

Design de Software

Mestrados em Informática e Engenharia Informática

Exercício de Avaliação

2019/2020

O objetivo deste exercício é que os alunos apliquem e consolidem alguns dos conceitos relacionados com arquitetura de software que estão a ser lecionados na disciplina. O exercício deve ser resolvido em grupo de 3 elementos. A entrega é feita através da página da disciplina, até 24 de Novembro.

Problema

O foco principal do trabalho é a análise e documentação da arquitetura de um sistema de software existente a escolher entre:

TornGest (versão 1.5) — uma aplicação empresarial de gestão de provas desportivas, desenvolvida com Java EE por um grupo de ex-alunos da LEI no contexto da disciplina de CSS, e disponível na página de DS no moodle

Aqualush — uma aplicação que controla um sistema de irrigação, desenvolvida em Java por C. Fox e disponível na página de DS no moodle

A análise de um outro sistema de software também é possível, desde que seja considerado apropriado para o efeito.

Contexto

Devem realizar este trabalho **imaginando** que

- fazem parte da equipa de um projeto que depende fortemente desse sistema (por exemplo, um projeto que visa desenvolver uma versão comercial do *TornGest* ou do *Aqualush*)
- ambicionado acumular experiência, ofereceram-se para o papel de arquiteto do sistema e a vossa oferta foi aceite!

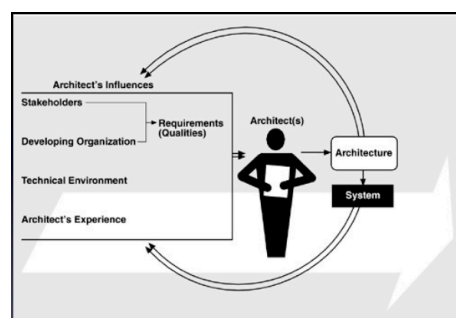


FIGURE I Architecture Business Lifecycle de acordo com o livro [BCK13]

Tarefas

1. Pretende-se que, a partir do código fonte do sistema e de informação adicional que consigam obter, produzam um SAD documentando a arquitetura do sistema escolhido.

Considerando o propósito (fictício) deste documento, devem complementar as decisões arquiteturais que conseguirem extrair do código ou obter de outras fontes, com outras decisões que considerem apropriadas. Por exemplo, podem complementar os atributos de qualidade que conseguirem inferir terem sido levados em consideração no desenho do sistema, com outros que sejam cumpridos pela sua arquitetura e que considerem importantes levar em consideração em futuras evoluções ou alterações do mesmo.

A análise e documentação da arquitetura do sistema deve ser feita à luz dos conceitos estudados na disciplina.

O documento deve seguir, no geral, o estilo do SAD apresentado em [C11] e exemplificado no *Adventure Builder*. É fornecido o *template word* disponibilizado pelo SEI e têm ainda uma versão wiki deste *template* em

https://wiki.sei.cmu.edu/sad/index.php/Software_Architecture_Documentation_Template

Em particular, o documento deve

- identificar quais os objetivos do sistema, as suas funcionalidades, e principais características;
 - identificar quem são os principais *stakeholders* do sistema;
 - identificar os requisitos de qualidade mais importantes, formulando pelo menos alguns deles em termos de cenários de qualidade;
 - apresentar diferentes vistas arquiteturais do sistema, recorrendo às notações gráficas e textuais adequadas;
 - identificar *architectural drivers* assim como importantes decisões de desenho / táticas / padrões / estilos e o seu racional
 - identificar as fontes de informação adicionais usadas para obter a informação apresentada e identificar claramente as decisões da vossa responsabilidade
2. Pretende-se que sujeitem o código fornecido à análise pela ferramenta *SonarQube*. Se ainda não sabem como obter e correr a ferramenta, podem seguir as instruções aqui:

<https://docs.sonarqube.org/display/SONAR/Get+Started+in+Two+Minutes>

- A ferramenta produz vários relatórios com valores para diferentes métricas. Explore as diferentes informações que a ferramenta oferece e analisem os resultados de forma a conseguirem apresentar um resumo dos principais problemas detetados, identificando os atributos de qualidade afetados. Notem que a entrega de *screenshots* com outputs da ferramenta **não é** o pretendido!
- Explore também a ajuda que a ferramenta dá a remediar os problemas identificados. Escolham um conjunto de problemas e resolvam-nos. Analisem os efeitos das mudanças que efetuaram nos resultados da análise do projeto.
- **(Opcional/Bónus)** Seguindo o processo que é descrito no link abaixo, podem ver se conseguem usar o *plugin SonarGraph* para o *SonarQube* para analisar dependências cíclicas:

<http://blog.hello2morrow.com/2017/01/use-sonarqube-sonargraph-plugin-to-detect-cyclic-dependencies/>

No caso de terem sucesso, reportem os resultados da análise e proponham possíveis soluções.

Referências

- [BCK13] **Software Architecture in Practice (3rd edition)**. Len Bass, Paul Clements, Rick Kazman, Addison Wesley, 2013.
- [C11] **Documenting Software Architectures: Views and Beyond (2nd edition)**. Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford, Addison Wesley, 2011.