CURS 01A.
VERIFICARE ŞI VALIDARE

Verificarea şi validarea sistemlor soft
[27 Februarie 2018]

Lector dr. Camelia Chisăliţă-Creţu
Universitatea Babeş-Bolyai

Conţinut

- Verificare şi validare
- Calitatea produselor soft
  - Definiții
  - Asigurarea calității vs. Controlul calității
  - Activități asociate controlului calității
- Defect software
  - Terminologie
  - Costul unui bug software
  - Defecte/Buguri software celebre
- Bibliografie

VERIFICARE ŞI VALIDARE

Verificare
Validare
Verificare vs. Validare

Verificare și Validare. Definiţie SEI

- SEI (Software Engineering Institute) [NT2005]
- **Verificare**
  - procesul prin care se asigură că produsul este dezvoltat conform cerinţelor, specificaţiilor şi standardelor;
  - întrebare asociată: *Dezvoltăm corect produsul? (Are we building the product right?)*
- **Validare**
  - procesul prin care se asigură că produsul dezvoltat satisface cerinţele utilizatorului;
  - întrebare asociată: *Dezvoltăm produsul corect/cerut (cel de care are nevoie clientul)? (Are we building the right product?)*

Verificare și Validare. Definiţie NASA

- NASA (National Aeronautics and Space Administration) - Software Assurance Guidebook and Standard [NASA];
- **Verificare**
  - procesul care asigură pentru produsul soft că fiecare pas din procesul de dezvoltare duce la obținerea unui produs corect;
- **Validare**
  - procesul care asigură că produsul soft va satisface cerințele funcționale și non-funcționale.

Verificare vs. Validare

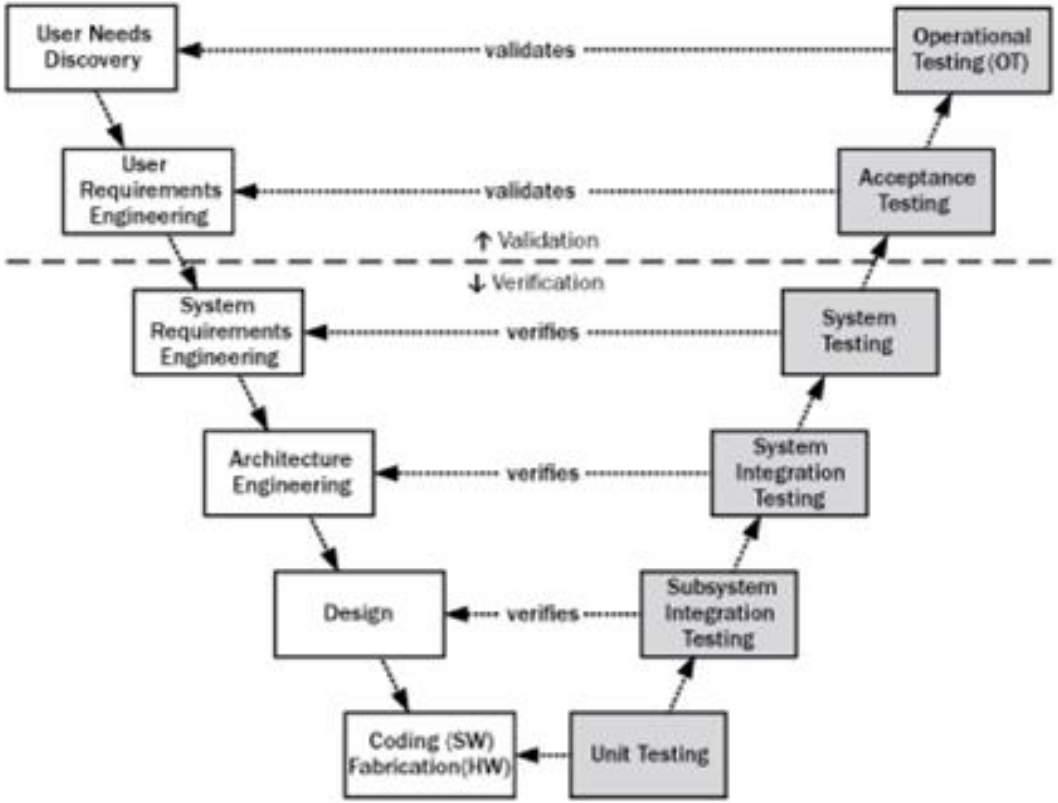
Verificare

- stabilește dacă rezultatul unei etape de dezvoltare satisface cerințele acelei etape;
- asigurare a consistenței, completitudinii, corectitudinii;
- aplică tehnici de analiză statică și analiză dinamică;

