Proiect de laborator MOPS

Dimensiunea recomandată a echipei pentru acest proiect **este de 4-5 studenți**. În acest proiect echipa va dezvolta **parțial** o aplicație (cerințele sunt la alegerea echipei) folosind **metodologia SCRUM** [1,21]. Dezvoltarea va consta dintr-un singur Sprint, având 5 zile lucrătoare (40 de ore). Se recomandă ca aceste 5 zile sa fie distribuite de-a lungul a 5 săptămâni (o zi / săptămână).

Proiectul trebuie să fie complex și inovativ. Cadrul didactic de la laborator vă va ghida în alegerea proiectului.

Roluri: Unul dintre membrii echipei își va asuma rolul de Product Owner, ceilalți fiind Dezvoltatori. Dintre Dezvoltatori, unul își va asuma simultan și rolul de Scrum Master. În echipă nu vor exista Testeri dedicați, testarea unitară 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 va fi însoțit de o Diagramă de Cazuri de Utilizare (Use Case Diagram) [1,25,26]. (Responsabilitate: Product Owner)
2. Din Requirements Document vor fi derivate User Stories inițiale [1-5]. (Notă: cerințele sistemului vor fi alese astfel încât să rezulte cel puţin 15 User Stories.) (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) [1-3]. 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 [7,8] și să se justifice estimarea. Se va demonstra utilizarea Planning Poker. (Responsabilitate: Dezvoltatori)
5. Se completeaza Sprint Backlog in Planning Meeting (care are loc in prima zi a Spint-ului). Observatie: 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 priorități. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner).
6. Se va folosi un tool (la alegerea echipei) pentru project management și issue tracking [5,12]. La prezentare, se va dovedi ca s-a utilizat tool-ul respectiv. (Responsabilitate: Dezvoltatori, Scrum Master, Product Owner).
7. Scrum Master-ul, cu ajutorul dezvoltatorilor, va realiza urmatoarele diagrame (aceste diagrame vor fi incluse în Project Backlog):
    a. Diagrama de workflow a proiectului.
    b. Diagrama bazei de date (dacă este cazul).
8. Product Owner-ul, cu ajutorul întregii echipe va dezvolta urmatoarele diagrame (aceste diagrame vor fi incluse în Project Backlog):
    a. Diagrama UML de cazuri de utilizare.
    b. Diagrama Gantt

9. Să se implementeze User Stories:
    a. aproximativ jumătate dintre acestea folosind metoda **Test-Driven Developmen**t [1,19].
    b. iar restul in mod traditional, însă folosind tehnica **Pair Programming** [22,23].
    c. Să se descrie procesul şi evoluţia testelor.
    d. Să se scrie cazurile de testare rezultate folosind un utilitar adecvat (e.g. JUnit pentru limbajul Java [17]).
    e. Să se stabilească nivelul de acoperire realizat de fiecare dintre seturile de teste folosind un utilitar de code coverage precum cele prezentate in [20].
    f. Să se comenteze avantajele si dezavantajele Test-Driven Development față de dezvoltarea tradițională.
    g. Să se comenteze avantajele si dezavantajele Pair Programming față de programarea individuală.

    (Notă: limbajul de programare este ales de echipa astfel încât să fie adecvat sistemului dezvoltat şi cerinţelor proiectului. Responsabilitate: Dezvoltatori).
10. Să se utilizeze un sistem de control al versiunilor, de exemplu SVN (Subversion), CVS (Concurrent Version System), GIT etc [13,14]. Să se folosească o platformă (cu găzduire free) pentru repository-ul proiectului dezvoltat, să se instaleze şi utilizeze diferiți clienți pentru sistemul de control al versiunilor ales (de exemplu SmartSVN, TurtoiseSVN, Subclipse pentru Subversion). (Responsabilitate: Dezvoltatori, Scrum Master)
11. Sprint Daily Standup Meetings vor avea loc în fiecare zi din Sprint, iar raportul proiectului va consemna principalele impedimente apărute. (Responsabilitate: întreaga echipa)
12. Implementarea (fiecărei User Story în parte) va fi prezentată (și validată/ invalidată de către Product Owner) în Review Meeting, care va avea loc in ultima zi a Sprint-ului. Tot în ultima zi va avea loc și o foarte scurtă (maxim 20 minute) Retrospective Meeting. (Responsabilitate: întreaga echipa, facilitată de Scrum Master)
13. Pentru comunicare intre membrii echipei se va folosi Slack [24]. Se va dovedi utilizarea la prezentare.
14. Să se precizeze (ca procentaj, la sfârșitul raportului) contribuția fiecărui membru al echipei la realizarea proiectului.

