

# CURS 01B.

# INSPECTARE

---

**Verificarea și validarea sistemelor soft**  
**[28 Februarie 2023]**

Lector dr. Camelia Chisăliță-Crețu  
Universitatea Babeș-Bolyai

# Conținut

- Calitatea produselor soft
  - Activități asociate controlului calității
  - Analiza statică. Clasificare
- Metode bazate pe factorul uman
  - Definiție. Motivație. Caracteristici
  - Inspectare Fagan
  - Walkthroughs
  - Technical Review
  - Pair-Programming
- Pentru examen...
- Bibliografie

# CALITATEA PRODUSELOR SOFT

---

Activități asociate controlului calității unui produs soft

Analiza statică. Clasificare

# Activități asociate calității

- *în procesul de dezvoltare, calitatea este abordată din perspectiva:*
  - **procesului** ==> **asigurarea calității** (*engl. quality assurance*):
    - **Obiectiv:** asigură respectarea standardelor, planurilor și etapelor proceselor de dezvoltare necesare elaborării adecvate a produsului cerut;
    - **Întrebare:** Cum se asigură calitatea activităților desfășurate în procesul dezvoltare?
  - **produsului** ==> **controlul calității** (*engl. quality control*):
    - **Obiectiv:** identifică deficiențele în produsul obținut;
    - **Întrebare:** Cum se controlează calitatea rezultatelor obținute (e.g., work products) în urma activităților desfășurate?



# Controlul calității. Activități asociate (1)

## Analiză statică (static testing)

- examinarea unor documente ( specificații, modele conceptuale, diagrame de clase, cod sursă, planuri de testare, documentații de utilizare);
- **exemple:** activități de inspectare a codului, analiza algoritmului, demonstrarea corectitudinii;
- se pot baza pe factorul uman (reviews) sau utilizarea tool-urilor (analiza statică).

## Analiză dinamică (dynamic testing)

- examinarea comportamentului programului cu scopul de a evidenția defecțiuni posibile;
- **exemple:** *tipuri de testare* (de regresie, funcțională, non-funcțională), *niveluri de testare* (testare unitară, testare de integrare, testare de sistem, testare funcțională, testare de acceptare);
- se bazează întotdeauna pe execuția programului.

# Controlul calității. Activități asociate (2)

## Analiză statică (static testing)

- **permit identificarea mai multor erori (greșeli) care pot fi corectate simultan;**
- **NU** presupune execuția propriu-zisă a programului dezvoltat;

## Analiză dinamică (dynamic testing)

- **sugerează doar un simptom**, fiecare eroare identificată fiind eliminată individual;
- **include activitatea de execuție propriu-zisă a programului (testare);**
- poate să evidențieze o defecțiune doar în anumite situații.

- metode de analiză complementare;
- dezvoltatorii aplică metode hibride, care folosesc avantajele celor două abordări.

# Analiză statică

- preconcepție (anii '60) –
  - „singura modalitate de a verificare a unui program este execuția pe calculator” [[Myers2004](#), Cap.3];
  - se presupunea că un program este scris doar pentru execuția de către calculator și nu este util și necesar să fie citit și înțeles de o persoană, e.g., programator, tester;
- metode de analiză statică bazate pe:
  - *factorul uman* (engl. **human-based testing**, HbT);
  - *instrumente specializate* (engl. **computer-based testing**, CbT).

# Analiză statică. Clasificare

- metode de analiză statică bazate pe:
  - *factorul uman* (engl. **human-based testing**, HbT), i.e., **reviews**:
    - **formale**: inspectare Fagan, technical review, walkthroughs;
    - **informale**: buddy check, pairing, pair review, over-the-shoulder, e-mail pass-around;
  - *instrumente specializate* (engl. **computer-based testing**, CbT), i.e., **static analysis**:
    - **tool-assisted**: style checker, corecteness checker;
- metode hibride: **pair-programming**.



