

QUALITY ASSURANCE

CURS 1

STRUCTURA CURSULUI

- **Saptamana 1**

- Introducere; Mindset developer vs Tester;
- Responsabilitatile unui QA;
- Tipuri de teste: white box, black box, performance, stress, unit testing
- Metodologii de testare;

- **Saptamana 2**

- Whitebox si blackbox testing; cum se scrie un plan de test; cum se raporteaza defecte

- **Saptamana 3**

- Unit testing
- Test driven development

STRUCTURA CURSULUI

- **Saptamana 4**
 - Sesiune speciala: Invitati din companii
- **Saptamana 5**
 - Masurarea calitatii (buguri noi, buguri reactivate, complexitati, code coverage, etc)
 -
- **Saptamana 6**
 - Raportarea defectelor
 - Sisteme de urmarire a defectelor (bug tracking): Bugzilla, Mantis, TFS
- **Saptamana 7**
 - Sisteme integrate de dezvoltare Team Foundation Server
 - Teste manuale, teste automate, unit testing, performance testing, daily build, code coverage, code metrics

NOTAREA

- **Examen (desigur!)**
- **Doua teme de seminar**
 - Estimarea efortului
 - Realizarea unui plan de QA
 - Software pentru QA
- **Prezenta conteaza. Slideurile nu contin totul.**

BIBLIOGRAFIA

- **Cursurile lui Biden**
- **Cursurile lui Tian**
- **The Art Of Software Testing, Wiley – doar 250 pagini**
- **Test-Driven Development By Example, Addison Wesley**
- **Software Engineering with Microsoft Visual Studio Team System, Addison Wesley**
- **Demystifying the Black Art (Best Practices (Microsoft)) (Paperback) by Steve McConnell**
- **Code Complete: A Practical Handbook of Software Construction, Steve McConnell**

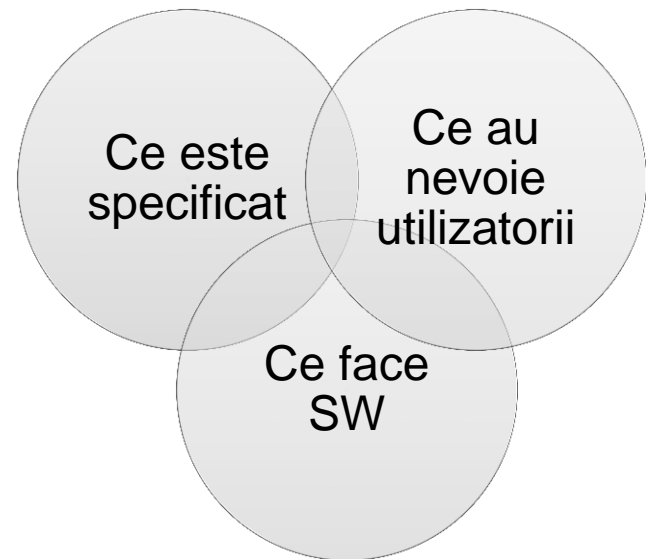
**CE ESTE
CALITATE?**

DEFINITIA CALITATII

- **Perceptia populara**
 - Ceva “bun” dar care nu poate fi cuantificat
 - Ceva luxos
- **Perceptia profesionistilor**
 - Conform cu cerintele (requirements)
 - Bun pentru utilizare
- **“Quality is in the eyes of the beholder”**

CALITATEA

- **Conform cu cerintele**
 - Cerintele au fost clar explicate si produsul trebuie sa se conformeze
 - Orice deviatie este considerata defect
 - Un produs de buna calitate contine mai putine defecte
- **Bun pentru utilizare**
 - Produsul se ridica la asteptarile utilizatorilor
 - Un produs de buna calitate ofera o satisfactie ridicata in utilizare

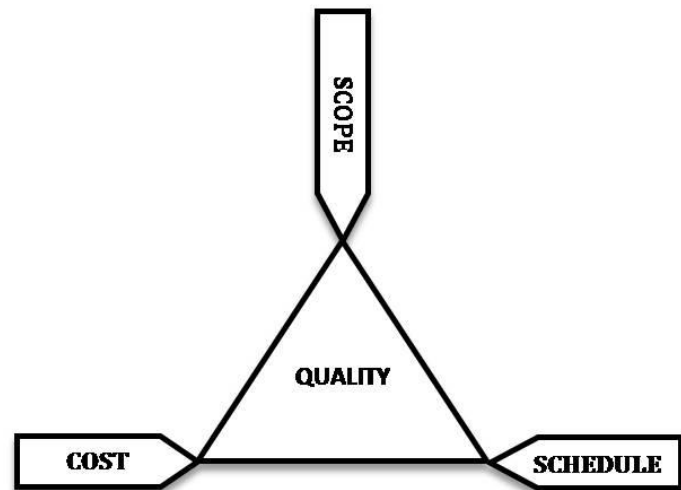


ACTORI IN SOFTWARE DEVELOPEMENT

- **Utilizatorii**
- **Dezvoltatorii**
- **QA**
- **Manageri de proiect**
- **Finantatorii**

