

## I. Proiect 1

În acest proiect echipa va dezvolta **parțial** o aplicație (cerințele sunt la alegerea echipei) folosind **metodologia SCRUM** [1,2]. Dezvoltarea va consta dintr-un singur Sprint, având 5 zile lucrătoare (40 de ore). Se recomandă ca aceste 5 zile să fie distribuite de-a lungul a 5 săptămâni (o zi / săptămână). **Accentul în acest proiect pe este management și organizare, nu pe dezvoltare!**

**Roluri:** Unul dintre membrii echipei își va asuma rolul de Product Owner, un altul își va asuma rolul de Scrum Master, iar ceilalți vor fi Dezvoltatori. În echipă nu vor exista Testeri dedicați, generarea de teste fiind inclusă în procesul de implementare, iar testarea de acceptanță va fi făcută încrucișat (de către un alt Dezvoltator decât cel care a realizat implementarea).

### Cerințe:

1. Se pornește de la un scenariu propriu care descrie cerințele sistemului ce urmează a fi dezvoltat (Requirements Document). Acesta poate cuprinde atât cerințe funcționale cât și cerințe nefuncționale și alte constrângeri. Cerințele sistemului vor fi alese astfel încât din acest document să rezulte cel puțin 15 User Stories [3]. (Responsabilitate: Product Owner)
2. Din Requirements Document vor fi derivate User Stories inițiale. (Responsabilitate: Product Owner)
3. User Stories de la punctul anterior vor fi rafinate astfel încât să satisfacă cele 6 criterii: Independent, Negotiable, Valuable to users or customers, Estimable, Small, Testable (INVEST) [3]. Să se justifice modificările făcute. Fiecărei User Story i se va atribui o prioritate, care va fi justificată. User Stories rezultate vor alcătui Project Backlog. (Responsabilitate: Product Owner și Dezvoltatorii)
4. Să se estimeze efortul (în story points) pentru fiecare User Story folosind Planning Poker [4] și să se justifice estimarea. (Responsabilitate: Dezvoltatori)
5. Se completează Sprint Backlog în Planning Meeting (care are loc în prima zi a Sprint-ului). Observație: În mod normal, **Sprint Backlog nu va cuprinde toate User Stories din Project Backlog**; din contră, având în vedere durata redusă a sprint-ului și dimensiunea echipei, este de așteptat ca doar o parte din ele să poată fi incluse în Sprint Backlog, selecția fiind făcută în funcție de estimarea efortului și priorității. Pentru managementul proiectului se va folosi un tool precum cele de la [5] sau [6]. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner)
6. Să se implementeze User Stories (doar cele din Sprint Backlog!) folosind tehnica **Pair Programming** [7]. Pentru fiecare User Story, dezvoltatorul va crea teste automate folosind un utilitar adecvat (e.g. JUnit pentru limbajul Java). Observație: limbajul de programare este ales de echipă astfel încât să fie adecvat sistemului dezvoltat și cerințelor proiectului. (Responsabilitate: Dezvoltatori)
7. Să se utilizeze un sistem de control al versiunilor, de exemplu SVN (Subversion), CVS (Concurrent Version System), GIT etc. Să se folosească o platformă (cu găzduire free) pentru repository-ul proiectului dezvoltat, să se instaleze și utilizeze diferiți clienți pentru sistemul de control al versiunilor ales (de exemplu SmartSVN, TortoiseSVN, Subclipse pentru Subversion). (Responsabilitate: Dezvoltatori, Scrum Master)
8. Sprint Daily Standup Meetings vor avea loc în fiecare zi din Sprint, iar raportul prezentat la sfârșitul proiectului va consemna principalele impedimente apărute. (Responsabilitate: întreaga echipă)
9. Implementarea (fiecărei User Story în parte) va fi prezentată (și validată/ invalidată de către Product Owner) în Review Meeting, care va avea loc în ultima zi a Sprint-ului. Tot în ultima zi va avea loc și o foarte scurtă (maxim 20 minute) Retrospective Meeting. (Responsabilitate: întreaga echipă, facilitată de Scrum Master)

10. Prezențați în cadrul laboratorului dezvoltarea aplicației în echipă (includeți un link la repository-ul configurat și user pentru acces), testarea aplicației, menținerea unei liste de bug-uri, asignarea lor membrilor echipei (pentru aceasta folosiți un bug tracking tool, de exemplu Bugzilla) (Responsabilitate: întreaga echipă)
11. Raportul scris al proiectului (însotit de slide-uri, care vor puncta ideile principale) va fi prezentat în cadrul cursului. Acesta va răspunde la punctele 1-9 (dar nu va cuprinde cod). În plus, raportul va evidenția avantajele și dezavantajele (greutățile întâmpinate în aplicarea) metodologiei de dezvoltare folosite în cadrul proiectului. (Responsabilitate: întreaga echipă)
12. Să se precizeze (ca procentaj, la sfârșitul raportului) contribuția fiecărui membru al echipei la realizarea proiectului.

## II. Proiect 2

În fiecare echipă se vor forma două grupuri: grupul A, care cuprinde dezvoltatorii din Proiectul 1 și grupul B, care cuprinde Product Owner și Scrum Master. Se vor alege 5 User Stories dintre cele selectate în Sprint Backlog. Fiecare dintre aceste User Stories va fi dezvoltată în mod tradițional de toți studenții din grupul A (refolosind implementarea din Proiectul 1 acolo unde este cazul) și folosind metoda Test-Driven Development (TDD) [8] de studenții din grupul B. (În ambele cazuri se recomandă folosirea Pair Programming; totuși, dacă nu folosește Pair Programming pentru un mod de dezvoltare, pentru consistență nu se va folosi nici pentru celălalt). Pentru fiecare implementare se va stabili nivelul de acoperire realizat de seturile de teste rezultate folosind un utilitar de code coverage precum cele prezentate în [9].

Raportul scris al proiectului (însotit de slide-uri, care vor puncta ideile principale) va fi prezentat în cadrul cursului. Acesta va compara dezvoltarea tradițională cu TDD urmărind următoarele aspecte:

Aspecte cantitative:

- Calitatea codului (măsurată prin code coverage)
- Timpul necesar dezvoltării

Aspecte calitative:

- Cât de productivă este practica TDD pentru programatori?
- Cât de eficientă este practica TDD?
- Cât de dificil de adoptat este practica TDD?

## Bibliografie

1. Florentin Ipat: Scrum, note de curs.
2. <https://theagiledirector.com/images/IntroductiontoScrum-coursenotes.pdf>
3. Mike Cohn, User Stories Applied, Addison-Wesley, 2004 (Chapter 2). <http://athena.ecs.csus.edu/~buckley/CSc191/User-Stories-Applied-Mike-Cohn.pdf>
4. <https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/>
5. <http://www.rallydev.com/>
6. <https://www.collab.net/>
7. <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/recursos/XPSardinia.pdf>
8. <https://books.google.ro/books?hl=en&lr=&id=CUlsAQAAQBAJ&oi=fnd&pg=PR7&dq=test+driven+development&ots=QChW467PRX&sig=CD->

[L\\_nqaKVpL8lvXtKCjt7z-](#)

[D\\_c&redir\\_esc=y#v=onepage&q=test%20driven%20development&f=false](#)

9. <https://www.softwaretestinghelp.com/code-coverage-tools/>