

# Raport de dezvoltare

**Planul de risc:**  Plan de risc

## Progres al echipei în dezvoltarea soluției:

Proiectul a fost împărțit în 3 sprinturi. În primul sprint s-au analizat algoritmi pe care i-am putea utiliza și s-a ales unul concret. S-a făcut citirea datelor și plasarea thresholdurilor inițiale.

În sprint-ul 2 s-a lucrat la dezvoltarea soluției pentru partea de binarizare globală. În acest sprint echipa de dezvoltatori și-au definit clasele necesare implementării soluției și au persistat datele de intrare. S-au făcut progrese la identificarea algoritmului de generare a arborelui și se lucrează la integrarea acestuia în proiect. În urma acestui pas, se va verifica validitatea construcției arborelui și se va putea antrena modelul pentru a se obține un threshold final cât mai bun.

Au fost respectate toate deadline-urile iar proiectul progresează în ritmul așteptat.

Echipa de dezvoltare a dus la bun sfârșit implementarea soluției. Prin intermediul algoritmului de generare identificat ultima dată s-a putut genera într-un mod facil și rapid scorul mediu final în cazul variantei de binarizare globală. Pentru abordarea variantei de binarizare locală, au fost necesare ușoare modificări la modalitatea stocării datelor, restul raționamentului bazându-se tot pe generarea arborilor de la pasul de binarizare globală, interpretarea valorii lor finale nemaivând acum un scop în calcularea unui prag final, ci a determinării clasei unui pixel. De asemenea, au fost validate de către echipa de testare atât metodele folosite în cadrul procesării datelor cât și cele pentru furnizarea secvenței de operații și calcularea valorii rădăcinii arborelui.

## Impactul metodologiei de dezvoltare folosite:

Noi am folosit o metodologie de tip Agile Hibridă. Suntem împărțiți în rolurile de Team Lead, Project Manager, Developeri și tester. Fiecare și-a respectat rolul și atribuțiile, ceea ce a eficientizat mult modul de lucru.

Avantajele pe care noi le-am sesizat au fost:

- *Sprint-urile* au fost utile pentru a putea împărți mai bine munca echipei și task-urile
- *Întâlnirile fizice de la laborator* care au înlocuit weekly-urile și au ajutat în stabilirea unor deadline-uri potrivite cu programul fiecăruia dintre noi
- Cu toate că nu am avut daily-uri, *Whatsapp* ne-a asigurat o comunicare rapidă și o modalitate de a oferi feedback constant și a fi la curent cu toată evoluția proiectului
- Fiecare membru din echipă a avut rolul bine stabilit, motiv pentru care ne-a fost mai ușor să ne organizăm, fiecare știind ce are de făcut

Astfel, am beneficiat de principalele beneficii ale Agile, anume livrare continuă și de calitate în timp scurt și de o colaborare constată și de nivel înalt.

În ultima parte a proiectului am continuat să utilizăm aceeași metodologie, având în vedere succesul avut anterior. Viteza de lucru a fost la fel de bună, iar dezvoltarea produsului a fost la fel de calitativă datorită numărului mare de verificări făcute asupra codului de către tester-ul nostru. Cooperarea în cadrul echipei a fost la fel de bine gestionată ca în celelalte două sprint-uri.

## Monitorizarea, evaluarea și controlul evoluției proiectului

Controlul evoluției proiectului a fost realizat cu ajutorul aplicației Trello unde au fost adăugate task-urile și deadline-urile pentru proiect astfel încât fiecare membru să fie la curent cu ceea ce are de făcut. Trello este un tool de tip Kanban Board pe care l-am folosit pentru împărțirea acestora, coloanele considerate necesare de noi fiind *TODO*, *Doing* și *Done*. Pentru versionare am utilizat Github, unde am folosit branch-uri separate pentru development și pentru testare. De asemenea, am folosit GanttProject pentru a putea ști o estimare de timp calculată și pentru task-urile organizatorice și pentru a compara mai ușor progresul proiectului cu ce era estimat la început. Pentru evaluarea progresului am avut întâlniri specifice și anume sprint review și planning.