Validare

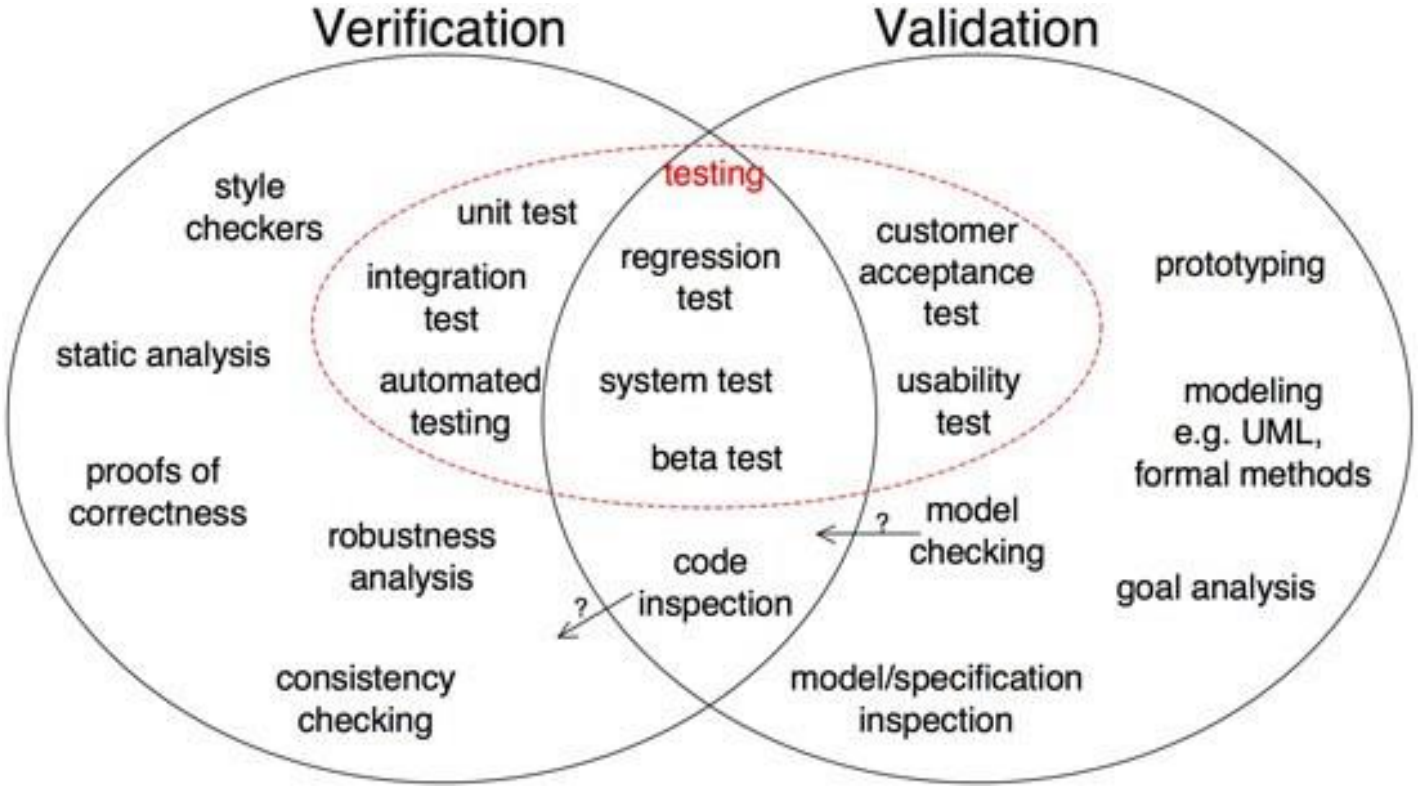
- confirmă dacă produsul satisface cerințele de utilizare;
- se desfășoară spre sfârșitul procesului de dezvoltare, cu scopul de a demonstra că întregul sistem satisface nevoile și așteptările;
- se aplică asupra întregului sistem, în contextul real în care va funcționa, folosind diferite tipuri de testare.