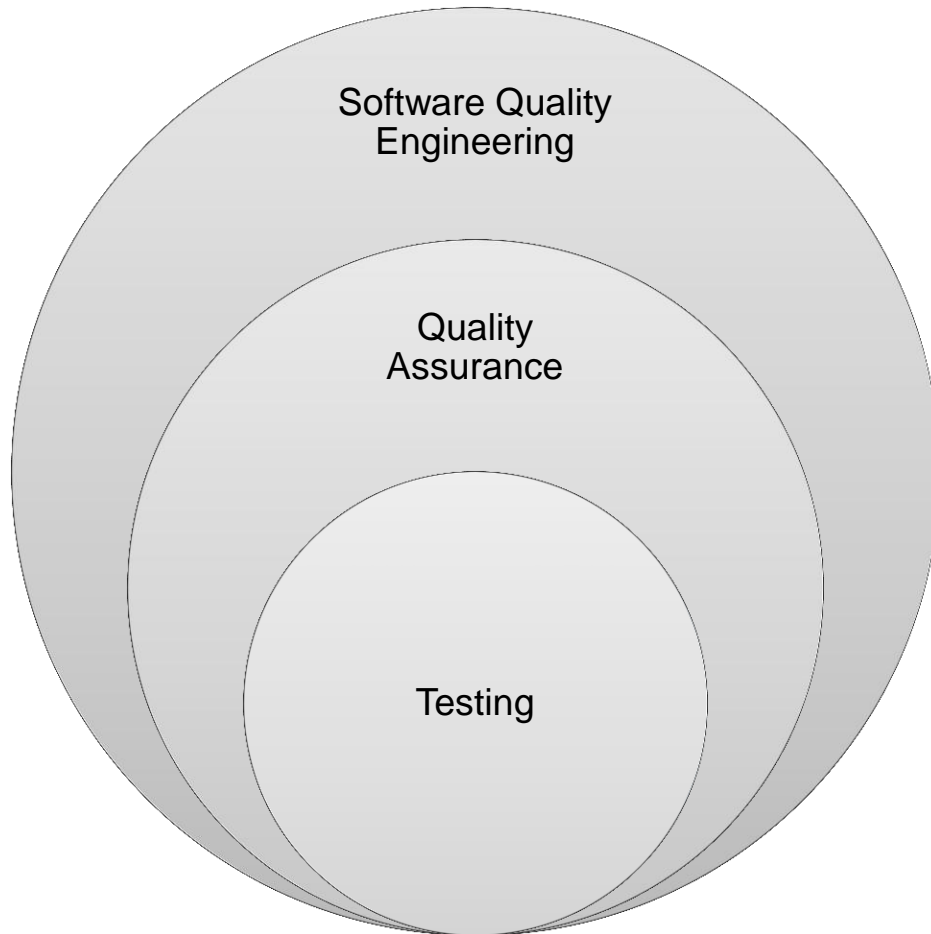
DILEMA: "QUICK, CHEAP, GOOD: PICK TWO"

- **Ce afecteaza calitatea?**
- **Schedule/Deadline**
 - Deadlineul proiectului
 - Sa intri pe piata la timp
- **Cost**
 - Sa te inscrii in costurile anticipate
- **Scope**
 - Aria de cuprindere a proiectului
 - Sa contina toate functionalitatile cerute sau in plus



**SOFTWARE
QUALITY
ENGINEERING**

SQE



- **Alte artefacte**
 - Cerintele sunt complete
 - Specificatiile sunt exacte
 - Documentele sunt corecte
 - Procedurile de instalare
- **Cititi “Code complete”**

TAP

Test

- Aproape de mediul de dezvoltare
- Acces la cod/baze de date
- Teste exhaustive

Acceptance

- Mediu comparabil cu cel de productie (economie)
- Acces la cod/baze de date
- Teste exhaustive

Production

- Timp pentru testare limitat
- Fara acces la cod/baze de date
- Teste limitate

TESTING

DEFINITIA

- **Cand testeaza programatorii pornesc de la o ipoteza falsa**
 - “Testarea este procesul prin care demonstreaza ca nu exista erori”
 - “Testarea are scopul de a demonstra ca programul implementeaza functiile necesare”
- **Mai corect:**
 - “Testarea este procesul de executie al unui program cu scopul de a gasi erori”
- **Implicatii**
 - Testarea este un proces distructiv
- **Strategii**
 - Black box, input/output
 - White box, logic driven

BLACK BOX

- **Se face abstractie de mecanismul intern de functionare**
- **Scopul este testarea exhaustiva a datelor de intrare**
- **Testarea tuturor perechilor valide si invalide de date de intrare**
- **Pentru unele sisteme este imposibil (ex. Compilatoare)**
- **Testarea exhaustiva este imposibila**
 - Nu putem garanta ca un sistem nu are erori
 - Problema fundamentala este una economica
 - Clase de echivalenta
- **Scopul este maximizarea numarului de erori cu un set finit de cazuri de test (test cases)**

WHITE BOX

- Este permisa investigarea structurii interne
- Setul de date este extras din examinarea logicii
- Scopul este executarea cel puțin o dată a tuturor instrucțiunilor din program
- **Probleme:**
 - Numarul de cai din program poate fi astronomic chiar și pentru programe simple
 - Nu identifică cai lipsă
 - Nu identifică probleme legate de corectitudinea datelor
- **Cel mai cunoscut instrument: unit testing**

PRINCIPII

- **O parte esentiala a unui plan de teste este definirea rezultatelor asteptate**
- **Programatorii ar trebui sa evite testarea propriului cod (constructiv vs distructiv)**
- **Firmele care fac software ar trebui sa evite testarea propriilor produse**
- **Inspecteaza cu atentie rezultatele fiecarui test**
- **Testele trebuie scrise atat pentru intrari invalide si neasteptate, cat si pentru intrari valide si asteptate**

