

CURS 02.

INSPECTARE

Verificare, validare și testare automată
[05 Octombrie 2019]

Lector dr. Camelia Chisăliță-Crețu
Universitatea Babeș-Bolyai, NTT Data
Programul Postuniversitar de Pregătire și Formare Profesională în Informatică

Conținut

- Calitatea produselor soft
 - Evaluarea calității unui produs soft
 - Metode de verificare
- Metode bazate pe factorul uman
 - Definiție. Motivație. Caracteristici
 - Inspectare
 - Walkthroughs
 - Pair-Programming
- Întrebări pentru examen
- Bibliografie

CALITATEA PRODUSELOR SOFT

Evaluarea calității unui produs soft

Metode de verificare

Calitatea produselor soft

- 4 definiții
 - ***“produsul soft este conform cu cerințele documentate”*** [[Pressman2005](#)]:
 - conformitatea cu cerințele funcționale și de performanță precizate și documentate explicit;
 - ***“produsul soft este conform cu cerințele reale ale utilizatorului”*** [[Crosby1980](#)]:
 - conformitate cu cerințele reale, nu doar cu cerințele documentate;
 - ***“produsul soft este adecvat pentru a fi utilizat”*** [[Juran1998](#)]:
 - orice aspect care îl mulțumește sau nu îl mulțumește pe utilizator, i.e., *satisfiers, dissatisfiers*;
 - ***“produsul soft este relevant/important pentru o persoană”*** [[Weinberg1992](#)]:
 - calitatea este subiectivă;

Evaluarea calității unui produs soft

- asigurarea calității vs. controlul calității (Curs 01);
- evaluarea calității = controlul calității
 - aplicarea unor *metode de verificare* prin care se stabilește modul în care produsul soft satisface anumite criterii (attribute) de calitate necesare care indică nevoile beneficiarului;
- beneficiar
 - client, utilizator final;
 - programator, tester, manager de proiect;

Activități asociate controlului calității

Analiza statică

- 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;
- **NU** presupune execuția propriu-zisă a programului dezvoltat;

Analiza dinamică

- 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);
- **include activitatea de execuție propriu-zisă a programului (testare);**

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

Metode de verificare

- 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 verificare bazate pe:
 - *factorul uman* (engl. **human-based testing**, HbT);
 - *calculator* (engl. **computer-based testing**, CbT);

METODE BAZATE PE FACTORUL UMAN

Definiție. Motivație. Obiective

Avantaje și dezavantaje. HbT vs CbT

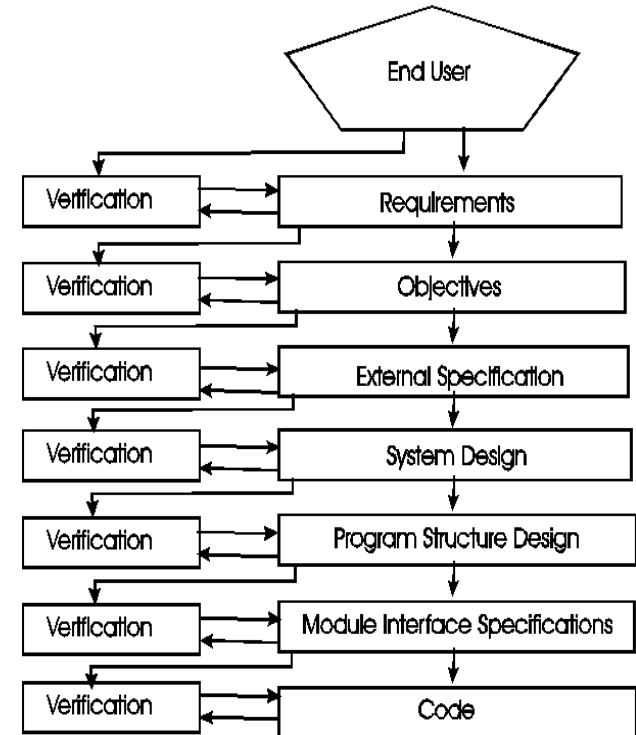
Inspectare

Walkthroughs

Pair-Programming

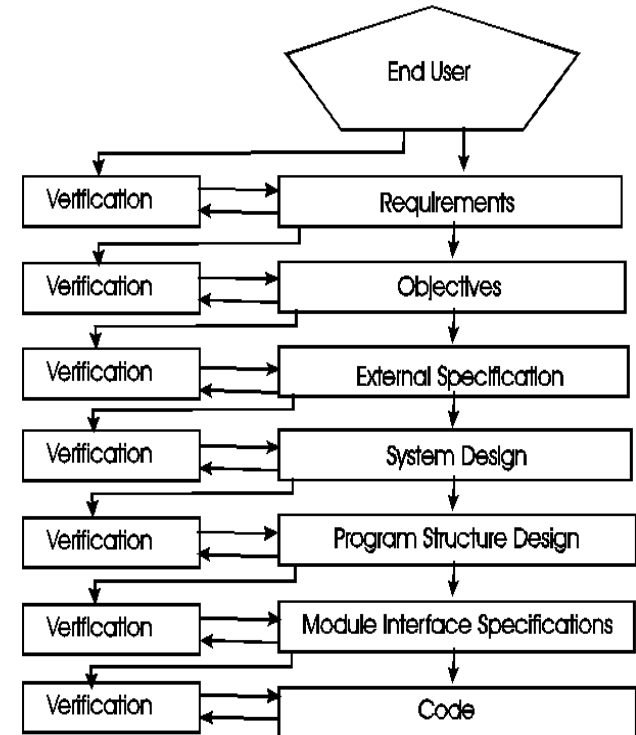
Metode bazate pe factorul uman

- **metodă HbT** [[Young2008](#), [Frentiu2010](#)]
 - verificare efectuată de o persoană sau un grup persoane la sfârșitul fiecărei etape a procesului de dezvoltare a softului și înainte de a demara următoarea fază de dezvoltare;
- **tipuri de metode HbT:**
 - inspectare, walkthroughs, pair-programing;
- **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 CbT, 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;

