Relatório Final do Projeto SisGHE

Pedro Henrique Potiguara Carvalho
Marcos Ronaldo Pereira Júnior
Guilherme de Lima Bernardes
Carlos Filipe Lima Bezerra

Brasília, 2013

Sumário

1. Introdução

Todo início de semestre é necessário que os alunos da Universidade de Brasília montem suas grades horárias com base nos horários das matérias disponibilizados no site do matrículaweb, e esta muitas das vezes é uma atividade desgastante e que ocupa muito tempo, podendo ocorrer até mesmo erros durante a montagem.

O objetivo deste projeto é a implementação de um software denominado SisGHE, capaz de gerar de forma automática uma grade horária de acordo com as matérias que o aluno desejar, dentre aquelas que forem possíveis dele cursar (levando em consideração pré- requisitos).

O escopo do projeto consiste no planejamento, análise e desenvolvimento do sistema, afim de construir um processo de desenvolvimento e um produto executável que corresponda as características descritas nos artefatos.

Com a utilização deste software, será possível a montagem rápida e confiável de diversas grades horárias pelos alunos da UnB, de forma a facilitar a escolha de quais matérias cursar, tendo como base diversas grades horárias geradas pelo software.

Através deste documento, busca-se elaborar um plano de projeto capaz de determinar toda a equipe que fará parte deste projeto, suas respectivas responsabilidades e as atividades que serão desenvolvidas ao longo do projeto.

2. O projeto

Todo início de semestre é necessário que os alunos da Universidade de Brasília montem suas grades horárias com base nos horários das matérias disponibilizados no site do matrículaweb, e esta muitas das vezes é uma atividade desgastante e que ocupa muito tempo, podendo ocorrer até mesmo erros durante a montagem.

O objetivo do projeto foi a implementação de um software denominado SisGHE, capaz de gerar de forma automática uma grade horária de acordo com as matérias que o aluno desejar, dentre aquelas que forem possíveis dele cursar (levando em consideração pré- requisitos).

O escopo do projeto consistiu no planejamento, análise e desenvolvimento do sistema, afim de construir um processo de desenvolvimento e um produto executável que corresponda as características descritas nos artefatos.

Com a utilização deste software desenvolvido, é possível a montagem rápida e confiável de diversas grades horárias pelos alunos da UnB, de forma a facilitar a escolha de quais matérias cursar, tendo como base diversas grades horárias geradas pelo

software.

3. Tracking e Métricas

O objetivo do gerenciamento de qualidade em um projeto de software é garantir que, mesmo no desenvolvimento de um software grande e complexo, sejam atendidos níveis significativos de qualidade, tanto relativos ao produto, quanto relativo ao processo, através de um processo minucioso de mensuração.

Para analisar o projeto ao longo de sua execução e poder manter o controle do mesmo, algumas informações são necessárias. Ao início do projeto, a gerência considerou necessário ter o controle sobre os seguintes aspectos: cumprimento dos prazos, qualidade de código fonte, custo do projeto e eficácia do treinamento. Para isso, foram definidas 4 métricas: a métrica de SPI, complexidade ciclomática, CPI e EFT (eficácia de treinamento). A métrica de complexidade ciclomática foi retirada pois sua utilização não demonstrou resultados satisfatórios, isto é, o grupo não soube analisar os resultados das medições a ponto de conseguir utilizar a métrica para melhorar o processo de desenvolvimento. Portanto, em uma segunda parte da análise da qualidade do projeto, a métrica de complexidade ciclomática foi substituída pela métrica de retrabalho.

3.1 Custos

Para o desenvolvimento de um projeto é imprescindível realizar um planejamento de todo o custo e consequentemente o seu monitoramento ao longo do ciclo de vida. Dentro do contexto em que se encontra o projeto SisGHE foi realizado uma estimativa dos custos com o objetivo de calcular o valor total. Para isto foram utilizados como parâmetros alguns itens importantes, como por exemplo, valor mensal de energia elétrica, custo de horas trabalhadas por cada integrante da equipe, valor cobrado pelo serviço de internet banda larga, valor individual de folhas impressas e aquisição de equipamentos.

