Se porneşte de la un scenariu propriu care descrie cerinţele sistemului ce urmează a fi dezvoltat (Requirements Document). Acesta poate cuprinde atât cerinţe funcţionale cât şi cerinţe nefuncţionale şi alte constângeri. Cerinţele sistemului vor fi alese astfel încât din acest document sa rezulte cel puţin 15 User Stories. – 50%

Din Requirements Document vor fi derivate User Stories iniţiale. – 50%

User Stories de la punctual anterior vor fi rafinate astfel încât să satisfacă cele 6 criterii: Independent, Negotiable, Valuable to users or customers, Estimatable, Small, Testable. Să se justifice modificările făcute. Pentru fiecare User Story să se scrie cazuri de testare iniţiale. Fiecărei User Story i se va atribui o prioritate, care va fi justificată – 0%

Să se estimeze efortul (în story points) pentru fiecare user story folosind Planning Poker şi să se justifice estimarea. – 0%

Să se implementeze 3 User Stories folosind practica XP Test First Programming. (Nota: limbajul de programare este ales de echipa astfel încât să fie adecvat sistemului dezvoltat şi cerinţelor proiectului). Să se descrie procesul şi evoluţia testelor. Sa se scrie cazurile de testare rezultate folosind un utilitar adecvat (e.g. JUnit pentru limbajul Java). Să se arate dacă şi în ce mod metodele de testare tradiţionale (funcţionale şi structurale) au fost utilizate în procesul de selectare a cazurilor de testare. 0%

Să se investigheze în ce măsură folosirea unui model (e.g. de tipul State Machines) este util în selectarea automată a cazurilor de testare pentru cele 3 User Stories de la punctual anterior. În cazul unui răspuns afirmativ, sî se indice strategia de selectare a cazurilor de testare. Pentru fiecare dintre cele 3 User Stories, să se scrie (folosind acelaşi utilitar ca mai sus) cazurile de testare rezultate. Să se discute în ce măsură folosirea unui model poate fi integrată într-o metodologie de dezvoltare agilă. 0%

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 şi utilizeze diferiţi clienţi pentru sistemul de control al versiunilor ales (de exemplu SmartSVN, TurtoiseSVN, Subclipse pentru Subversion). 100%

Prezentaţi în cadrul laboratorului dezvoltarea aplicaţiei în echipă (includeţi un link la repository-ul configurat şi user pentru acces), testarea aplicaţiei, menţinerea unei liste de bug-uri, asignarea lor membrilor echipei. 0%

Să se evidenţieze avantajele şi dezavantajele (greutăţile întâmpinate în aplicarea) metodologiei de dezvoltare folosite în cadrul proiectului.0%

Să se precizeze (ca procentaj) contribuţia fiecărui membru al echipei la realizarea proiectului. 0%