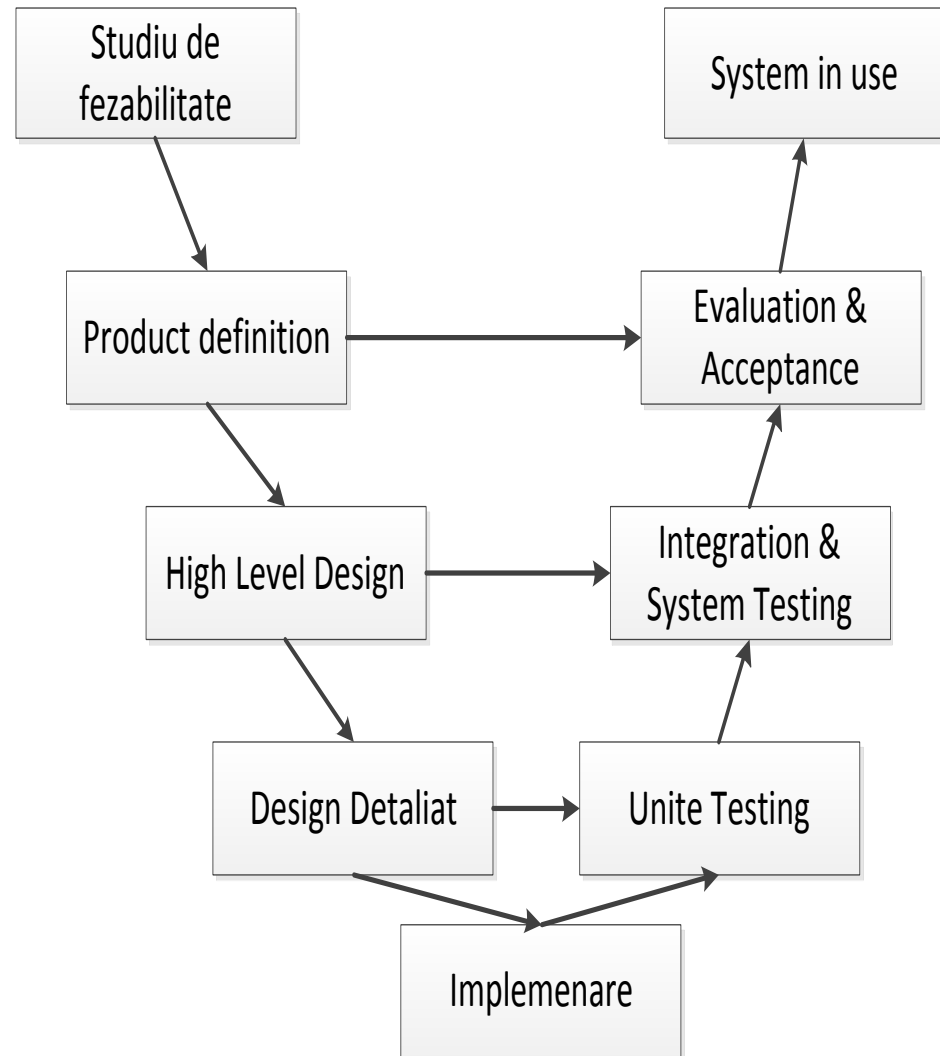
PRINCIPII

- **Daca un program nu face ce ar trebui sa faca e doar jumatate din efort; cealalta jumatate este ca programul nu face ce nu ar trebui sa faca**
- **Evitati testarea ad-hoc**
- **Nu planificati efortul pentru testare sub ipoteza tacita ca nu vor fi gasite erori**
- **Probabilitatea ca erori sa existe intr-o sectiune de program este proportionala cu numarul de erori deja gasite in acea sectiuni**
- **Testarea este un proces extrem de creativ si de dificil**

NIVELE DE TESTARE

- **Testarea are loc in intreg ciclul de viata al proiectului**

- Modul (Unit)
- Integrare & Sistem
- Evaluare & Acceptare (Acceptance)
- Instalare
- Regression



UNIT TESTING

- **Testare white-box intr-un mediu controlat a unui singur modul in izolare de celelalte**
 - Un “unit” este o singura functie sau o mica librarie
 - Suficient de mica cat sa poate fi testata cat mai complet
 - Testata in izolare fata de celelalte
 - Intr-un mediu de test controlat

INTEGRATION TESTING

- **Unitatile sunt combinate in module**
- **Focus pe interfata dintre unitati**
- **White Box si Black Box**

EXTERNAL FUNCTION TEST

- **Testare Black Box**
- **Verifica ca sistemul implementeaza corect functiile specificate**
- **Cunoscut si sub numele de test Alpha**
- **Echipa de test priveste sistemul din perspectiva utilizatorului final**

INTEGRATION TESTING

- **Modulele terminate si testate independent**
- **Sunt integrate toate in acelasi timp**
 - Repede si ieftin (fara stubs, drivers)
 - Erori descoperite tarziu si scump de reparat
- **Cea mai comuna abordare**

SYSTEM TEST

- **O versiune mai robusta a testelor externe**
- **Diferenta este platforma de test**
 - Pe hardware dedicat
 - Baze de date de dimensiuni apropiate celor din productie
 - Sistemul complet
 - Factori externi (hosting)
- **Poate testa precis cerintele nefunctionale (performanta, securitate, etc)**

