

Engenharia de Software

Parte 4 - Estrutura Analítica

Grupo 7 - Iago e Thomas

Desenvolvimento ágil de software

- Busca minimizar o risco pelo desenvolvimento do software em curtos períodos com duração de menos de uma semana até um mês;
- Cada iteração é como uma miniatura do próprio projeto incluindo novas funcionalidades;
- Métodos ágeis enfatizam a comunicação em tempo real entre todos do projeto e a redução de documentação.

Valores Fundamentais - Manifesto Ágil

- Os indivíduos e suas iterações acima de procedimentos e ferramentas;
- O funcionamento do Software acima de documentação abrangente;
- A colaboração com o cliente acima da negociação e contrato;
- A capacidade de resposta a mudanças acima de um plano pré-estabelecido.

Princípios - Manifesto Ágil

1. **Garantir a satisfação do cliente, entregando rápido e continuamente softwares funcionais;**
2. **O Software funcional é a principal medida de progresso;**
3. **A capacidade de resposta a mudanças acima de um plano pré-estabelecido;**
4. **Simplicidade: A arte de maximizar a quantidade de trabalho que não precisou ser feito;**
5. **Construir projetos ao redor de indivíduos motivados;**
6. **Contínua atenção à excelência técnica e bom design, aumenta a agilidade.**

Framework Scrum

- Possui um formato dinâmico nas etapas de desenvolvimento de projetos;
- Utiliza sprints (iterações) começando do planejamento do sprint e terminando numa revisão ou retrospectiva do que foi feito;
- Os sprints são geralmente curtos e um novo só é iniciado ao fim do anterior;
- O desenvolvimento é acompanhado por reuniões diárias;
- Três papéis são definidos no Scrum: ***Product Owner***, ***Scrum Master*** e ***Scrum Team***.

Product Owner

Atua como o “dono” do projeto. Responsável por definir prioridades a serem desenvolvidas em cada sprint e fazer a intermediação entre a área de negócios e a equipe de scrum.

Scrum Master

Atua como um líder-servo, blindando os demais membros e assegurando que a equipe siga a metodologia do Scrum sem interrupções externas.

Também é responsável por remover obstáculos que possam prejudicar o desenvolvimento realizado pela equipe e ajudá-la a cumprir suas tarefas com a melhor performance possível.

Scrum Team

É a equipe de desenvolvimento. Todos devem se comprometer em realizar as entregas estabelecidas dentro de uma sprint.

É importante para a eficiência que não haja muitos membros nessa equipe e que eles sejam membros multidisciplinares.

As 4 Iterações do Scrum

Sprint Planning

O planejamento da sprint é realizado nessa reunião, decidindo como o trabalho da equipe será feito dentro do período estabelecido. Nesse momento, devem ser priorizadas as descrições de todas as funcionalidades desejadas para o produto, que passarão a integrar o *sprint backlog*, composto pelo repositório de projetos e ações da empresa.

A cada sprint seleciona-se de acordo com a prioridade e dificuldade do projeto, os projetos que sairão do *backlog* e entrarão para o sprint.

Daily Scrum

Diariamente, a equipe de desenvolvimento deve se reunir para discutir aquilo que está sendo desenvolvido dentro da sprint. A reunião deve ser rápida e três perguntas devem ser respondidas por cada membro da equipe:

- **O que foi feito ontem para ajudar a equipe de desenvolvimento e atender a meta da sprint?**
- **O que será feito hoje para ajudar a equipe de desenvolvimento a atender a meta da sprint?**
- **Há algum obstáculo que impeça o atendimento da meta da sprint?**

Se a resposta da última pergunta for afirmativa, cabe ao *Scrum Master* buscar uma solução para o problema.

Sprint Meeting Review

Ao final da sprint, uma reunião de revisão é realizada para a discussão daquilo que foi desenvolvido naquele ciclo.

Cabe ao *Product Owner* analisar se cada tarefa foi concluída conforme o esperado ou se alguma delas deverá retornar ao *Product Backlog*.

Sprint Retrospective

A reunião de retrospectiva da sprint é realizada após a reunião de revisão e anteriormente à reunião de planejamento, para que seja discutido um plano de melhorias.

O objetivo da reunião é promover a colaboração entre os membros da equipe de desenvolvimento para corrigir possíveis desvios de rota e aprimorar aquilo que já está nos trilhos.

Métrica de Software

Métrica de Software

É uma técnica de mensuração aplicada no processo de desenvolvimento de software ou atributos de um produto com o objetivo de melhorá-lo de forma contínua, que quando utilizado ao longo do projeto auxilia na estimativa, no controle de qualidade, na avaliação da produtividade e no controle do projeto.