# METODE BAZATE PE FACTORUL UMAN

---

Definiție. Motivație. Obiective

Avantaje și dezavantaje. HbT vs CbT

Inspectare

Walhthroughs

Pair-Programming

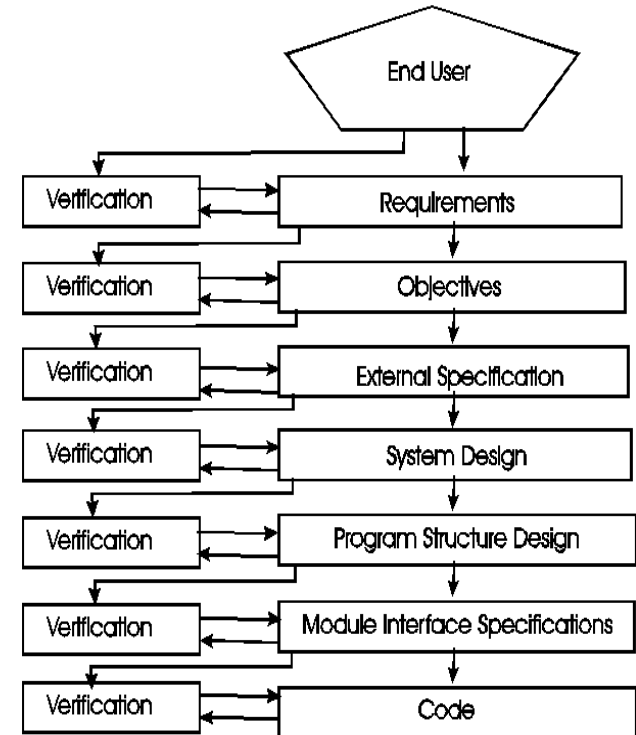
# Metode bazate pe factorul uman. Definiție

- **metodă HbT** [[Young2008](#), [Frentiu2010](#)]
  - verificare efectuată de o persoană sau un grup persoane la sfârșitul unei etape a procesului de dezvoltare și înainte de a demara următoarea fază de dezvoltare;
- **exemplu:**
  - activitate: **inspectarea codului sursă**;
  - se efectuează *după* etapa de implementare și *înainte* de începerea testării.



# Metode HbT. Obiective. Motivație

- **obiective**
  - *identificarea defectelor;*
- **motivație**
  - *utilizarea metodelor HbT contribuie la creșterea productivității și a gradului de încredere că rezultatul obținut îndeplinește cerințele:*
    - costul de corectare (eliminare) al defectelor crește odată cu parcurgerea etapelor de dezvoltare a softului;
    - modificarea comportamentului programatorilor la demararea analizei dinamice, i.e., la depanare se introduc mai multe bug-uri.



# Metode HbT. Avantaje și dezavantaje

## Avantaje

- sunt implicate în proces și **alte persoane** pe lângă autorul documentului verificat;
- permite **localizarea** defectelor;
- **identifică între 30% și 70% din bug-urile de proiectare și implementare ale produselor soft.**

## Dezavantaje

- nu sunt eficiente la identificarea **erorilor majore** de proiectare;
- nu pot evidenția situații excepționale care apar în utilizarea propriu-zisă a softului.

# INSPECTARE FAGAN

---

Definiție. Caracteristici

Echipa de inspectare. Atribuțiile membrilor

Activități de inspectare

Checklists. Definiție. Motivație. Tipuri de checklists

Avantaje

# Inspectare Fagan. Definiție. Caracteristici

- 1976 – Fagan [[Fagan1976](#)] introduce la IBM procesul de inspectare;
- **Inspectare**
  - proces structurat prin care se încearcă **identificarea defectelor** din **documentele elaborate** pe parcursul etapelor de dezvoltare a softului, pe baza unor **criterii prestabilite**;
- **Caracteristici**
  - **echipa de inspectare (4 membri):**
    - moderator, autor, secretar, prezentator;
  - **activități de inspectare (6 etape):**
    - planificarea, prezentarea, pregătirea, ședința de analiză, corectarea, reinspectarea;
  - **tipuri de erori căutate:** *checklists*, adaptate tipului de document inspectat;
  - **timp de desfășurare:** 90-120 minute.

