

# Raport de Dezvoltare

## MPS - Milestone 2

Membrii echipei

Blacioti Mihai - Tester  
Brînză Maria-Cristina - Software Developer  
Dranca Ștefana-Ioana - Team Leader  
Dumitrescu Ioana-Teodora - Project Manager  
Lovin Cosmin-Viorel - Software Developer

### 1. Planul de risc al dezvoltării proiectului

În urma unui brainstorming la nivel de echipă și a unei analize amănunțite a task-ului și implementării alese spre dezvoltare într-un timp restrâns, am identificat o serie de riscuri ce pot îngreuna sau chiar sista producția. Pe unele dintre acestea am reușit să le eradicăm, pe altele să le diminuăm, dar încă există riscuri active asupra cărora acordăm o atenție sporită pentru a nu le lăsa să scape de sub control.

Am abordat în detaliu problemele apărute cu ajutorul planului de risc din link-ul următor: [Risk Management Plan MPS](#).

Considerăm că cele mai multe dificultăți apar din cauza complexității pe care am ales-o raportat la timpul de dezvoltare a aplicației, dar și a dorinței de a atinge o acuratețe ridicată pentru fiecare rulare a algoritmului.

### 2. Progres al echipei în dezvoltarea soluției

Până în acest moment am reușit implementarea unui prototip intermediar al aplicației software. Suntem într-o continuă încercare a îmbunătățirii algoritmului, optând pentru înlocuirea/retușarea bucatărilor de cod problematice. Cel mai bun exemplu în acest sens și cel mai clar de înțeles este augmentarea unor parcurgeri secvențiale prin paralelizare. Odată cu introducerea noului set de date, a devenit clar că o parcurgere naivă este un obstacol ce trebuie evitat, datorită creșterii sesizabile a timpului de execuție. Dată fiind dimensiunea resurselor, diferența făcută de utilizarea thread-urilor este sesizabilă, bottleneck-ul înlăturat permițând rularea accelerată a codului.

Având acest aspect în minte, câștigul riscului de implementare este acela că media threshold-urilor obținută la fiecare pas atinge o valoare adecvată, sporind astfel acuratețea. Desigur, scenariul prezentat anterior nu este unul constant, însă după executarea repetată a programului se poate observa o înclinare spre amplificarea valorilor.

### 3. Impactul metodologiei de dezvoltare folosite

Faptul că ne-am folosit de metodologia de dezvoltare Agile, varianta Scrum ne-a ajutat extrem de mult în asignarea în mod egal din punct de vedere al volumului de muncă și al timpului necesar soluționării task-ului. Astfel, task-urile au fost bine stabilite

de la bun început, fiecare având un overview al situației generale a proiectului, dar cunoscând în același timp și statusul curent al celorlalți membri ai echipei. Totuși, am pus accent și pe lucrul în echipă, oferind sau cerând ajutor pentru diverse task-uri.

Funcționalitatea de backlog a platformei Jira pe care o folosim ne-a ajutat să stabilim o cursivitate a etapelor de implementare, să putem împărți timpul total mai eficient prin cunoașterea tuturor task-urilor până la final.

La ședințele de planificare a sprintului am estimat task-urile corespunzătoare acestuia pentru a le atribui membrilor echipei în funcție de productivitatea deja cunoscută a acestora. De asemenea, am învățat să ne adaptăm mai bine volumul de muncă per sprint.

Prin retrospectivă, am evaluat propria performanță, am trecut în revista cunoștințele noi dobândite și am oferit feedback pentru a ne afla într-un constant progres organizatoric.

#### 4. Monitorizare, evaluare și controlul evoluției proiectului

##### **MONITORIZARE**

Pe lângă întâlnirile de la laborator, am organizat o serie de ședințe de verificare a progresului, unde fiecare a prezentat statusul task-urilor asignate, blocajele întâmpinate și rezoluțiile de viitor. În plus, am încercat să îmbunătățim în mod constant abordarea aleasă pentru soluționarea fiecărei sarcini, acest fapt ducând de obicei la o dezbatere.

Conflictele în cadrul unui proiect sunt inevitabile, însă asta nu înseamnă că blocajele produse de acestea nu pot fi gestionate. De obicei, consensul era obținut ușor, chiar dacă fiecare membru al echipei a avut un motiv anume pentru orice decizie și o justificare pe măsură. În momentele în care discuția nu avansa către o concluzie am recurs la utilizarea tacticilor de rezolvare a conflictelor. Strategia cel mai des folosită a fost cea de negociere, fiind susținută de tot colectivul. Pentru îmbunătățirea funcționalității metodei, am apelat la o listă Pro/Contra, gestionând fiecare item cu o pondere a importanței. Urmărind acest set de reguli, conflictele erau soluționate ușor. Cu toate acestea, în condițiile în care eșuam, se opta mai întâi pentru ocolire, ulterior abordând direct problema, dreptul de veto revenind proiect managerului.

##### **EVALUARE**

Evaluarea performanțelor membrilor echipei s-a realizat de către proiect manager, întrucât acesta a interacționat cel mai mult cu părțile implicate. În urma întocmirii documentului: [Evaluation Form - Milestone 2](#), fiecare a avut o discuție cu managerul pentru a-și putea înțelege și depăși greșelile.

##### **CONTROLUL EVOLUTIEI**

Una dintre metricile pe care le-am folosit pentru evaluarea eficienței proiectului

în fază intermediară este *SPI (Schedule Performance Index)*:

EV ("Cât de mult s-a realizat până în acest punct din ce s-a planificat?")

- Proiectare soluție intermediară 100%
- Paralelizare soluție 70%
- Obținere rezultate precise 90%
- Redactare Rapoarte 100%

$SPI = (100 + 70 + 90 + 100) / (4 * 100) = 360 / 400 = 0.9$  =: proiectul este ușor în întârziere