

Conținut

1. Inspectarea cerințelor (Requirements Inspection)	1
2. Inspectarea arhitecturii (Architectural Inspection)	1
3. Inspectarea codului sursă (Source Code Inspection)	1

1. Inspectarea cerințelor (Requirements Inspection)

- Se parcurge documentul cu cerințe sau enunțul problemei, e.g., [Biblioteca.pdf](#).
- Pentru fiecare element **R01..R07** din documentul [Lab02_RequirementsPhaseDefects.pdf](#) se caută în enunțul problemei dacă elementul **Rxx** este îndeplinit sau nu;
- În fișierul [Lab02_ReviewReport.xlsx](#), în sheet-ul corespunzător inspectării cerințelor, se completează:
 - pentru fiecare element **Rxx** cel puțin câte o observație proprie (e.g., **Rxx** apare în enunțul problemei dar nu este adecvat, **Rxx** nu apare în enunțul problemei deși este potrivit să fie inclus în enunțul problemei);
 - timpul utilizat pentru realizarea inspectării cerințelor;
- Se corectează fișierul cu cerințe în conformitate cu observațiile incluse în raportul de inspectare a cerințelor.

2. Inspectarea arhitecturii (Architectural Inspection)

- Se identifică arhitectura aplicației (i.e., care sunt clasele și pachetele, care sunt dependențele logice între clasele aplicației, etc.), e.g., proiectul [Biblioteca](#);
- Se verifică corespondența dintre elementele de arhitectură (i.e., clase, pachete) și diagama de clase, e.g., [DiagramaBiblioteca.png](#);
- Pentru fiecare element **A01..A11** din documentul [Lab02_ArchitecturalDesignPhaseDefectsChecklist.pdf](#) se caută în arhitectura aplicației și în diagrama de clase dacă elementul **Axx** apare în arhitectură și în diagrama de clase sau nu;
- În fișierul [Lab02_ReviewReport.xlsx](#), în sheet-ul corespunzător inspectării arhitecturii, se completează:
 - pentru fiecare element **Axx** cel puțin câte o observație proprie (e.g., **Axx** apare în arhitectura aplicației și/sau în diagrama de clase dar nu este adecvat, **Axx** nu apare în arhitectura aplicației și/sau diagrama de clase deși este potrivit să fie inclus);
 - timpul utilizat pentru realizarea inspectării arhitecturii;
- Se corectează arhitectura aplicației și/sau diagrama de clase în conformitate cu observațiile incluse în raportul de inspectare a arhitecturii aplicației.
- Dacă este necesar, modificările la nivelul arhitecturii se propagă la nivelul documentului cu cerințele aplicației, e.g., [Biblioteca.pdf](#).

3. Inspectarea codului sursă (Source Code Inspection)

- Se identifică elementele de implementare prezente în codul sursă (i.e., obiecte și/sau variabile și tipul acestora, metodele și semnăturile acestora, algoritmi implementați și complexitatea acestora, structuri de programare (e.g., **for**, **if**, **while**, **repeat**, **switch**, etc.)), e.g., proiectul [Biblioteca](#);
- Pentru fiecare element **C01..C12** din documentul [Lab02_ProgramCodingPhaseDefectsChecklist.pdf](#) se stabilește dacă elementul **Cxx** apare în codul sursă al aplicației sau nu;

3. În fișierul [Lab02_ReviewReport.xlsx](#), în sheet-ul corespunzător inspectării codului sursă, se completează:
 - pentru fiecare element **Cxx** cel puțin câte o observație proprie (e.g., **Cxx** apare în codul sursă al aplicației dar nu este adecvat, **Cxx** nu apare în codul sursă al aplicației deși este potrivit să fie inclus);
 - timpul utilizat pentru realizarea inspectării codului sursă;
4. Se corectează codul sursă în conformitate cu observațiile 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, e.g., [DiagramaBiblioteca.png](#). În continuare, dacă este necesar, actualizările la nivelul arhitecturii se propagă la nivelul documentului cu cerințele aplicației, e.g., [Biblioteca.pdf](#).