# Echipa de inspectare. Atribuțiile membrilor

- **moderator**
  - distribuie materialele și planifică sesiunile de inspectare; conduce sesiunea de inspectare;
  - urmărește modul în care sunt corectate erorile;
- **autorul** documentului inspectat (analist, proiectant, programator, tester);
  - răspunde la întrebările adresate de membrii echipei, clarifică nelămuririle semnalate de către aceștia;
  - participă la discuțiile purtate în timpul ședinței de analiză; remediază defecțiunile constatate;
- **secretar**
  - redactează concluziile ședinței de analiză;
  - înregistrează defectele semnalate și problemele discutate într-un document (raport de inspectare);
- **prezentator** (reader)
  - citește în cadrul ședinței de analiză părți ale documentului inspectat;
- **inspectori** – cu excepția autorului, toți ceilalți sunt considerați inspectori;
  - analizează documentul primit cu scopul de a identifica cât mai multe defecte (bug-uri).

# Activități de inspectare (1)

## 1. **planificarea** (*engl. planning*)

- moderatorul alege membrii echipei de inspectare;
- distribuie materialele tuturor membrilor echipei și atribuie sarcini de inspectare;
- verifică dacă documentul care trebuie inspectat este complet și acceptabil pentru a fi inspectat;

## 2. **prezentarea** (*engl. overview*) – **nu este obligatorie**

- se prezintă detaliile materialului inspectat tuturor membrilor echipei de inspectare;
- moderatorul poate decide dacă este necesară etapa de prezentare sau se trece direct la pregătirea individuală;

## 3. **pregătirea individuală** (*engl. preparation*)

- citirea atentă și înțelegerea documentului primit pentru inspectare;
- inspectorii rețin toate observațiile critice și formulează întrebări referitoare la aspectele care nu sunt clare.



# Activități de inspectare (2)

## 4. **ședința de inspectare** (*engl. inspection meeting*)

- se discută observațiile critice ale fiecărui inspector;
- secretarul notează observațiile considerate prin consens ca fiind defecte și ulterior redactează concluziile inspectării;
- concluziile inspectării sunt predate autorului documentului inspectat pentru a corecta greșelile;

## 5. **corectarea** (*engl. rework*)

- autorul efectuează modificările necesare și corectează erorile;

## 6. **reinspectarea** (*engl. follow-up*)

- se verifică dacă modificările efectuate au eliminat erorile;
- se poate reduce la o întâlnire între autor și moderator.

# Checklists. Definiție. Motivație. Tipuri de checklists

- **checklist** = listă cu defecte frecvent întâlnite într-un anumit tip de document;
- **motivație**
  - obiectivul inspectării: identificarea defectelor;
  - în raport cu documentul analizat, se urmărește identificarea unor bug-uri specifice;
- **tipuri de checklists** pentru inspectarea:
  - documentației de specificare;
  - documentației de analiză;
  - documentației de codificare;
  - documentației de testare.
- **Fiecare checklist conține aspecte particulare documentelor inspectate și sunt rezultatul experienței acumulate în identificarea greșelilor întâlnite frecvent în desfășurarea unor etape de dezvoltare software.**

# Tipuri de checklists (1)

- checklist utilizat la inspectarea **documentației de specificare**:
  1. specificația respectă cerințele beneficiarului?
  2. există ambiguități în specificare?
  3. datele de intrare și de ieșire, cât și condițiile de intrare și de ieșire asociate sunt specificate corect?
  4. există cerințe care nu sunt specificate în document?
  5. există cerințe de precizie a datelor? sunt clar exprimate?
  6. există cerințe de performanță?
- checklist utilizat la inspectarea **documentației de analiză**:
  1. se respectă specificația?
  2. toate funcționalitățile din specificare au fost descrise?

## Tipuri de checklists (2)

- checklist utilizat la inspectarea **documentației de codificare**:
  1. codul sursă respectă cerințele de proiectare, specificațiile și cerințele utilizatorului?
  2. sunt apelate toate metodele?
  3. sunt inițializate toate variabilele?
  4. aspecte analizate în mod special: cicluri infinite, accesarea unui index non-valid, alocarea și accesarea memoriei.
- checklist utilizat la inspectarea **documentației de testare**:
  1. toate cazurile de testare au fost documentate complet?
  2. cazurile de testare sunt relevante?
  3. datele de testare satisfac criteriul de acoperire ales?
  4. la testarea de integrare este clară ordinea de integrare?