Evaluare

Evaluarea acestui proiect se va face la laborator: cadrul didactic de laborator va stabili obiectivele de etapă (**project milestones)**, evaluarea fiind făcută la sfârșitul fiecărei etape.

Termene:

- **I.** până în săptămâna 4: ați fixat echipa și cerințele/backlogul proiectului (cadrul didactic de laborator își dă acordul)
- **II.** până în săptămâna 8: ați implementat cel putin 1/3 din backlog (cadrul didactic de laborator verifică și renegociați schimbări ale backlogului, dacă e cazul)
- **III.** până în săptămâna 12: ați implementat cel puțin 2/3 din backlog (cadrul didactic de laborator verifică și renegociați schimbări ale backlogului, dacă e cazul)
- **IV.** săptămânile 13, 14 și o data aleasă din sesiune: prezentați întreg proiectul (cadrul didactic de la laborator verifică dacă a fost implementat tot backlogul) și primiți nota pe proiect. Prezentarea va dura 20 de minute, iar timpul va fi alocat astfel: o prezentare (5 minute) cu slide-uri a proiectului care va cuprinde cerințele atinse; o prezentare live (10 minute) a proiectului; iar 5 minute vor fi alocate pentru întrebări.

Pentru prezentarea finală a proiectului până în săptămâna 14 (inclusiv), primiți 1 pct în plus.

Pentru fiecare checkpoint care nu este prezentat satisfăcător, pierdeți 0,75 pct din nota finală la laborator.

Prezența individuală la laborator: daca ați participat la mai puțin de 4 laboratoare din cele 7, pierdeți 2 pct din cele de mai sus. Pentru fiecare checkpoint prezentat înainte de termen, primiți 0.25 pct în plus.

Testarea proiectului valorează 1,5 puncte și se va face dovada testării prin teste automate.

Complexitatea proiectului valorează 2 puncte, iar cadrul didactic de laborator stabilește dacă proiectul este suficient de complex pentru a primi integral cele 2 puncte.

În cadrul prezentării finale, cea cu slide-uri, se va puncta fiecare cerință atinsă din cele 14 de mai sus. Rămîne la latitudinea cadrului didactic de laborator dacă el consideră că ați atins cu succes toate subpunctele 1-14.

Bibliografie

1. Florentin Ipate: note de curs (Scrum, XP, TDD, Use Case Model).
2. Mike Cohn, User Stories Applied, Addison-Wesley (Chapter 2), http://athena.ecs.csus.edu/~buckley/CSc191/User-Stories-Applied-Mike-Cohn.pdf
3. https://www.mountaingoatsoftware.com/uploads/presentations/User-Stories-Applied-Agile-Software-Development-XP-Atlanta-2004.pdf
4. https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/
5. https://www.atlassian.com/agile/project-management/user-stories
6. https://www.adaptavist.com/blog/how-to-create-an-effective-jira-user-story
7. https://www.7pace.com/blog/user-story-estimation-techniques
8. https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/
9. https://www.sciencedirect.com/science/article/pii/S0164121212001021
10. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0ae75cc0678f6b01f1066093b19cad9dd4379385
11. https://is.muni.cz/th/pbpa1/thesis_gyyik.pdf
12. https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems
13. https://www.sciencedirect.com/science/article/pii/S1877050918314819
14. https://www.atlassian.com/git/tutorials/what-is-version-control
15. https://arxiv.org/abs/2004.10777
16. https://refactoring.guru/refactoring/techniques
17. https://www.browserstack.com/guide/junit-and-automation-testing
18. https://www.jetbrains.com/help/idea/tutorial-introduction-to-refactoring.html#f6c540f0
19. 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
20. https://www.softwaretestinghelp.com/code-coverage-tools/
21. https://theagiledirector.com/images/IntroductiontoScrum-coursenotes.pdf
22. http://www.dsc.ufcg.edu.br/~jacques/cursos/map/recursos/XPSardinia.pdf

23. https://files.spiderboy.fr/madchat/coding/xp/xpexplored.pdf
24. https://slack.com/intl/en-gb/
25. Martin Fowler: UML Distilled: A Brief Guide to the Standard Object Modeling Language.
26. https://csis.pace.edu/~marchese/CS389/L9/Use%20Case%20Diagrams.pdf