarefas urgentes são relacionadas à tarefas que ficaram pendentes da última Sprint do projeto, e necessitavam serem concluídas o quanto antes possível, como a elaboração de inúmeras classes de teste e problemas que algumas funcionalidades apresentavam. Não foi possível concluir todas as tarefas urgentes dentro do período da Sprint, faltando 2 das 18 tarefas.

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

A equipe desenvolveu ao longo da sprint a história descrita no tópico 1 deste documento, que continha 21 pontos. Em contraste com a sprint anterior, foram planejados 2 pontos adicionais.

Para esta Sprint, assim como na anterior, 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 pra todas as classes. 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 no fim do projeto ficaram sem seus respectivos testes unitários, tornando a qualidade do código duvidosa.

4. Retrospectiva do Sprint

O principal impedimento, que deveria ser um auxílio à equipe de desenvolvimento, foi a ferramenta de controle de versão. Devido a mal uso pela equipe de desenvolvimento, as últimas versões do projeto não estavam devidamente postadas no repositório, dificultando o trabalho da gerência de acompanhamento do projeto. Foram criados pelos membros da equipe repositórios locais, onde o trabalho era realizado individualmente e divulgado para outros integrantes utilizando meios físicos de armazenamento.

A utilização da ferramenta foi melhor que nas sprints anteriores, alcançando níveis de uso aceitáveis para utilização efetiva da mesma. Consequentemente, o gráfico de burndown e burnup da sprint foram melhor apresentados, mostrando sua utilidade para análise de trabalho realizado e trabalho a se realizar.

É importante observar que a sprint foi planejada para ter cerca de 4 dias adicionais em função da data de apresentação ter sido prolongada anteriormente, e que a redução brusca no tempo de entrega fez com que desestabilizasse o ritmo de trabalho da equipe de desenvolvimento, atrapalhando assim o seu desempenho. Em função disso, o escopo foi reduzido em 1 critério de aceitação, que representava uma micro funcionalidade relacionada a funcionalidade geral de montagem de grade horária.

Outra consequência da redução do tempo de trabalho previamente planejado pela equipe de gerência foi o esforço destinado a testes ao longo da sprint. Os testes estavam sendo feitos em paralelo com o desenvolvimento, e a mudança repentina fez com que houvesse uma priorização das tarefas que afetavam diretamente a funcionalidade do software, isto é, tarefas necessárias para que a funcionalidade fosse implementada. Visto isso, a qualidade do código deixou a desejar em alguns módulos, pois o foco final foi entregar a funcionalidade a qualquer custo, o que não deveria ter ocorrido, afinal um produto só deve ser entregue e só é considerado pronto quando

está devidamente implementado e testado em todos os seus níveis. Se o prazo tivesse sido mantido, os testes poderiam certamente ter sido implementados, e isso refletiria na melhora da qualidade do produto final.

Apesar de todos os problemas, a comunicação entre as equipes de gerência e desenvolvimento deu mais um passo. Isto significa que as reuniões estavam mais dinâmicas e com mais participantes ao longo dessa última sprint, fazendo com que fossem mais produtivas e também com que o trabalho de toda a equipe, tanto de gerência como de desenvolvimento, ficasse mais transparente para os integrantes. O velocity da equipe estava cada vez melhor calibrado para o projeto, e as estimativas ficariam cada vez mais próximas do real se o projeto se prolongasse por mais algum tempo.

Referências Bibliográficas

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