Verificare şi Validare în modelul V



- sursa: [Firesmith2015]

Activităţi de Verificare şi Validare



- sursa: [Easterbrook2010]

CALITATEA PRODUSELOR SOFT

Definiții

Asigurarea calității vs. Controlul calității;

Activități asociate controlului calității unui produs soft

Calitatea produselor soft. Definiții (1)

- ***"produsul soft este conform cu cerințele documentate"*** [Pressman2005]:
  - conformitatea cu cerințele funcționale și de performanţă precizate şi documentate explicit în standarde de dezvoltare și caracteristicile implicite pe care un produs soft dezvoltat le are;

- ***"produsul soft este conform cu cerințele reale ale utilizatorului"*** [Crosby1980]:
  - conformitatea cu cerințele reale ale utilizatorului care pot fi incluse sau nu în specificațiile scrise;
  - **conformitate cu cerințele (nevoile) reale, nu doar cu cerinţele documentate**;

Calitatea produselor soft. Definiții (2)

- ***"produsul soft este adecvat pentru a fi utilizat"*** [Juran1998]:
  - *satisfiers* – orice aspect care îl mulțumește pe utilizator;
  - *dissatisfiers* – orice aspect care îl nemulțumește pe utilizator;
  - categorii (tipuri) de utilizatori: manager de proiect, programator, tester, client;

- ***"produsul soft este relevant/important pentru o persoană"*** [Weinberg1992]:
  - **calitatea este subiectivă;**
  - un aspect care are relevanță/importanță însemnată pentru un utilizator poate fi mai puțin important pentru un alt utilizator din aceeaşi categorie de utilizatori.

Asigurarea calității vs. Controlul calității

Asigurarea calității

- *engl.* **Quality Assurance (QA):**
  - **Focalizare**: calitatea proceselor;
  - **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?

Controlul calității

- *engl.* **Quality Control (QC):**
  - **Focalizare**: calitatea produsului elaborat;
  - **Obiectiv**: identifică problemele în produsul obținut;
  - **Întrebare**: Cum se controlează calitatea rezultatelor obținute (e.g., work products) în urma activităților desfășurate?

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, funcţională, non-funcţională), *niveluri de testare* (testare unitară, testare de integrare, testare de sistem, testare funcţională, testare de accpetare);
- **include activitatea de execuție propriu-zisă a programului (testare);**

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

DEFECT SOFTWARE

Terminologie
Când apare un bug într-un produs soft?
De ce apare un bug în procesul de dezvoltare software?
Costul unui bug software
Defecte/Buguri software celebre

Terminologie (1)

- **eroare** (*engl.* **error, mistake**; greșeală):
  - o acțiune umană care are ca rezultat un defect în produsul software [Patton2005];
- **defect** (*engl.* **fault**, i.e., **bug**):
  - consecință a unei erori [Patton2005];
  - un defect poate fi latent: nu cauzează probleme până când nu apar anumite condiții (*engl.* **failure triggers**) care determină execuția anumitor linii de cod sursă;
- **defecțiune** (*engl.* **failure**):
  - devierea de la comportamentul obișnuit al unei componente software;
  - apare atunci când comportamentul observabil al programului nu corespunde specificației sale;
  - procesul de manifestare a unui defect: când execuția programului întâlnește un defect, acesta provoacă o defecțiune [Patton2005];

Terminologie (2)

- **defect** (*engl.* **bug**, **software error**)
  - **orice aspect al unui produs soft care**
    - **cauzează reducerea inutilă și inadecvată a calității produsului soft** [BBST2008];
    - **constituie o amenințare asupra imaginii produsului** [BBST2008];
  - **exemple:** deficiențe de proiectare, greșeli în documentații, utilizare cu dificultate a programului;