# Inspectare Fagan. Avantaje

- **avantaje**

- permite descoperirea defectelor devreme;
- reducere costul și timpul de dezvoltare;
- metodă de grup – membrii echipei conlucrează;
- modalitate de învățare la nivelului echipei;
- **stabilește sursa defecțiunii, nu oferă doar indicii** referitoare la existența lor, e.g., testarea;
- elimină stresul depanării într-un timp foarte scurt.

- **Inspectare vs. Testare** [[Collard2003](#)]

- identificarea, localizarea și eliminarea defectului;
- abordare aplicată în două etape (individual și apoi în grup);
- checklists se focalizează pe anumite părți ale documentului care sunt **predispușe la introducerea de defecte** pe parcursul dezvoltării softului.

# WALKTHROUGHS

---

Definiție. Caracteristici

Walkthroughs vs Inspectare

# Walkthroughs. Definiție. Caracteristici

- **walkthroughs** [[Yourdon1979](#)]
  - procesul prin care se încearcă identificarea defectelor din documentele elaborate pe parcursul etapelor de dezvoltare a softului sub îndrumarea autorului documentului;
- **caracteristici** [[Yourdon1979](#), [Collard2003](#)]
  - **echipa de realizare (3-5 membri)** :
    - secretar, inspectori și moderator (autorul documentului inspectat, i.e., analist, proiectant, programator, tester);
  - **activitățile de walkthrough (4 etape)**:
    - planning, meeting, rework, follow-up;
  - aplică tehnici de identificare a erorilor diferite de inspectarea Fagan, i.e., **nu se folosesc checklists**;
  - **timp de realizare**: 90-120 minute.

# Walkthroughs vs Inspectare Fagan

## Walkthroughs

- activitate mai puțin riguroasă;
- echipa este formată din 3-5 membri;
- se desfășoară în 4 etape;
- nu are pretenția identificării tuturor defectelor;
- autorul conduce echipa de walkthrough;
- se folosesc scenarii prestabilite.

## Inspectare Fagan

- activitate riguroasă;
- echipa este formată din 4 membri;
- se desfășoară în 6 etape;
- identifică defectele des întâlnite;
- moderatorul conduce echipa de inspectare;
- folosește checklists pentru identificarea defectelor.



# TECHNICAL REVIEW

---

Definiție. Caracteristici

Technical review vs Inspectare

# Technical Review. Definiție. Caracteristici

- **technical review** [[TechReview2019](#)]
  - **tip de review formal realizat de o echipă** formată din **personal calificat tehnic** care examinează conformitatea unui document (work product) cu scopul pentru care este utilizat și identifică diferențele față de specificații și standarde;
- **caracteristici**
  - **echipa de realizare (3-5 membri) :**
    - secretar, inspectori, moderator (conduce echipa) și autorul documentului inspectat (toți fiind persoane calificate în același domeniu și fiind considerate *peer reviewers* față de autor);
    - autorul nu este și secretar;
  - **activitățile de technical review (3-4 etape):**
    - planning, preparation (obligatoriu), meeting (opțional), rework;
  - **timp de realizare:** 60-90 minute.

# Technical Review vs Inspectare Fagan

## Technical Review

- activitate mai puțin riguroasă;
- echipa este formată din 3-5 membri;
- se desfășoară în 4 etape;
- **objective:** identificarea unui consens, identificarea posibilelor defecte, identificarea de idei noi și motivarea autorului să îmbunătățească documentele elaborate folosind implementări alternative;
- utilizarea checklists este opțională.

## Inspectare Fagan

- activitate riguroasă;
- echipa este formată din 4 membri;
- se desfășoară în 6 etape;
- **objective:** evaluarea calității, identifică defectele des întâlnite;
- folosește checklists pentru identificarea defectelor.

# INFORMAL REVIEW

---

Definiție. Caracteristici

Informal Review vs Inspectare

# Informal Review. Definiție. Caracteristici

- **informal review** [[TechReview2019](#)]
  - se realizează fără o procedură formală sau documentată;
- exemple: **buddy check, pairing, pair review, over-the-shoulder, e-mail pass-around.**
- **caracteristici**
  - **realizare în pereche (2 persoane) sau echipe (>2 persoane):**
    - autorul, inspectori (cel puțin unul, toți sunt *peer reviewers* față de autor);
  - **activitățile asociate unui review informal (1-2 etape):**
    - meeting (opțională), rework;
  - **timp de realizare: 15-60 minute.**

# Informal Review vs Inspectare Fagan

## Informal Review

- activitate se scurtă durată, puțin riguroasă;
- perechi (autor, inspector) sau echipe de membri;
- se desfășoară în 1-2 etape;
- **obiective:** identificarea posibilelor defecte, identificarea de **idei noi**, rezolvarea unor probleme minore;
- inspectorul este un coleg;
- utilizarea checklists este opțională, rezultatele se pot documenta;
- gradul de utilitate depinde de inspector;
- utilizată frecvent metodologiile Agile.