ACCEPTANCE TESTING

- Cunoscut si sub numele de Beta testing
- Testare realizata de utilizatorii finali pe sistemul complet
- Cel mai realistic test
- Valideaza sistemul din perspectiva asteptarilor clientului
- Determina daca sistemule este pregatit pentru a intra in productie

INSTALLATION TESTING

- Testarea procesului de instalare/dezinstalare complet, partial, upgrade
- De obicei nu este deloc documentat

REGRESSION TESTING

- **Testeaza modificari ale SW**
- **Verifica ca schimbarile sunt corecte si nu afecteaza alte componente ale sistemului**
- **Selectia cazurilor de test care sunt considerate necesare pentru a valida modificarile**
- **Cu fixarea defectelor se pot intampla 4 lucruri:**
 - Bug reparat
 - Adaugat un bug nou
 - Stricarea structurii programului
 - Stricarea integritatii programului
- **Trei situatii din cele mai sus nu sunt dorite**

INTREBARI / RASPUNSURI

SOFTWARE METRICS

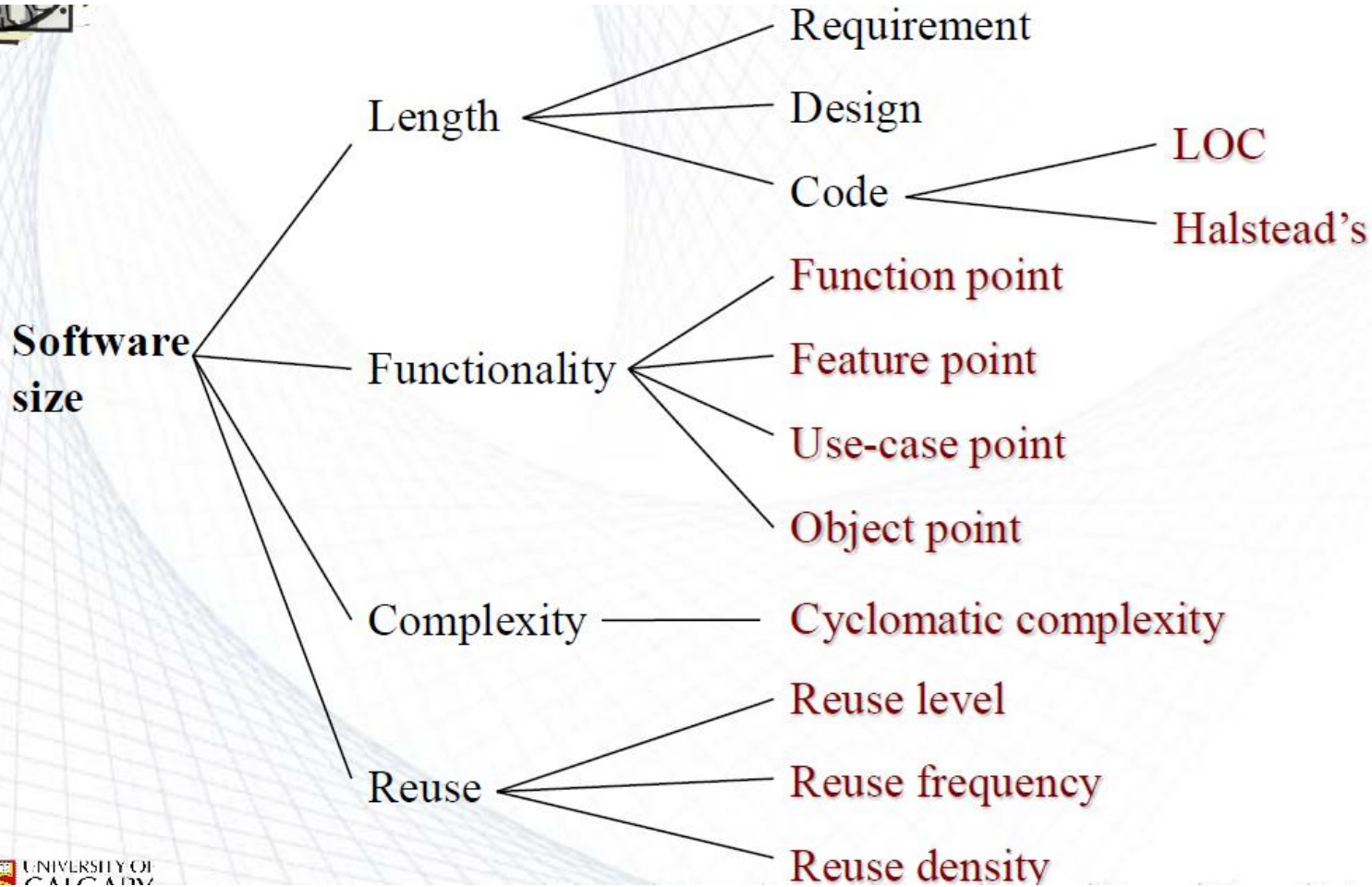
MASURAREA

- **De ce e importanta?**
 - Axioma management: “What gets measured gets done”
 - “Measure everything of significance - I swear this is true. Anything that is measured and watched, improves” (Fondatorul GoDaddy)
- **Masurarea atributelor interne: marime si structura (complexitate)**
- **Masurarea atributelor externe: calitate si reliability**
- **Predictii legate de proces: estimarea efortului, marimea, data de lansare (release)**
- **Masurarea in SE reprezinta selectarea unui subset de attribute (300+ definite deja) si interpretarea pentru a obtine o perspectiva completa asupra sistemului**

CINE BENEFICIAZA

- **Manageri**
 - Cat costa fiecare proces?
 - Cat de productiva este echipa?
 - Cat de bun este codul?
 - Va fi utilizatorul multumit?
- **Inginerii**
 - Sunt toate cerintele testabile?
 - Am gasit toate defectele?
 - Ce putem prezice despre SW in viitor?