Para obter o custo individual por hora trabalhada de cada integrante da equipe foi utilizado o valor anual de custo da universidade com cada aluno. Este montante é disponibilizado pela FUB (Fundação Universidade de Brasília) no relatório de prestação de contas de 2011[1], disponível em no qual a parir deste valor anual se calculou o custo de um estudante por hora, ou seja, o valor utilizado para planejar o custo.

Os valores de energia elétrica foram calculados de acordo com o preço cobrado pela CEB (Companhia Elétrica de Brasília) por kWh (quilowatt-hora) multiplicando o número de horas trabalhadas. Já os preços de aquisições de equipamentos foram obtidos com base em notebooks da marca LG com processador Intel core i5, 4GB de

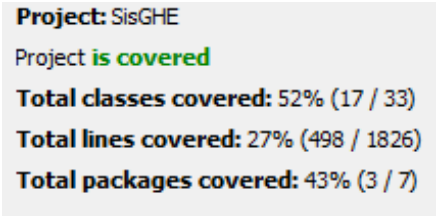
memória. O valor de internet foi obtido com base no plano NET Vírtua de 10MB.

Uma planilha detalhada, contendo valores de custo planejado, valores agregado, custos atuais, e gráficos, tanto da primeira metade do projeto, desenvolvida por planejamento seguindo o processo RUP, quanto a segunda metade do projeto, seguindo o planejamento pelo método de desenvolvimento SCRUM, pode ser visualizada no repositório do projeto.

3.2 Cobertura

A cobertura de testes vêm com a responsabilidade de definir a quantidade de abrangência de testes que um software possa ter.

Especificamente para o projeto SisGHE, a cobertura de testes foi relativamente baixa, contando apenas com 52% de cobertura, como pode ser visualizado na imagem abaixo:



Vários fatores resultaram nesta resultado insatisfatório, mas o principal que pode ser citado é o alto nível de esforço que necessitou ser realizado na segunda release do projeto, devido ao acumulo de testes não implementados na primeira release.

Apesar da cobertura não ter sido correspondida aos 90% esperado, vale ressaltar que ao menos foi obtido uma cobertura acima dos 50% de testes ao projeto.

3.3. Retrabalho

A taxa de retrabalho pode ser visualizada na tabela abaixo:

	Até Sprint 1	Até Sprint 2

Horas Gastas em Retrabalho	31	43
Percentual de retrabalho	9,85%	8,56%

Essas taxas são referentes à mudança de metodologia de desenvolvimento de RUP para SCRUM, onde foi necessário um dia de retrabalho para a refatoração do código, devido à inúmeros problemas, principalmente à compreensão do código e à falhas de implementação do modelo de MVC.

Na segunda sprint, foram gastos tempo de retrabalho com relação à correção de bugs e diversos problemas de quebra de arquitetura.

De forma geral, a taxa de retrabalho foi reduzida ao longo do projeto, uma vez que um acompanhamento mais intensivo foi realizado, e consequentemente uma melhoria dos resultados da implementação.

4. A Equipe

A equipe do projeto era composta por 9 integrantes, sendo 5 deles da disciplina de MDS, compondo assim a equipe de desenvolvimento (codificação), e o restante dos integrantes são alunos da disciplina de GPP, que representam a gerência da equipe.

Para a organização de todo o time, foram necessárias reuniões semanais, tanto presenciais, com o time inteiro presente ao menos uma vez na semana, quanto virtuais, onde cada time se reunia para realização das tarefas que estivessem em aberto.

Para planejamento de sprints, de release e também restrospectiva de sprint, a equipe inteira se reunia através do Skype em uma data marcada com cerca de 3 a 7 dias de antecedência. Assim, para as decisões importantes que tem impacto direto em todos os membros do time, foi garantido que a maioria dos integrantes estivesse presente.

5. Conclusões e Resultados

5.1 Software

O resultado do projeto foi o software de montagem de grade horária estudantil. O software possui recursos de cadastro do aluno, no qual o aluno informa as matérias já cursadas. Essa informação é importante pois determinará quais as matérias o aluno está apto a pegar e o software irá disponibilizar apenas estas para o aluno pegar.

