Conţinut

1. Inspectarea cerinţelor (Requirements Inspection)..... 1
2. Inspectarea arhitecturii (Architectural Inspection) 1
3. Inspectarea codului sursă (Source Code Inspection)..... 2
4. Evaluarea dinamică a calităţii codului sursă (Dyanmic Code Analysis) 2

1. Inspectarea cerinţelor (Requirements Inspection)

1. Se parcurge documentul cu cerinţe (enunţul problemei), e.g., **RequirementsTasks_v1.0.pdf**.
2. Pentru fiecare element **R01..R07** din documentul **Lab01_RequirementsPhaseDefects.pdf** se caută în enunţul problemei dacă elementul **Rxx** este îndeplinit sau nu;
3. În fişierul **Lab01_ReviewReport.xlsx**, în worksheet-ul corespunzător inspectării cerinţelor, se completează:
    - pentru fiecare element **Rxx** cel puţin câte o observaţie proprie (e.g., **Rxx** apare în enunţul problemei dar nu este adecvat, **Rxx** nu apare în enunţul problemei deşi este potrivit să fie inclus în enunţ);
    - timpul necesar pentru realizarea inspectării cerinţelor;
4. Se modifică enunţul iniţial, astfel încât să reflecte observaţiile incluse în raportul de inspectare a cerinţelor, i.e., **RequirementsTasks_v2.0.pdf**.

2. Inspectarea arhitecturii (Architectural Inspection)

1. Se identifică arhitectura aplicaţiei (i.e., clasele, pachetele, dependenţele logice între clasele aplicaţiei, etc.), e.g., proiectul **Tasks**;
2. Se verifică corespondenţa logică dintre elementele de arhitectură (i.e., clase, relaţii între clase, atribute, metode, pachete, etc.) şi diagrama de clase supusă inspectării, i.e., **DiagramTasks_v1.0.pdf**;
3. Pentru fiecare element **A01..A10** din documentul **Lab01_ArchitecturalDesignPhaseDefectsChecklist.pdf** se caută în arhitectura aplicaţiei şi în diagrama de clase dacă elementul **Axx** apare în arhitectură şi în diagrama de clase sau nu;
4. În fişierul **Lab01_ReviewReport.xlsx**, în worksheet-ul corespunzător inspectării arhitecturii, se completează:
    - pentru fiecare element **Axx** cel puţin câte o observaţie proprie (e.g., **Axx** apare în arhitectura aplicaţiei dar nu este adecvat, **Axx** nu apare în arhitectura aplicaţiei deşi este potrivit sa fie inclus);
    - timpul necesar pentru realizarea inspectării arhitecturii;
5. Se corectează arhitectura aplicaţiei în conformitate cu observaţiile incluse în raportul de inspectare a arhitecturii aplicaţiei, i.e., **DiagramTasks_v2.0.pdf**.
6. **Dacă este necesar, modificările la nivelul arhitecturii se propagă la nivelul documentului cu cerinţele aplicaţiei, i.e., RequirementsTasks_v2.0.pdf**.

3. Inspectarea codului sursă (Source Code Inspection)

1. Se identifică elementele de implementare prezente în codul sursă (i.e., obiecte şi/sau variabile şi tipul acestora, metodele şi semnăturile acestora, algoritmii implementaţi şi complexitatea acestora, instrucţiuni de programare (e.g., **for**, **if**, **while**, **repeat**, **switch**, etc.)), e.g., proiectul **Tasks**;
2. Pentru fiecare element **C01..C12** din documentul **Lab01_ProgramCodingPhaseDefectsChecklist.pdf** se stabileşte dacă elementul **Cxx** apare în codul sursă al aplicaţiei sau nu;
3. În fişierul **Lab01_ReviewReport.xlsx**, în worksheet-ul corespunzător inspectării codului sursă, se completează:
    - pentru fiecare element **Cxx** cel puţin câte o observaţie proprie (e.g., **Cxx** apare în codul sursă al aplicaţiei dar nu este adecvat şi/sau corect, **Cxx** nu apare în codul sursă al aplicaţiei deşi este potrivit să fie inclus);
    - timpul necesar pentru realizarea inspectării codului sursă;
4. Se corectează **codul sursă** în conformitate cu observaţiile incluse în raportul de inspectare a codului sursă.
5. **Dacă este necesar, modificările de la nivelul codului sursă se propagă la nivelul arhitecturii, actualizând diagrama de clase, i.e., DiagramTasks_v2.0.pdf. În continuare, dacă este necesar, actualizările de la nivelul arhitecturii se propagă la nivelul documentului cu cerinţele aplicaţiei, i.e., RequirementsTasks_v2.0.pdf.**

4. Evaluarea statică a calităţii codului sursă (Tool-based Code Analysis)

1. Se instalează plugin-ul **SonarLint** pentru IntelliJ IDEA (vezi **Tutorial SonarLint**).
2. Se realizează analiza statică a tuturor fişierelor de cod sursă din proiect şi se operează modificări asupra codului sursă conform sugestiilor **SonarLint**.
3. În fişierul **Lab01_ReviewReport.xlsx**, în worksheet-ul corespunzător analizei statice a codului sursă, se completează:
    - cel puţin 5 aspecte pentru care s-au sugerat modficări şi acestea s-au efectuat. Dacă nu se vor urma indicaţiile de modificare precizate, se va specifica motivul (un argument) pentru care codul va rămâne neschimbat;
    - timpul necesar pentru realizarea analizei statice asupra codului sursă.