SUMAR



LUNGIMEA

- **Unul dintre cei mai importanti indicatori ai complexitati este marimea fizica a sistemului**
- **Poate fi masurat static (fara executia sistemului)**
- **Trei faze traditionale in dezvoltarea de SW**
 - Specification
 - Design
 - Code
- **Lungimea specificatiei este corelata cu lungimea codului**

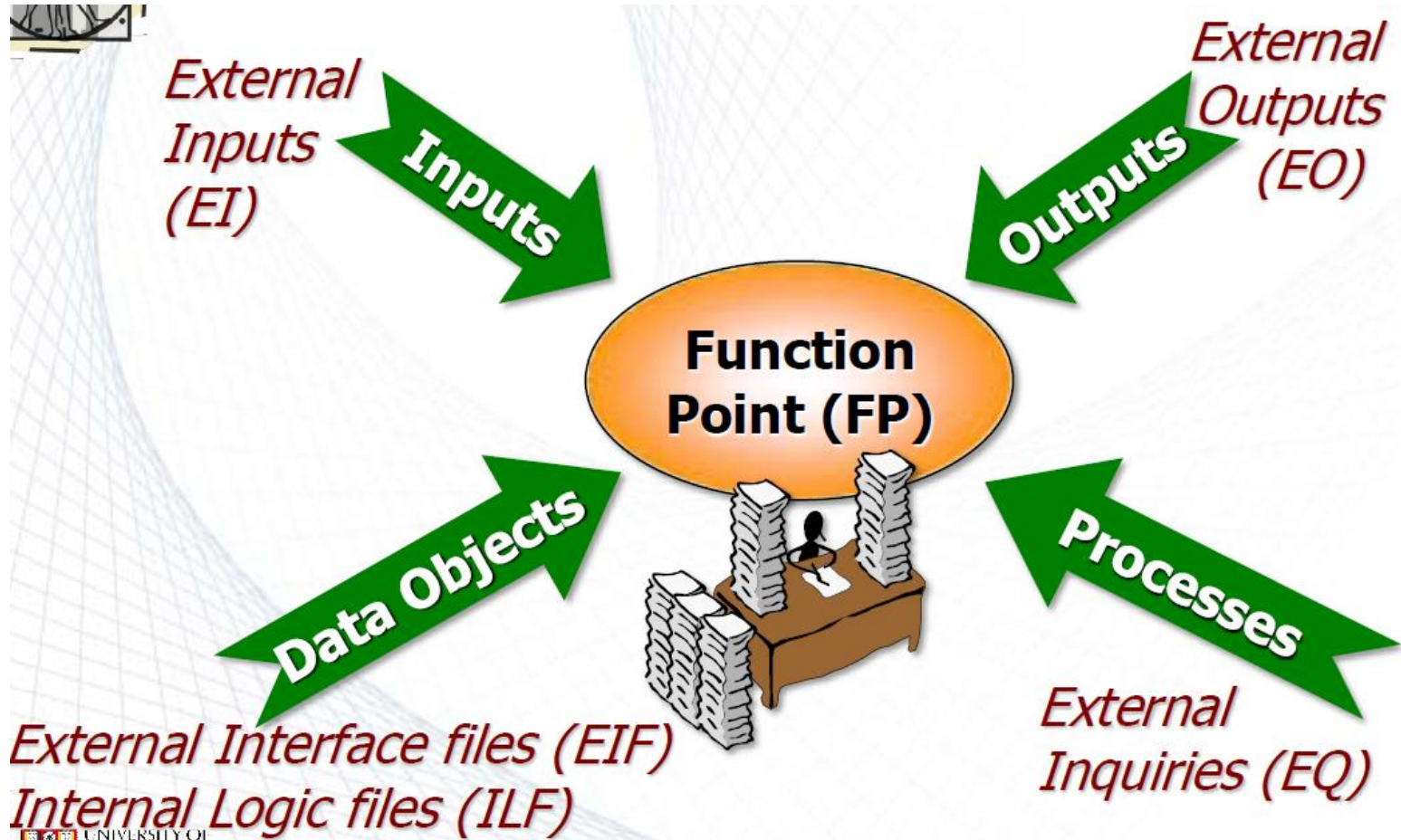
LUNGIMEA

- **Masurata in mod comun este numarul de linii de cod (LOC)**
 - Variatii: linii necomentate efective (ELOC); linii comentate (CLOC)
- **Avantaje**
 - Simplu de automatizat
 - Corelat puternic cu efortul si costul
- **Dezavantaje**
 - Definitie vaga
 - Depinde de limbaj
 - Incurajeaza programarea “sumo”

FUNCTIONALITATE

- **Function Points (FP) reprezinta o metrica balansata (vedem mai departe) a functionalitatii oferite de sistem**
- **Idee ar fi ca un sistem cu mai multa functionalitate este mai mare**
- **FP permit masurarea cantitatii de functionalitate din sistem pe baza specificatiilor => estimare inainte de implementare**
- **Inventat in 1979 la IBM, in 2004 a ajuns la versiunea 4.2, standardizat ISO**