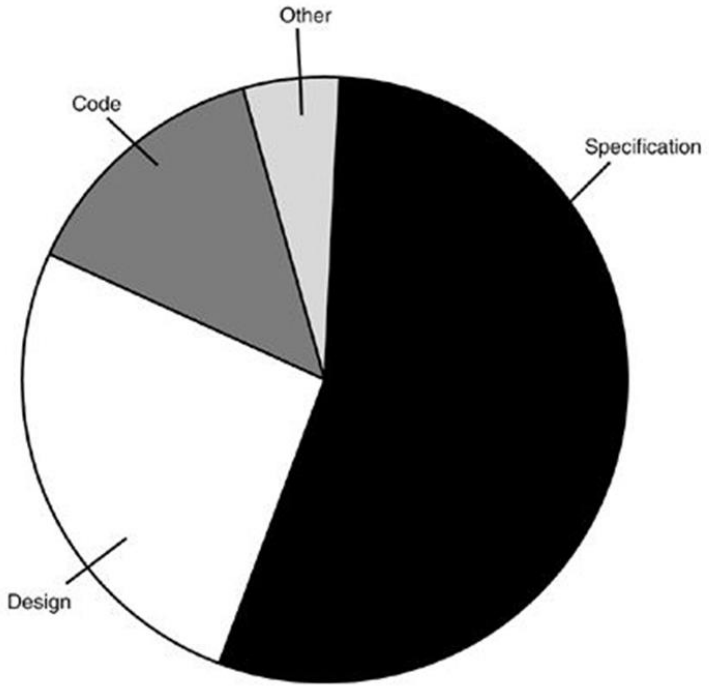
  - **totuși, anumite aspecte ale produsului pot limita calitatea acestuia, dar nu pot fi considerate defecte!**
  - **exemplu:** constrângeri de utilizare precizate sau nu în specificații;

  - **În cadrul acestui curs, orice deficiență sau problemă a produsului soft este denumită bug (defect).**
  - sinonime pentru bug: *engl.* **variance**, **problem**, **inconsistency**, **error**, **incident**, **anomaly** [Patton2005].

Când apare un bug într-un produs soft?

- Un bug software apare atunci când cel puțin una din următoarele situații are loc [Patton2005]:
  - Produsul soft nu face ce este precizat în specificația lui.
  - Produsul soft face ce nu este precizat în specificație.
  - Produsul soft face ce specificația precizează că **nu** trebuie făcut.
  - Produsul soft nu face ceea ce specificația ar trebui să precizeze.
  - Produsul soft este dificil de înțeles, greu de utilizat, lent. Testerul pune în evidență perspectiva utilizatorului final asupra podusului soft, adică produsul nu funcționează conform așteptărilor lui.

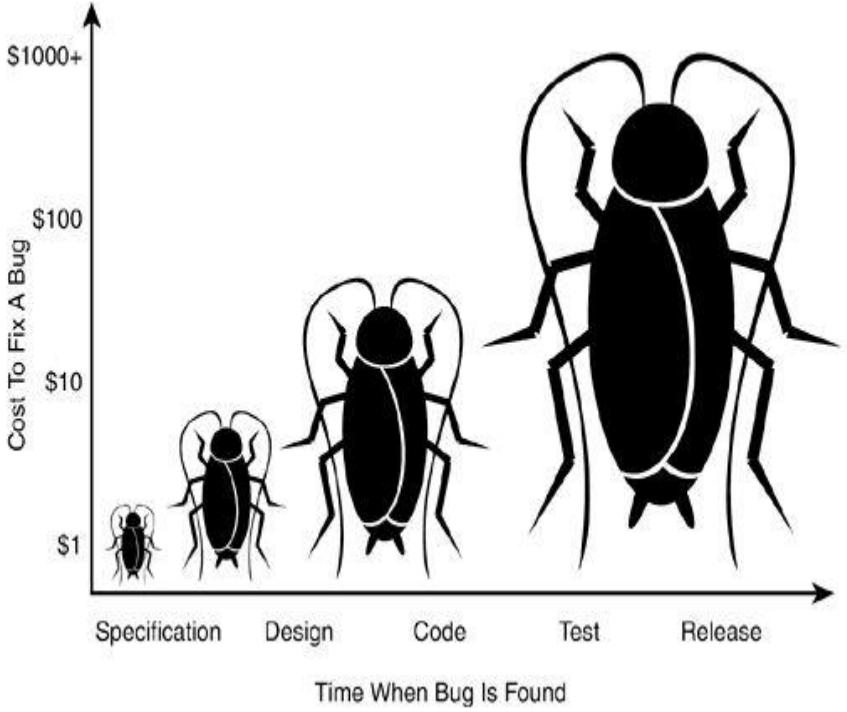
De ce apare un bug în procesul de dezvoltare software?