Metode HbT vs Metode CbT

Metode HbT

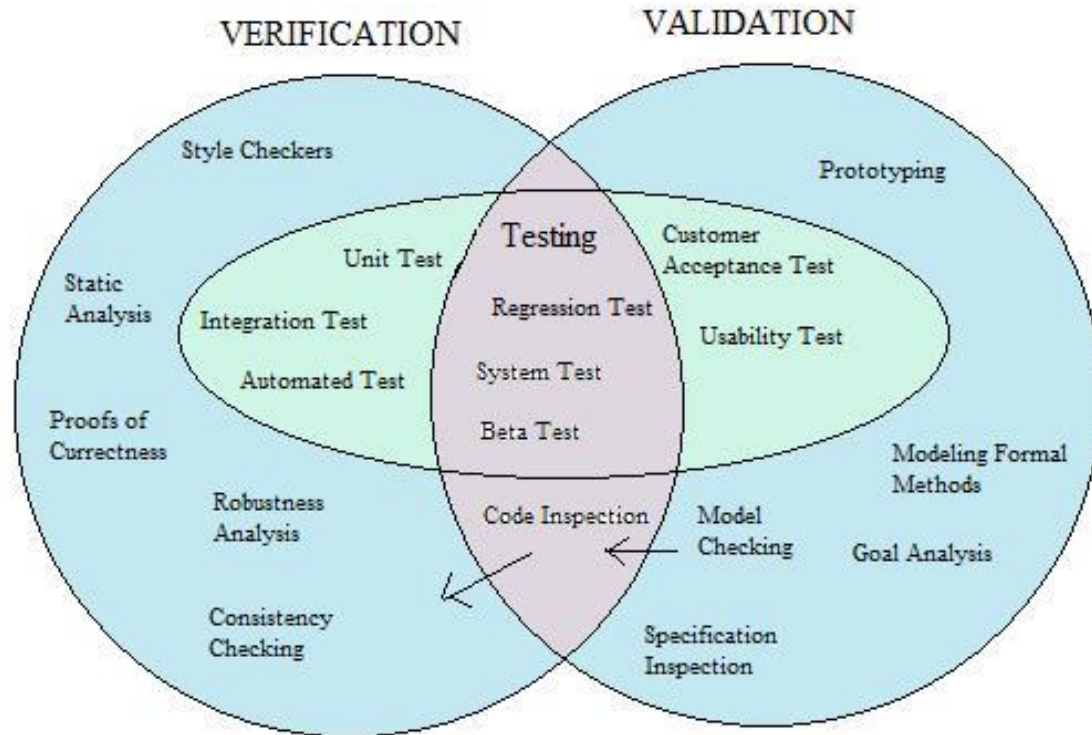
- permit identificarea mai multor erori (greșeli) care pot fi corectate simultan;
- nu presupun execuția programului;

Metode CbT

- sugerează doar un simptom, fiecare eroare identificată fiind eliminată individual;
- este necesar ca programul să fie executat;
- poate să evidențieze o defecțiune doar în anumite situații;

Metodele HbT (e.g., inspectare) și metodele CbT (e.g., testare) sunt metode complementare.

Metode HbT și metode CbT în Verificare și Validare



- metodele HbT și metodele CbT sunt activități de verificare și validare

- sursa: [\[Pal2013\]](#)

INSPECTARE

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 specificație 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 nivelul echipei;
- **stabilește sursa defecțiunii, nu oferă doar indicii** (e.g., testarea) referitoare la existența lor;
- 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 de grup);
- checklists se focalizează pe anumite părți ale documentului care sunt predispuse 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;
- **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

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;
- **folosind scenarii prestabilite.**

Inspectare

- 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.**

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.

ÎNTREBĂRI PENTRU EXAMEN

Întrebări cu răspuns scurt

Întrebări cu răspuns lung

Întrebări cu răspuns scurt

- **Întrebări cu răspuns scurt:**

1. Definiți noțiunea: analiză statică. Exemplificați.
2. Definiți noțiunea: analiză dinamică. Exemplificați.
3. Enumerați și explicați trei avantaje ale metodelor HbT.
4. Definiți noțiunea: inspectare Fagan. Exemplificați.
5. Enumerați și descrieți pe scurt componența echipei de inspectare Fagan.
6. Enumerați și descrieți pe scurt activitățile de inspectare Fagan.
7. Definiți noțiunea: walkthroughs. Exemplificați.
8. Enumerați caracteristicile noțiunii: pair-programming. Exemplificați.

Întrebări cu răspuns lung

- **Întrebări cu răspuns lung:**

1. Comparați noțiunile: analiză statică (HbT) și analiză dinamică (CbT). Exemplificați.
2. Descrieți motivația și obiectivele aplicării metodelor HbT. Exemplificați.
3. Descrieți avantajele și dezavantajele aplicării HbT. Exemplificați.
4. Enumerați componența echipei de inspectare Fagan. Descrieți atribuțiile membrilor echipei de inspectare Fagan. Exemplificați.
5. Descrieți activitățile de inspectare Fagan. Exemplificați pentru un document de cod sursă.
6. Comparați noțiunile: inspectare Fagan și walkthroughs. Exemplificați.
7. Explicați trei elemente ale unui checklist (la alegere). Exemplificați.
8. Descrieți noțiunea: pair-programming. Exemplificați.

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/296091>
- **[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.