FUNCTIONAL POINTS



FUNCTION POINTS

- **In esenta se refera la a numara toate interactiunile pe care SW le are cu mediul inconjurator**
 - External Inputs – date introduse de utilizatori
 - External Outputs – rapoarte care includ logica de procesare
 - External Inquiries – rapoarte care nu includ logica
 - Internal Logical Files – tabele in baze de date, configurari, fisiere
 - External Interface Files – servicii web
- **Se calculeaza in 2 pasi**
 - Calcularea **Unadjusted Function Point Count (UFC)**
 - Multiplicarea UFC cu **Value Adjustment Factor**
- **Rezultatul final (ajustat) este**
 - **$FP = UFC \times VAF$**

CONTROVERSA

- **Avantaje**

- Pot fi numarate inainte ca documentul de design sau codul sa existe
- Este standardizat
- Ajuta la negocierea contractelor
- Poate fi folosit pentru a estimare costuri, efort, planificari

- **Dezavantaje**

- Este subiectiva (afectata de VAF selectati de cei care numara)
- Necesita o specificatie absolut completa (foarte dificil de obtinut)
- Nu este adaptat inca pentru a numara i/o moderne: data streams, etc
- Necesita un efort foarte mare => nu se preteaza pentru sisteme foarte complexe

USE CASE POINTS

- **FP este o metoda de a masura dimensiunea sistemului din perspectiva specificatiilor**
- **Use case este o metoda de a crea specificatii**
- **Un use case este o secventa de evenimente care luate impreuna fac ca sistemul sa produca ceva “util”**
- **Sunt un instrument foarte puternic deoarece descriu fragmente relativ independente ale sistemului complet**

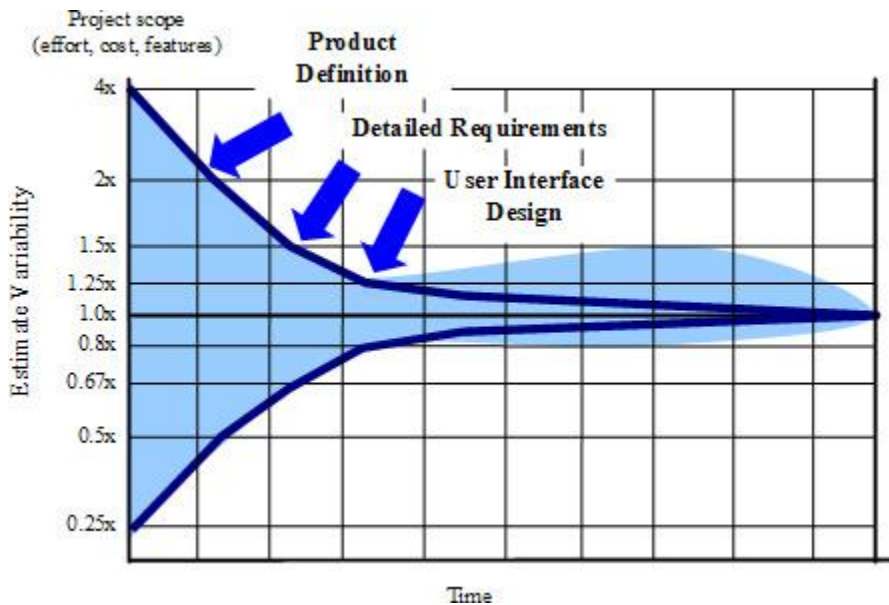
NUMERE FIBONACCI

- **Se folosesc numere Fibonacci (1, 2, 3, 5, 8, 13, 21) pentru a clasifica use case-urile ca si complexitateunul relativ la altul in tot proiectul.**
 - Cea mai simpla functionalitate este marcata cu 1 si cea mai dificila cu 21
- **Procesul este foarte simplu de implementat**
 - O lista a scenariilor este mentinuta per proiect si agregata la nivel de companie
 - Proiectul este impartit in interatii si un numar de scenarii va fi inclus intr-o iteratie
 - La sfarsitul proiectului se calculeaza numarul de puncte/saptamana
 - Totalul este agregat la nivel de companie

NUMERE FIBONACCI

- **Se aplica legea numerelor mari. Cand s-au acumulat date istorice multe aberatiile se balanseaza**
- **Numarul de puncte poate fi folosita ca metrica (KPI) in firma pentru a masura imbunatatiri in viteza de dezvoltare si productivitate**
- **In timp, dupa colectarea datelor, punctele se pot traduce in zile de munca**

CONUL INCERTITUDINII



- Conul incertitudinii are cateva ramificatii
- Prima este ca estimarile date la inceputul proiectului vor fi foarte inexacte
- Estimările date în faza de concept pot fi inexacte cu un factor de 4x în sus, sau 1/4x în jos

IN PRACTICA

- **Aplicati descompunerea**
- **Estimati atat crearea unui plan de test cat si executarea planului de test**
- **Estimati minimum 2 runde complete de test**
- **Luati in calcul ca prima runda de test este mai lunga deoarece aveti nevoie de timp sa introduceti toate bugurile**
- **Niciodata nu estimati mai putin de 20% din timpul alocat dezvoltarii**