## Plan de testare și rezultate

Testarea a fost făcută automat cu ajutorul framework-ului de testare GoogleTest. Au fost testate, pe bucăți, funcționalitățile implementate pentru a decide dacă fiecare parte componentă a soluției este corectă. Exemple de funcționalități testate: parsarea corectă și salvarea informațiilor din fișierele de input, operațiile folosite de arborele de calcul, etc. În urmă testării rezultatele au fost pozitive, deci putem spune că modulele și funcțiile care alcătuiesc soluția problemei au fost implementate corect și s-a putut prezice după aceea cel mai bun scor. Mai jos puteți regăsi un exemplu de rulare a testelor:

```
andrei@DESKTOP-011A7VJ:/mnt/e/Facultate Andrei/Anul 4/Sem 1/WPS/Project/WPS/build$ ./WPS_Project
test
[*****] Running 7 tests from 3 test suites.
[*****] Global test environment set-up.
[*****] 3 tests from TestParser
[ RUN ] TestParser.setFilePath_should_be_true (0 ms)
[ OK ] TestParser.setFilePath_should_be_true (0 ms)
[ RUN ] TestParser.parseFile_should_be_correct_when_file_exists
[ OK ] TestParser.parseFile_should_be_correct_when_file_exists (1 ms)
[ RUN ] TestParser.parseFile_should_fail_when_file_missing
[ OK ] TestParser.parseFile_should_fail_when_file_missing (0 ms)
[*****] 3 tests from TestParser (2 ms total)

[*****] 3 tests from TestOperation
[ RUN ] TestOperation.correct_answer_geometric_mean
[ OK ] TestOperation.correct_answer_geometric_mean (0 ms)
[ RUN ] TestOperation.correct_answer_arithmetic_mean
[ OK ] TestOperation.correct_answer_arithmetic_mean (0 ms)
[ RUN ] TestOperation.correct_answer_multiplication
[ OK ] TestOperation.correct_answer_multiplication (0 ms)
[*****] 3 tests from TestOperation (0 ms total)

[*****] 1 test from TestRandomTree
[ RUN ] TestRandomTree.correct_random_tree_generation_links
[ OK ] TestRandomTree.correct_random_tree_generation_links (0 ms)
[*****] 1 test from TestRandomTree (0 ms total)

[*****] Global test environment tear-down
[*****] 7 tests from 3 test suites ran. (2 ms total)
[ PASSED ] 7 tests.
andrei@DESKTOP-011A7VJ:/mnt/e/Facultate Andrei/Anul 4/Sem 1/WPS/Project/WPS/build$
```

Sau output-ul direct:

[=====] Running 7 tests from 3 test suites.

[-----] Global test environment set-up.

[-----] 3 tests from TestParser

[ RUN ] TestParser.setFilePath\_should\_be\_true

[ OK ] TestParser.setFilePath\_should\_be\_true (0 ms)

[ RUN ] TestParser.parseFile\_should\_be\_correct\_when\_file\_exists

[ OK ] TestParser.parseFile\_should\_be\_correct\_when\_file\_exists (1 ms)

[ RUN ] TestParser.parseFile\_should\_fail\_when\_file\_missing

[ OK ] TestParser.parseFile\_should\_fail\_when\_file\_missing (0 ms)

[-----] 3 tests from TestParser (2 ms total)

[-----] 3 tests from TestOperation

[ RUN ] TestOperation.correct\_answer\_geometric\_mean

[ OK ] TestOperation.correct\_answer\_geometric\_mean (0 ms)

[ RUN ] TestOperation.correct\_answer\_arithmetic\_mean

[ OK ] TestOperation.correct\_answer\_arithmetic\_mean (0 ms)

[ RUN ] TestOperation.correct\_answer\_multiplication

[ OK ] TestOperation.correct\_answer\_multiplication (0 ms)

[-----] 3 tests from TestOperation (0 ms total)

[-----] 1 test from TestRandomTree

[ RUN ] TestRandomTree.correct\_random\_tree\_generation\_links

[ OK ] TestRandomTree.correct\_random\_tree\_generation\_links (0 ms)

[-----] 1 test from TestRandomTree (0 ms total)

[-----] Global test environment tear-down

[=====] 7 tests from 3 test suites ran. (2 ms total)

[ PASSED ] 7 tests.