- **specificarea cerințelor:**
  - nu se scriu specificațiile, sunt superficiale, se schimbă continuu, nu sunt comunicate corespunzător întregii echipe de dezvoltare;
- **proiectare:**
  - sunt superficiale, nu se comunică eficient, se modifică;
- **implementare:**
  - complexitatea produsului soft, lipsa documentației (pentru codul sursă îmbunătățit), erori de redactare, presiunea termenului limită.

- **Care este etapa de dezvoltare în care se introduc cele mai multe defecte?**

Cât costă un bug?



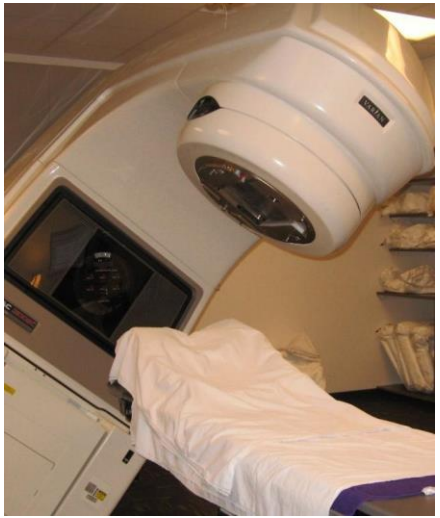
- **Care sunt costurile de eliminare a unui bug software?**
- costul eliminării bug-urilor crește pe măsură ce produsul soft este dezvoltat.

Buguri software celebre (1)

- **Naveta spațială Mariner 1 – 1962**
  - naveta spațială Mariner 1 a deviat de la traiectoria ei la scurt timp după lansare; a fost distrusă la 293 secunde după lansare;
  - **cauza:** eroare la transcrierea unei instrucțiuni în limbajul FORTRAN, determinând calculul eronat al traiectoriei;
  - **cost:** 18.5 milioane $

Buguri software celebre (2)

- **Tratamente împotriva cancerului – 1985**
  - dispozitivul Therac-25 fost folosit în terapia prin radiații;
  - **cauza:** programul a calculat greșit doza de radiații pe baza datelor de intrare, unii pacienți primind o doză de câteva ori mai mare decât cea normală;
  - **cost:** 3 pacienți decedați, 3 răniți prin iradiere.

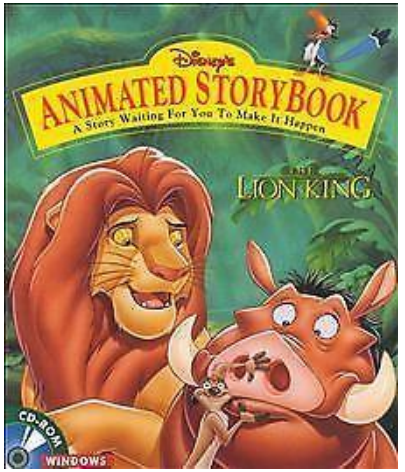


Buguri software celebre (3)

- **Sistemul de apărare american anti-rachetă – 1991**
  - sistemul american de apărare antirachetă MIM-104 Patriot situat în Arabia Saudită nu a reușit să detecteze atacuri cu rachete Scud irakiene;
  - **cauza:** o eroare de rotunjire la ceasul sistemului (un sfert de secundă) s-a cumulat, astfel încât la 14 ore, sistemul de urmărire își pierdea acuratețea, devenind incapabil să localizeze și să intercepteze rachetele;
  - **cost:** în atacul asupra unei cazarme din Dhahran au decedat 28 soldați americani;
  - eroarea fusese deja remediată de experții armatei americane, iar noua versiune a softului urma să ajungă cu o zi mai târziu.

