
ENUNCIADO DO TRABALHO

Disciplina de Engenharia de Software

Engenharia de um sistema à sua escolha

INTRODUÇÃO

Este enunciado detalha a avaliação na disciplina de Engenharia de Software que será feita por meio de um trabalho. Serão 4 entregas de 10, 20, 20, 20 pontos, cada avaliada com apresentação dos alunos em aula presencial. As datas tentativas das apresentações serão **xxxxxxxxxx**.

1a. Entrega: 1) Concepção de um sistema a escolha do grupo. 2) Definição do ferramental a ser utilizado. 3) Requisitos Preliminares do sistema apresentado no Github. 4) Planejamento do trabalho apresentado na ferramenta escolhida.

2a. Entrega: 1) Revisão da concepção e estudo de viabilidade. Especificação dos requisitos completa. Projeto Arquitetural Preliminar

3a. Entrega: 1) Revisão dos artefatos produzidos. 2) Apresentação da Arquitetura e Resultado das primeiras iterações de implementação.

4a. Entrega: Trabalho completo.

ENTREGÁVEIS — *DELIVERABLES*

A sua equipe deverá entregar uma série de artefatos para completar o trabalho desta disciplina. Todos os artefatos deverão ser entregues em um projeto no github.com. A Figura 1 mostra em linhas gerais como funciona a criação de um projeto no GitHub. Todos os artefatos devem ser carregados no GitHub. Os artefatos deverão ser carregados em suas diversas versões que deverão ser controladas pelo GitHub.



Figura 1: Passos gerais para trabalhar no GitHub

CONCEPÇÃO DO SISTEMA E ESTUDO DE VIABILIDADE

Neste documento, a equipe deverá mostrar quais serão as principais funcionalidades que serão incorporadas na ferramenta, preferencialmente demonstrando os seus diferenciais competitivos. Neste ponto é possível que a equipe exagere no tamanho do sistema e produza um sistema cuja entrega final

não seja possível. Logo, é necessário demonstrar a viabilidade de conclusão do projeto dentro do prazo disponível.

PLANEJAMENTO E GERENCIAMENTO DO PROJETO

Neste documento, a equipe deverá mostrar quais serão as principais atividades a serem desenvolvidas com o projeto. É necessário usar um sistema de gerenciamento de projeto. Sugerimos que escolha um sistema online livre para fazer este gerenciamento. Uma sugestão é o uso do Trello, mas o grupo pode escolher outro. A equipe deve modelar o processo de software que vão utilizar, descrevendo quais são as principais atividades. Sugere-se o uso de diagramas de atividades da UML.

ESPECIFICAÇÃO DOS REQUISITOS DO SOFTWARE

A equipe deverá documentar a análise de requisitos do software seja através de uma tabela enumerando cada requisito e/ou através de documentos de casos de uso com suas respectivas descrições. Na página 66 do livro do Sommerville 9ª edição (Capítulo Engenharia de Requisitos) são mostradas alternativas para especificação de requisitos. A especificação de requisitos deve ser carregada no GitHub em formato PDF.

PROJETO DA ARQUITETURA DO SISTEMA

A equipe deverá documentar a arquitetura do sistema com diagramas de blocos tradicionais. Poderão ser utilizados diagramas de componentes e/ou implantação da UML. A descrição arquitetural deve ser carregada no GitHub em formato PDF. É importante notar que os requisitos vão impactar fortemente na arquitetura. Por exemplo, se as informações pessoais ficarem armazenadas na rede de alguma forma, haverá necessidade de sincronização com a rede o que impacta na arquitetura. Caso contrário, a arquitetura fica mais simplificada neste aspecto.

IMPLEMENTAÇÃO DO SISTEMA

A implementação do sistema deverá ser contínua e realizada através de *commits* no GitHub. A própria história de *commits* mostrará o trabalho da equipe durante o desenvolvimento do sistema. Cabe ressaltar novamente que não necessariamente (aliás, nem é indicado) que o sistema seja desenvolvido “do zero”. Quanto mais código e componentes a equipe reutilizar melhor vai ser a chance de conseguir um produto final de melhor qualidade. A equipe deve tomar cuidado com o escopo do projeto para conseguir fazer iterações que produzam incrementos interessantes do sistema.

PLANEJAMENTO E EXECUÇÃO DE TESTES COM RESPECTIVAS MUDANÇAS

Os casos de testes principais a serem executados no software deverão ser documentados em um arquivo PDF. Preferencialmente (ponto-extra para no mínimo 10 testes de unidade), os testes de unidade devem ser codificados em alguma ferramenta como, por exemplo, JUnit para quem usar Java. Deve ser montada uma equipe separada de testes, a qual deverá se basear no documento de casos de testes para executar os testes. Os defeitos que venham ser encontrados por esta equipe deverá ser reportado na ferramenta de “Issue Tracker” do GitHub. A equipe de manutenção e desenvolvimento deverá estar atenta aos erros levantados e os *commits* de correção deverão ser ligados às respectivas *issues/bugs* levantados.

ANÁLISE E DISCUSSÃO DE ASPECTOS DE MELHORIA DE PROCESSO

A equipe deve produzir um documento mostrando as principais dificuldades encontradas durante o processo de elaboração do trabalho e as alternativas que poderiam ser adotadas (preferencialmente

ranqueadas) para mitigar tais dificuldades. Deve ser apresentado no final também como uma forma de auto-avaliação.

DICAS FINAIS

1. Comece a inicializar o ambiente o mais rapidamente possível.
2. Programe reuniões rápidas no mínimo semanais da equipe
3. Diminua o risco de não conseguir terminar, fazendo um desenvolvimento incremental. Consiga um incremento funcional já nas primeiras semanas. É melhor entregar um “pouco” que funcione do que um “muito” que nem compila e executa.
4. Defina o papel de cada elemento do grupo e cadastre cada elemento no GitHub. Na minha opinião é fundamental eleger um líder. O líder deve cobrar dos demais membros da equipe de maneira construtiva.