[Software Quality Assurance Plan](https://ocw.cs.pub.ro/courses/mps/laboratoare/laborator-09#sqap_software_quality_assurance_plan) - Echipa 1

1. Scopul documentului (Purpose)
2. Documente referite
3. Management
4. Documentație
5. Standarde, practici, convenții, metrici
6. Revizii și audituri
7. Managementul riscului
8. Raportarea problemelor și acțiuni de corecție
9. Utilitare, tehnici și metodologii
10. Controlul furnizorului
11. Training
12. Colectarea înregistrărilor, mentenanța
13. Scopul documentului este acela de a prezenta planul de asigurare a calitatii proiectului Echipei 1, un proiect ce are ca scop implementarea unui software dedicat jocurilor de societate.
14. Documente referite

* IEEE Guide for Software Quality Assurance Planning, IEEE Std 730.1 – 1995
* IEEE Standard for Software Quality Assurance Plans, IEE Std 730 – 1998

1. Documentatie:

* Lecture notes, CIS 748 Software Management, Dr. Scott Deloach, Spring 2002
* Lecture Notes, CIS 771 Software Specifications, Dr. John Hatcliff, Spring 2001
* Software Engineering, Roger S. Pressman, 5th Ed.

1. Management

ORGANIZARE:

Structura echipei este formata din Project manager, Team lider, 2 developeri si un tester.

- Bunea Andrei, 341C4 - software developer

- Constandache Florina, 343C5 - team leader

- Elefterescu Cristina, 343C3 - project manager

- Sorescu Teodora, 341C2 - software developer

- Stefan Marinela, 343C5 - tester

TASKS AND RESPONSABILITIES:

Developer/Tester:

-implementarea aplicatiei si testarea aplicatiei

Team leader:

-dezvolta planul de proiectare si planul de testare impreuna cu project managerul

Project manager:

-dezvoltarea documentatiei

1. Standarde, practici, convenții, metrici

Metrici: LOC - lines of code is used to measure the size of the software

Standarde: Coding standards, Coding Documents standards – Java Documentation

6. Revizii si audituri:

- Revizia va fi facuta in intalnirile saptamanale ale echipei pentru a dezbate noile modificari, problemele aparute si solutii pentru urmatoarele etape ale dezvoltarii

7. Managementul riscului:

* Se va evita pe cat posibil dependenta de funtionalitati de stocare a datelor externe.
* Se va implementa intern toata logica proiectului
* In cazul folosirii unor API-uri se va face un research constant asupra potentialelor probleme si schimbari in structura serviciului extern

8. Raportarea problemelor si actiuni de corectie:

* Pe durata implementarii se vor efectua teste in mod activ, raportand bug-urile aparute pe platorma github, urmand sa fie rezolvate de catre developeri in ordinea importantei

9. Utilitare, tehnici si metodologii:

* Se vor folosi ca utilitare: Intellij IDEA, Github, Oracle Database si VsCode
* Ca tehnici vom folosi: Java Spring si React
* Ca metodologii vom impementa: Agile

10. Controlul furnizorului:

* Stabilirea unor sesiuni cuprinzatoare de testare pe parcursul, dar in special la finalul implementarii proiectului

11. Training:

* Proper Web Development course by CS50, Standford Lectures 2016
* Data Flow Management, coursera lectures
* Design management in React, coursera lectures

12. Colectarea inregistrarilor, mentenanta:

* Datele vor fi colectate in baza de date.
* Informatii aditionale vor fi afisate in fisiere de tip logger