

Raport de dezvoltare

Membrii echipei

- Project Manager - David Capragiu
- Team Lead - Ștefăniță Ioniță
- Software developers
 - Bogdan Brutaru
 - Florian Velea
 - Maria Niță
- Tester - Alexandru Măciucă

1. Planul de risc al dezvoltării proiectului

În urma unui brainstorming la nivel de echipă și al unei examinări aprofundate a sarcinilor și implementărilor care au fost alese pentru dezvoltare într-un interval de timp limitat, am descoperit o serie de amenințări care ar putea afecta sau chiar opri producția. Pe unele dintre acestea le-am eliminat, pe altele le-am redus, dar încă există riscuri active asupra cărora trebuie să acordăm mai multă atenție pentru a nu le lăsa să scape de sub control.

Planul de management al riscurilor MPS poate fi găsit aici pentru a afla mai multe despre problemele care au apărut.

Considerăm că majoritatea problemelor apar din complexitatea pe care am ales-o în timpul procesului de dezvoltare a aplicației, precum și din dorința de a atinge o acuratețe ridicată pentru fiecare rulare a algoritmului ([plan de risc aici](#)).

2. Progresul echipei în dezvoltarea soluției propuse

Până acum, am reușit să implementăm prototipul intermediar al aplicației software. În timp ce alegem să înlocuim sau să retușăm bucățile de cod care prezintă probleme, continuăm să îmbunătățim algoritmul. Augmentarea unor parcurgeri secvențiale prin paralelizare este cel mai bun exemplu în acest sens și cel mai ușor de înțeles. Datorită creșterii notabile a timpului de execuție, a devenit evident că o parcurgere naivă trebuie evitată. Deoarece resursele sunt mari, diferența dintre utilizarea țesăturilor este evidentă. Lipsa bottleneck-ului permite codului să funcționeze mai repede.

Având în vedere acest aspect, câștigul riscului de implementare este că acuratețea crește ca urmare a mediei pragurilor obținute la fiecare pas. Chiar dacă scenariul de mai sus nu este o situație constantă, se poate observa o creștere a valorilor după ce aplicația este executată din nou și din nou.

3. Impactul metodologiei scrum în cadrul echipei

În ceea ce privește volumul de muncă și timpul necesar pentru finalizarea sarcinilor, utilizarea metodei de dezvoltare Agile (variante Scrum) a fost foarte benefică pentru noi. Prin urmare, sarcinile au fost bine organizate de la început, fiecare având o înțelegere a situației generale a proiectului, dar cunoscând în același timp situația actuală a celorlalți membri ai echipei. Totuși, am pus accent și pe lucrul în echipă, oferind sau cerând ajutor pentru o varietate de activități.

Funcționalitatea de backlog a platformei Azure Devops pe care o folosim ne-a ajutat să stabilim o cursivitate a etapelor de implementare și să putem împărți timpul total mai eficient prin cunoașterea tuturor sarcinilor până la finalizare.

În timpul ședințele de planificare a sprintului, am calculat sarcinile corespunzătoare pentru a le atribui membrilor echipei în funcție de productivitatea lor anterioară. Am descoperit, de asemenea, cum să ne adaptăm mai bine volumul de muncă pentru fiecare sprint.

Am făcut o analiză a propriei performanțe, am revizuit ceea ce am învățat recent și am oferit critici pentru a ne ajuta să continuăm să progresăm ca organizație.

4. Monitorizarea activităților și evaluarea acestora

a. monitorizarea

Am organizat o serie de ședințe de verificare a progresului. În aceste ședințe, fiecare a discutat despre statusul sarcinilor pe care le-au primit, problemele pe care le-au întâmpinat și soluțiile pe care le putem lua în viitor. În plus, am încercat să îmbunătățim în mod constant metoda pe care am ales-o pentru a rezolva fiecare sarcină, ceea ce de obicei duce la o dezbatere.

Cu toate că conflictele în timpul unui proiect sunt inevitabile, acest lucru nu înseamnă că blocajele care rezultă din aceste conflicte nu pot fi gestionate. Chiar dacă fiecare membru al echipei a avut o justificare pe măsură pentru orice decizie, de obicei era ușor să se ajungă la un consens. Am folosit strategii de rezolvare a conflictelor atunci când discuția nu a ajuns la o concluzie. Negocierea, care a fost susținută de tot colectivul, a fost strategia cea mai răspândită. Am folosit o listă Pro/Contra cu o pondere an importanței pentru a îmbunătăți funcționalitatea metodei. Problemele au fost rezolvate ușor dacă s-au respectat aceste reguli. Cu toate acestea, în situațiile în care am eșuat, s-a ales mai întâi o ocolire, apoi s-a abordat direct problema, cu proiect managerul având dreptul deciziei finale.

b. evaluarea

Evaluarea performanțelor echipei a fost realizată de către managerul de proiect. Pentru aceasta ne-am folosit de funcția de retrospective din Azure Devops pentru a putea să specificăm ce a mers bine și ce a mers rău în iterația curentă și cum ne-am simțit pe parcursul acesteia. Evaluarea pentru primul milestone poate fi găsită [aici](#). Totodată, putem găsi și evaluarea pentru al doilea milestone de implementare [aici](#).

În final, colaborarea între membri echipei a fost per total eficientă cu toate că în continuare au existat probleme de sincronizare. Chiar și așa, am reușit să ne oferim feedback pentru fiecare implementare adăugată și acestea au fost și analizate de cel puțin un coleg, astfel fiind foarte ușor să putem pasa sarcini de la unul la celălalt.

5. Controlul evoluției

Pentru a evalua eficiența proiectului în faza intermediară, am folosit Schedule Performance Index.

- Proiectarea unui prototip funcțional - 100%
- Obținere rezultate precise
 - 95% F Measure mediu
- Redactare rapoarte - 100%
- Posibilitatea de paralelizare a soluției - 76%

$$SPI = (100 + 95 + 100 + 76) / 400 = 360 / 400 = 0.9275$$

Din acest calcul putem observa că suntem aproximativ la timp cu livrarea proiectului.