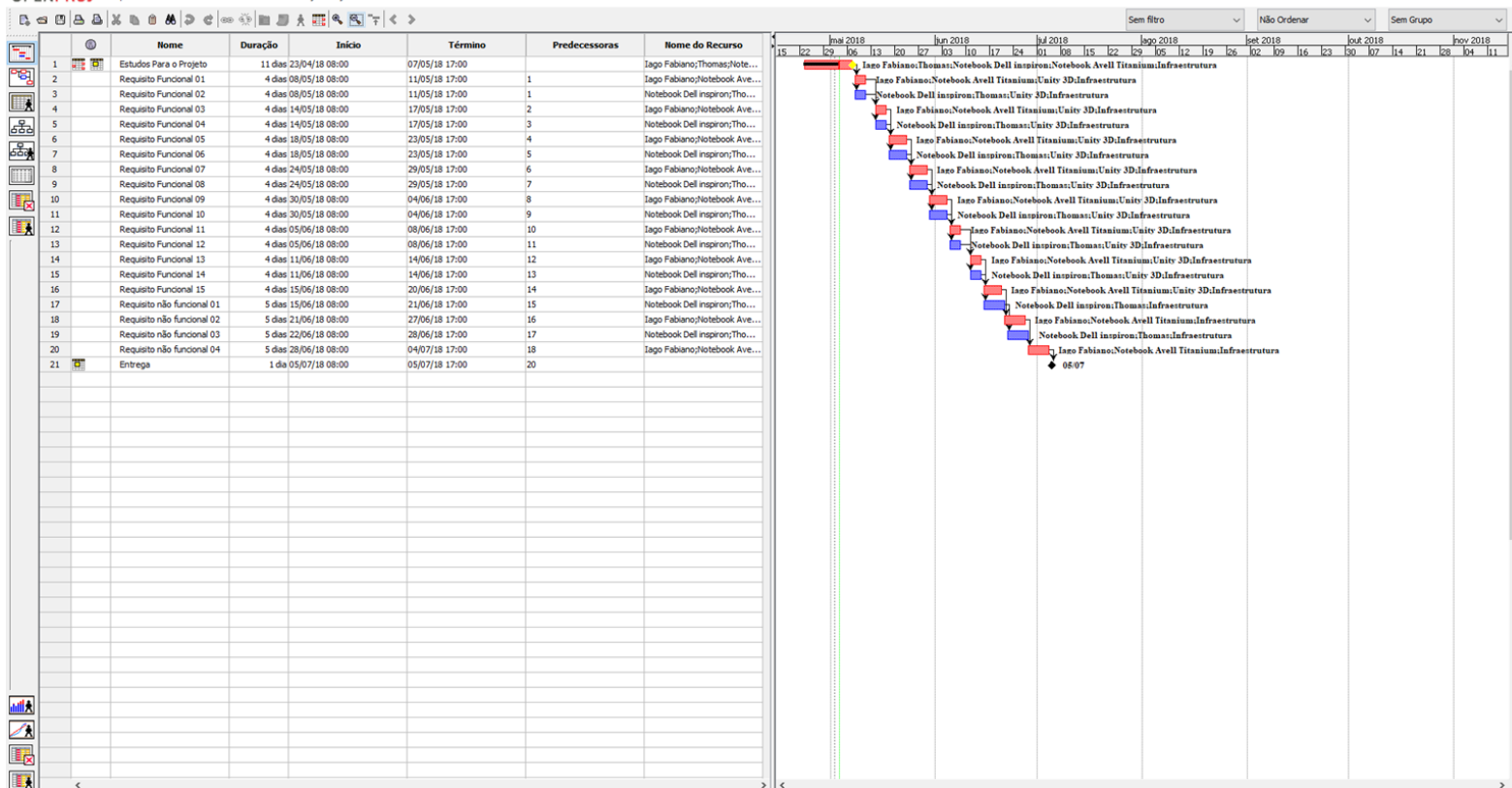
Ideal Day

Corresponde à quantidade de trabalho que um profissional da área consegue concluir em um dia de trabalho.

A velocidade é calculada a partir do número de horas que a equipe gasta para implementar um trabalho equivalente a um *Ideal Day*. Caso o item passe um dia de trabalho, é sugerido decompor esse item em itens menores que se consiga implementar em apenas um dia.

OpenProj

OPENPROJ[®] Arquivo Editar Exibir Inserir Ferramentas Projeto Ajuda



Jogo - Engenharia de Software - Grupo 7

Dates			
Start	02/05/18 08:00	Finish	05/07/18 17:00
Baseline Start		Baseline Finish	
Actual Start	23/04/18 08:00	Actual Finish	

Duration			
Scheduled	47 dias	Remaining	47 dias
Baseline	0 dias	Actual	0 dias
		Percent Complete	9%

Work			
Scheduled	824 horas	Remaining	683,2 horas
Baseline	0 horas	Actual	140,8 horas

Costs			
Scheduled	R\$ 72300,00	Remaining	R\$ 59940,00
Baseline	R\$ 0,00	Actual	R\$ 12360,00
		Variance	-R\$ 12360,00

Notes			
Jogo - Engenharia de Software - Grupo 7			

Relatório

Relatório Gerado pela da ferramenta case OpenProj

Onde após colocar as tarefas e suas durações e atribuir a elas os recursos, a ferramenta calcula todos os dados do projeto.