INTREBARI / RASPUNSURI

STANDARDE

STANDARDE

- **Suita de standarde ISO 9000 pentru managementul calitatii**
- **ISO 9126 - Standard pentru masurarea calitatii software**
 - ISO/IEC 9126-2:2003 Software engineering product quality
 - Partea I - Modelul de calitate
 - ISO/IEC TR 9126-2:2003 Software engineering product quality
 - Partea II: Metrici externe
 - ISO/IEC TR 9126-3:2003 Software engineering product quality
 - Partea III: Metrici interne
 - ISO/IEC TR 9126-4:2004 Software engineering Product quality
 - Partea IV: Quality in use metrics

CARACTERISTICI ALE CALITATII SOFTWARE

- **ISO 9126**
 - “A set of attributes of a software product by which its quality is described and evaluated. A software quality characteristic may be refined into in multiple levels of sub—characteristics.”
- **ISO 9126 este un standard de evaluare a produselor software.**
- **Defineste 6 caracteristici care descriu, cu suprapunere minima, calitatea software**
 - Functionality
 - Reliability
 - Usability
 - Efficiency
 - Maintainability
 - Portability

CARACTERISTICI

- **Functionality**
 - Un set de attribute legate de existenta unui set de functii. Aceste functii satisfac cerinte explicite sau implicite.
- **Reliability**
 - Un set de attribute legate de capacitatea sistemului de a mentine un nivel de performanta in anumite conditii si pentru o anumita durata de timp
- **Usability**
 - Un set de attribute legate de efortul necesar pentru utilizarea produsului de catre un set explicit sau implicit de utilizatori

CARACTERISTICI

- **Efficiency**

- Un set de attribute legate de relatia dintre nivelul de performanta al sistemului si cantitatea de resurse utilizata

- **Maintainability**

- Un set de attribute legate de efortul necesar pentru a aduce modificari

- **Portability**

- Un set de attribute legate de abilitatea sistemului de a fi transferat dintr-un mediu in altul

PERSPECTIVE - UTILIZATORII

- **Utilizatorii sunt interesati de utilizarea software-ului, performantele si efectele**
- **Evalueaza SW fara a cunoaste cum a fost construit**
- **Intrebari**
 - Functiile necesare sunt disponibile in sistem?
 - Cat de stabil este sistemul?
 - Cat de eficient este?
 - Cat de usor este de folosit?
 - Cat de usor este de schimbat?