## Inspectare Fagan

- activitate riguroasă;
- echipa este formată din 4 membri;
- se desfășoară în 6 etape;
- **obiective:** evaluarea calității, identifică defectele des întâlnite;
- moderatorul conduce echipa de inspectare;
- folosește checklists pentru identificarea defectelor.

# PAIR-PROGRAMMING

---

Definiție. Caracteristici

# Pair-Programming. Definiție. Caracteristici

- **pair-programming**
  - metodă de elaborare a programelor, în care două persoane lucrează împreună;
- **caracteristici**
  - combină activitățile: inspectarea codului și implementarea (codificarea);
  - programatorii alternează rolurile;
  - **activități de inspectare:**
    - nu sunt determinate de checklists;
    - **se bazează pe împărtășirea acelorași principii de programare și a unui stil de programare asemănător;**
  - **timp de desfășurare:** durata unei zile normale de muncă, fără exces de ore suplimentare sau presiunea unui program de lucru strict;
  - nu există mediatori, iar responsabilitatea pentru atmosfera de lucru deschisă și non-agresivă depinde de programatori.



PENTRU EXAMEN...

---

# Pentru examen...

- **concepte, caracteristici, asemănări și diferențe:**
  - verificare, validare; verificare vs. validare;
  - eroare, defect/bug, defecțiune; eroare vs. defect/bug vs. defecțiune;
  - stakeholders, calitate, QA, QC;
  - analiza statică vs analiza dinamică;
  - HbT, motivație;
  - inspectare Fagan, walkthroughs, technical review, informal review:
    - descriere, rolurile membrilor echipei, activitățile asociate și descrierea lor, avantaje;
  - pair-programming:
    - caracteristici, avantaje.

# Cursul următor...

- **Testare**
  - modele folosite în testare;
  - planuri de testare;
  - cazuri de testare;
  - raportarea testării;
- **Technici de testare Black-box**
  - împărțirea în clase de echivalență;
  - analiza valorilor limită;
- **Testing Management Tool – TestLink**
  - prezentare tool.

# Referințe bibliografice

- **[Crosby1980]** Philip B. Crosby, *Quality Is Free*, Signet Shakespeare, 1980.
- **[Juran1998]** A. Blanton Godfrey, Joseph Juran, *JURANS QUALITY HANDBOOK*, McGraw-Hill, 1998.
- **[Weinberg1992]** Gerald Weinberg, *Quality Software Management , Vol. 1: Systems Thinking*, Dorset House Publishing, 1992.
- **[Pressman2000]** Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, Inc., 2000.
- **[Pal2013]** Kaushik Pal, *Software Testing: Verification and Validation*, <http://mrbool.com/software-testing-verification-and-validation/29609>
- **[Fagan1976]** M. E. Fagan, *Design and code inspections to reduce errors in program development*, IBM Systems Journal, pages 182–211, 1976.
- **[Collard2003]** J. F. Collard, I. Burnstein. *Practical Software Testing*. Springer-Verlag New York, Inc., 2003.
- **[Yourdon1979]** E. Yourdon, *Structured Walkthroughs*, Prentice-Hall, Englewood Cliffs, NJ, 1979.
- **[Myers2004]** Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004
- **[Young2008]** M. Pezzand, M. Young. *Software Testing and Analysis: Process, Principles and Techniques*. John Wiley and Sons, 2008.
- **[Frentiu2010]** M. Frentiu, *Verificarea si validarea sistemelor soft*, Presa Universitara Clujeana, 2010.
- **[TechReview2019]** Cania Consulting, Informal and formal testing reviews, <https://cania-consulting.com/2019/10/12/test-manager-guide-to-reviews/>