O recurso principal é o de montagem de grade. Nesse recurso o aluno seleciona as matérias a fim montar a grade do semestre corrente. São selecionadas disciplinas e seus respectivos horários. Em caso de choque, o programa apontará ao usuário. São ofertadas apenas as matérias que é possivel cursar, baseada nos pré requisitos feitos já cadastrados juntamento com o cadastro de aluno.

Não foi possivel implementar uma função da funcionalidade de montagem de grade horária que consiste em clicar no área vazia da tabela de grade e mostrar quais matérias encaixam nesse horário.

O software ainda oferece a exportação da grade e dos dados do aluno em formato pdf.

5.2 Interação da Equipe e Processo de Desenvolvimento

No periodo em que o processo de desenvolvimento era o RUP, a comunicação entre as equipes (MDS/GPP) não foi muito eficiente. Resultado disso foi que o planejamento (cronograma) não teve muita utilidade, visto que a equipe de MDS e GPP realizavam seus trabalhos e entregas de maneira um tanto que independente. Todos estavam sientes do que tinham que entregar e prepararam suas entregas sem muita integração e comunicação.

Mesmo com a falha na comunicação o software saiu, porém o processo e o controle ficaram desorganizados.

Com a virada do tipo do processo para métodos ágeis, os erros de organização foram consertados. Foi estabelecido reuniões semanais (DailyScrum) com todos os integrantes, o que impactou na melhoria de comunicação pelos meios virtuais também. A gerêcia do projeto melhorou. O acompanhamento foi feito com mais frequência, facilitando na identificação de bugs no projeto. Foram planejadas 3 sprints: a primeira de uma semana, e as outras de duas semanas.

A primeira sprint foi apenas de teste com o objetivo de aprender a mecher na ferramenta IceScrum e dar os primeiros passos em métodos ágeis. Não ocorreu de maneira correta. As tarefas propostas nao foram feitas, então ela foi desconsiderada e aproveitada apenas o aprendizado para as próximas.

Na segunda sprint, foram criadas todas as hitórias que iriam compor o projeto e feito o planejamento e a alocação das histórias nas sprints. Falhamos em não estimar o tempo de duração das atividades logo de início, o que resultou em um burndow completamente errado. Ficou acordado entre as equipes que ao executar uma atividade, deveria implementar também os testes, ou seja, a atividade só poderia ser considerada concluída caso fosse feito os testes também. Porém, alguns desenvoldores esqueciam desse detalhe, o que resultou em muitas tarefas urgentes relacionadas a testes na

terceira sprint.

Na terceira sprint nao foi possível realizar a implementação da funcionalidade de clicar no horário vago na tabela e visualizar quais matérias encaixariam naquele determinado horário, como também não foi possível a conclusão de todos os testes especificados nas terefas urgentes. Podemos de certo modo responsabilizar essas falhas a redução do tempo planejado, visto que foi previsto duas semanas para a sprint e foi feita em uma semana e meia. Caso a sprint tivesse duas semanas, provavelmente teríamos alcançado a meta de cobertura de código.

Podemos perceber que a equipe progrediu em relação ao tempo, melhorando a relação entre membros e a comunicação.

6. Experiência final

Muitas experiências foram acrescentadas para o grupo da gerência em si, mas a que com certeza foi de maior valor foi a resolução de problemas entre as pessoas (integrantes do grupo), de forma todos entendessem que faziam parte de um grupo, e que cada um necessitava do outro para que pudesse ser alcançado o objetivo final da disciplina: um software que atendesse às necessidades levantadas.

Sem dúvidas, a equipe de gerência aprendeu bastante com relação à como conduzir uma equipe e ouvir a todos quando as idéias se opõem e necessita-se chegar a um consenso.

Diferente do que muitos pensam, a área de engenharia de software não se resume a tecnologia e raciocínio lógico. O maior desafio é saber utilizar os conhecimentos científicos em um ambiente cooperativo onde a interação humana faz toda a diferença no resultado do trabalho como um todo. É importante saber lidar com os companheiros de equipe, saber valorizar o trabalho do outro, e reconhecer os erros que se comete, crescendo como profissional.