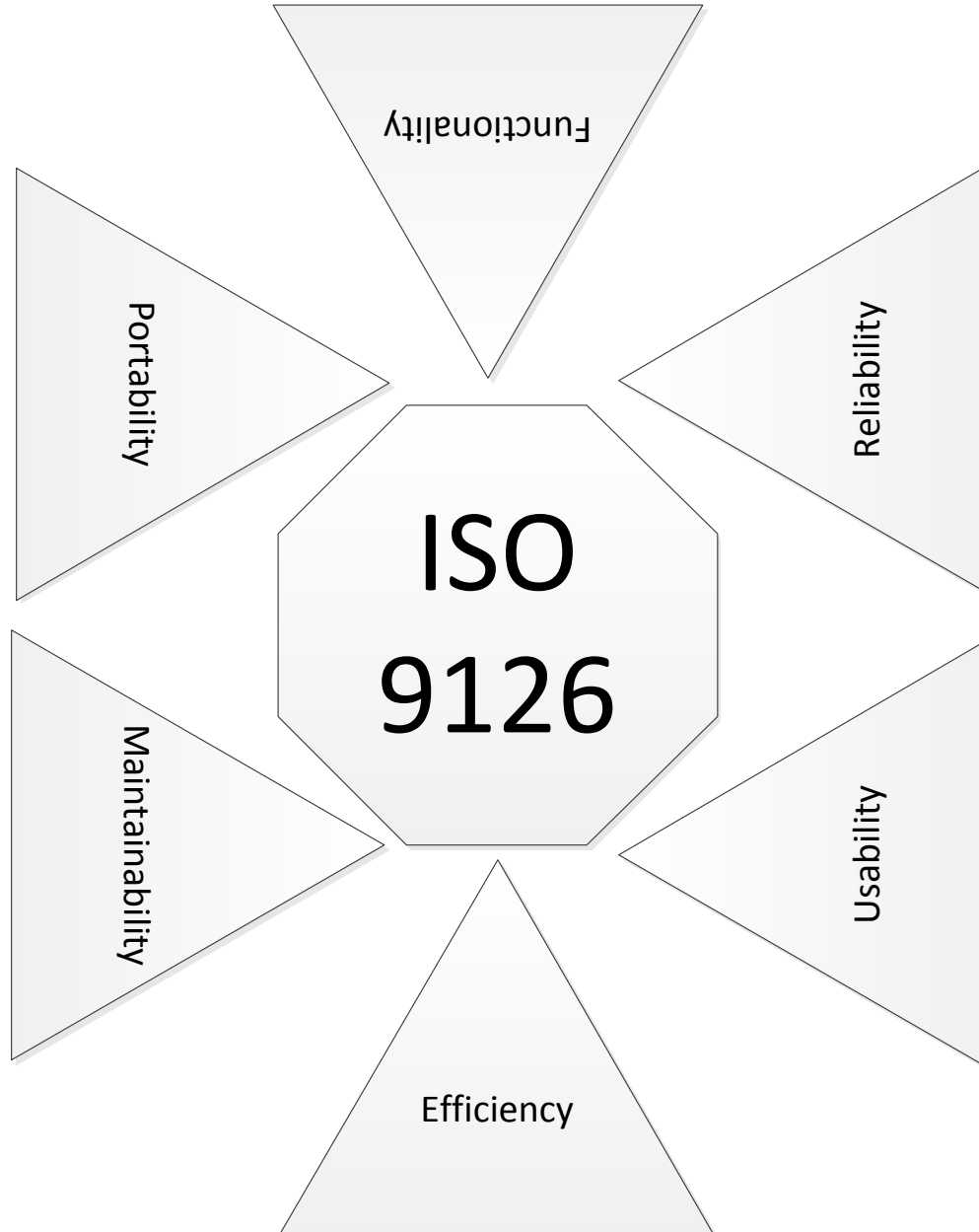
PERSPECTIVE - DEVELOPER

- **Procesul de dezvoltare necesita ca programatorii si utilizatorii sa aplice aceleasi caracteristici de calitate. Ele vor fi aplicate si la “acceptance”**
- **Dezvoltatorii trebuie sa fie interesati atat de calitatea produsului final cat si de calitatea produselor intermediare din diferite faze ale dezvoltarii**
- **Pentru evaluarea calitatii produselor intermediare diferite metrice sunt aplicate**

PERSPECTIVE – MANAGERI

- **Managerii sunt in general interesati de calitatea “per ansamblu” mai mult decat de o anumita caracteristica**
- **Managerii sunt nevoiti sa balanseze cerintele de calitate cu alti parametri legati de management cum ar fi intarzieri in executie sau depasiri de buget**
- **Challenge: atingerea cerintelor de calitate in limitele impuse de costurilor, resurselor umane si timp**

MODELUL DE CALITATE



ISO 9126: 1. FUNCTIONALITY

- **Suitability:** Atribute care descriu prezenta si corectitudinea unui set de functii pentru un task specific
- **Accuracy:** Atribute ce descriu corectitudinea rezultatelor si efectelor
- **Interoperability:** Atribute care descriu abilitatea sistemului de a interactiona cu alte sisteme
- **Security:** Atribute care descriu abilitatea sistemului de a preveni accesul neautorizat, accidental sau intentionat, la date

ISO 9126: 2. RELIABILITY

- **Maturity: Atribute care descriu frecventa cu care sistemul pica datorita defectelor din SW**
- **Fault tolerance: Atribute care descriu abilitatea de a mentine nivelul specificat de performante in cazul defectelor**
- **Crash frequency: Numarul de crash-uri ale sistemului pe unitatea de timp**
- **Recoverability: Atribute ale sistemului legate de capacitatea de a restabili nivelul de performanta specificat si datele afectate, cat si timpul si efortul necesar pentru a face acest lucru**

ISO 9126: 3. USABILITY

- **Understandability:** Atribute care descriu efortul pe care utilizatorii trebuie sa il faca pentru a intelege conceptele logice ale aplicatiei
- **Learnability:** Atribute care descriu efortul utilizatorilor pentru a invata aplicatia (ex. operatiuni de control, input, output)
- **Operability:** Atribute care descriu efortul utilizatorilor pentru operarea sistemului

ISO 9126: 4. EFFICIENCY

- **In ce masura sistemul poate opera folosind cantitatea minima de resurse**
- **Time behavior: Atribute care descriu timpul de procesare si raspuns cat si rata de throughput in operare**
- **Resource behavior: Atribute care descriu cantitatea de resurse utilizate cat si durata utilizarii acestor resurse in functionare**

ISO 9126: 5. MAINTAINABILITY

- **Analyzability:** Atribute care descriu efortul necesar pentru a diagnostica deficiente sau cauze ale erorilor, si a identifica parti ce trebuie modificate
- **Changeability:** Atribute ale SW ce descriu efortul necesar pentru modificarea SW sau a mediului de executie
- **Stability:** Atribute ce descriu riscul efectelor neasteptate sau modificarilor neasteptate
- **Testability:** Atribute ce descriu efortul necesar pentru validarea sistemului modificat

ISO 9126: 6. PORTABILITY

- **Adaptability:** Atribute ce descriu adaptabilitatea SW la alte medii sau scenarii fara a aplica alte modificari in afara celor desemnate special pentru acest scop (ex. Fisiere de configurare)
- **Installability:** Atribute ce descriu efortul necesar pentru instalarea SW intr-un mediu
- **Conformance:** Atribute ce descriu conformarea la standarde si conventii legate de portabilitate
- **Replaceability:** Atribute ce descriu efortul necesar pentru utilizarea SW in locul unui alt sistem

TOTAL QUALITY MANAGEMENT

TQM: CUSTOMER SATISFACTION

- **Studiile au aratat ca:**
 - Este de 5 ori mai costisitor sa “recrutezi” un nou client decat sa pastrezi un client existent
 - Clientii nesatisfacuti spun la 7 pana la 20 de persoane
 - Clientii satisfacuti spun la 3 pana la 5 persoane
- **TQM are ca scop dezvoltarea durabila a afacerilor de succes prin corelarea calitatii cu satisfactia clientului**
- **O data cu cresterea competitiei globale “customer focus” este singura modalitate de a pastra clientii si a creste cota de piata**
- **... mai multe saptamana viitoare**

CURSUL 2

SAPTAMANA VIITOARE

- **Whitebox si blackbox testing**
- **Cum se scrie un plan de test**
- **Cum se raporteaza defecte**
- **Invitat**
 - Ana Figher, QA Manager, Embarcadero

VA

MULTUMESC!