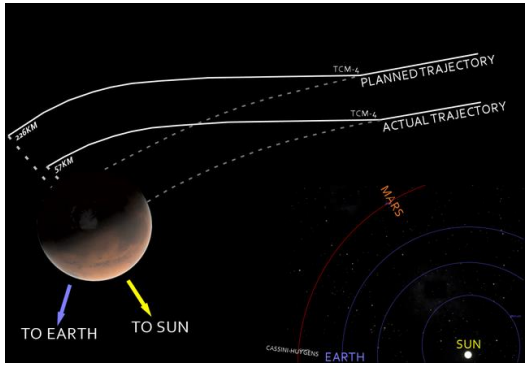
Buguri software celebre (4)

- **Jocul asociat desenului animat Disney Lion King – 1995**
  - la prima apariție pe piață a companiei Disney cu un joc pentru copii - ***The Lion King Animated Storybook*** - unii utilizatori nu au reușit să folosească produsul soft achiziționat;
  - **cauza:** compania Disney nu a testat produsul pe diferite modele de calculatoare personale existente pe piață;
  - **cost:** credibilitatea companiei, schimbarea unităților CD-ROM.



Buguri software celebre (5)

- **Naveta spațială Mars Climate Orbiter – 1998**
  - **obiectiv:** orbitarea planetei Marte și transmiterea informațiilor despre condițiile meteo;
  - **eveniment:** după o călătorie de 286 zile de pe Pământ, la intrarea în atmosfera planetei Marte, motoarele au deviat traiectoria navetei;
  - **rezultat:** dezintegrarea navetei în atmosferă;
  - **cauza:** două dintre echipele implicate în dezvoltarea aplicației foloseau sisteme de măsurare a distanței diferite, imperial (**inch**, **feet**) și cel metric (**m**, **km**).



Buguri software celebre (6)

- **Naveta spațială Mars Polar Lander – 1998**
  - **obiectiv:** studierea solului și a climei din regiunea Planum Australe de pe Marte;
  - pentru mecanismul de identificare a momentului când mototarele trebuie să fie oprite, NASA nu a folosit radare costisitoare, ci un senzor pe talpa picioarelor navetei, care determina oprirea alimentării cu combustibil;
  - **eveniment:** la intrarea în atmosfera planetei Marte, programul a interpretat vibrațiile navetei – cauzate de turbulențele din atmosferă – că aceasta ar fi aterizat și a oprit motoarele navetei;
  - **rezultat:** prăbuşirea navetei de înălţimea de 40m faţă de suprafaţa planetei Marte;
  - **cauza:** testare incompletă – procedura de aterizare a fost împărțită în două etape, care au fost testate independent; nu s-a realizat testarea de integrare.

Buguri software celebre (7)

- **Knight Capital Group – 2012**
  - casa de brokeraj Knight Capital Group a suferit o pierdere consistentă pe bursa din New York;
  - **cauza:** sistemul a introdus pe bursa de la New York tranzacții care au provocat fluctuații violente ale prețurilor multor acțiuni;
  - **cost:** pierderi de 440 milioane $ în doar 45 minute.

PHOTO: STAN HONDA/AFP/GETTY IMAGES

Buguri software celebre (8)

- **Termostatul Nest – 2016**
  - termostatul Nest Learning Thermostat (achiziţionat de Google în 2014 pentru 3.2 mld $) nu a permis controlul temperaturii în locuinţele în care a fost instalat - imposibilitatea de a-l utiliza pentru încălzire sau prepararea apei calde în timpul unui weekend friguros;
  - **cauza:** update-ul de firmware pentru device împreună cu existenţa unor filtre necurăţate şi centrale termice incompatibile; aceşti factori au dus la descărcarea bateriei device-ului;

Referințe bibliografice

- **[Firesmith2015]** Donald Firesmith, *Four Types of Shift Left Testing,* https://insights.sei.cmu.edu/sei_blog/2015/03/four-types-of-shift-left-testing.html
- **[NT2005]** K. Naik and P. Tripathy. *Software Testing and Quality Assurance*, Wiley Publishing, 2005.
- **[NASA]** NASA, https://www.grc.nasa.gov/www/wind/valid/tutorial/glossary.html.
- **[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.
- **[BBST]** BBST – Bug Advocacy Course, http://testingeducation.org/BBST/(http://testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf.
- **[Patton2005]** R. Patton, *Software Testing,* Sams Publishing, 2005.
- **[Easterbrook2010]** S. Easterbrook, *Software Testing,*http://www.easterbrook.ca/steve/2010/11/the-difference-between-verification-and-validation/.