

Ingineria Sistemelor de Programare

Introducere in Ingineria Software

Mihai Hulea

2020

Introducere

- mihai.hulea@aut.utcluj.ro
- Nota finala: 50% Laborator + 25% Partial + 25% Examen final
- Resurse interne:
 - <http://178.128.193.0/doku.php>
 - <https://piazza.com/utcluj.ro/spring2020/seriab>
 - <https://github.com/isp-cluj/examples-course-cluj-2020>
- Carti:
 - [Data Structures and Algorithms in Java Fourth Edition.pdf](#)
 - [Think Java](#)
 - [Javanotes 8.1](#)
 - [Thinking in Java 3th](#)
- Unele folosite:
 - Java 8 / Java 9
 - IDE Eclipse \ [IntelliJ](#) \ Netbeans
 - StarUML 2.x
 - Git client
 - Maven 3.x

Scop curs

Cursul va fi orientat pe aplicatii in care conceptele vor fi ilustrate prin aplicatii practice.

Competente dobandite:

- Dezvoltare de aplicatii Java standad (Java SE)
- Aplicarea tehnicilor de programare OOP
- Utilizarea conceptului de testare a unitatilor pentru verificarea functionarii programelor
- Utilizarea principiilor de proiectare si sabloanelor de proiectare
- Proiectarea aplicatiilor utilizand UML

Cuvinte cheie:

Java SE, OOP, Design Patterns, UML, Unit testing

Ingineria Software

- Ce este ingineria software (eng. Software Engineering) si de ce este ea importanta ?
- Care sunt procesele si metodele care definesc ingineria software

Ingineria Software

- Ce este ingineria software ?

Aplicarea tehnicilor ingineriei intr-un mod sistematic pentru implementarea sistemelor informatice.



...the application of a **systematic**, **disciplined**, and **quantifiable** approach to the **development**, **operation**, and **maintenance** of software; that is, the application of engineering to software...

- IEEE

*“Where **computer science** is about taking complex problems and deriving a solution from mathematics, science and computational theory, **software engineering** is very much focused around designing, developing and documenting beautiful, complete, user-friendly software.”*

David Budden

Ingineria Software

Process

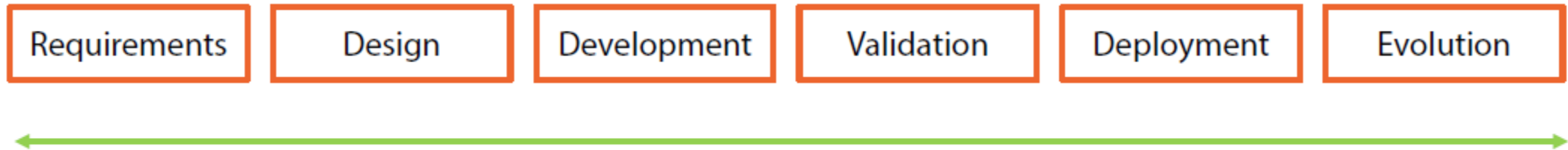
Methods

Tools

- Proces :
 - Definirea unui cadru si a unei liste de **activitati**
- Metode :
 - Definirea de **tehnici** pentru efectuarea activitatilor
- Unelte :
 - **Unelte** ce ajuta in aplicarea proceselor si metodelor

Activitati

Activitatile de dezvoltare software

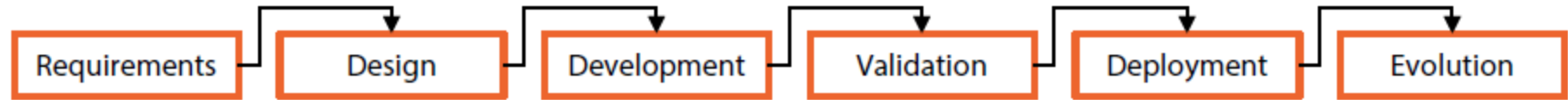


Activitati umbrela

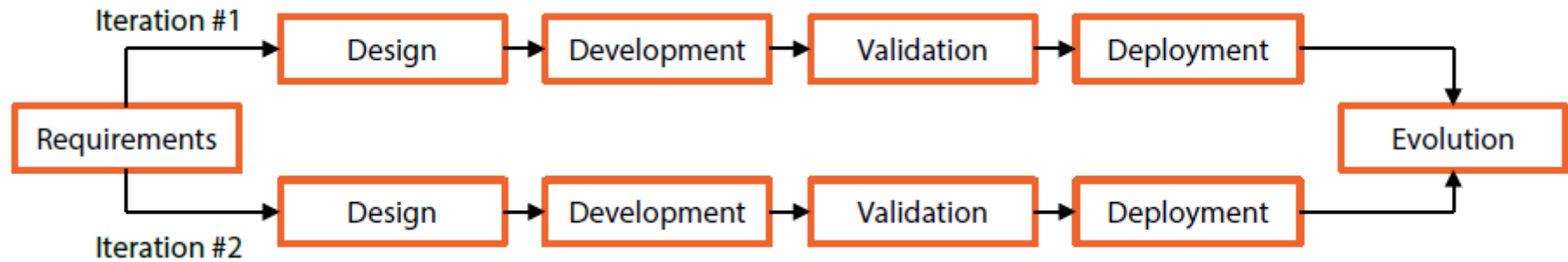


Procese

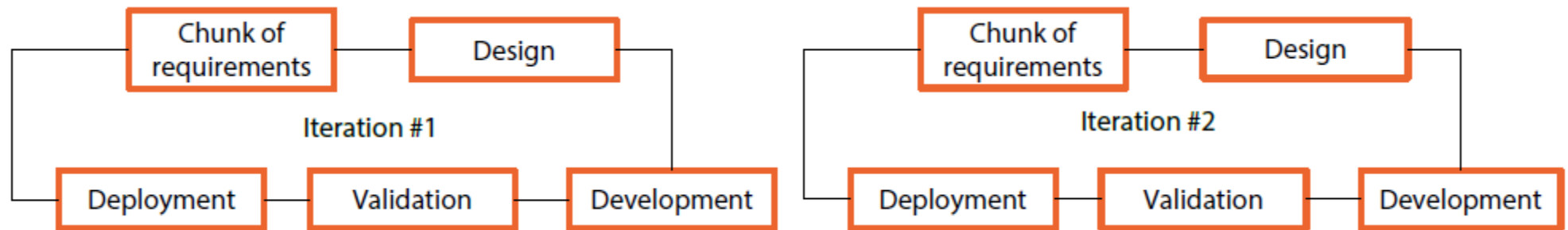
Model liniar



Model iterativ



Model agil



Artefacte



Modele de aplicare a proceselor software

- Modelul clasic
 - *Waterfall (linear)*
- Modele interactive si incrementale
 - *Prototyping*
 - *Spiral*
 - *Agile*
 - *Unified Model*
- Modele specializate
 - *Component-based development*
 - *Formal methods*
 - *Aspect-oriented development*

Ce model sa aplicam ?

- Care este tipul proiectului;
- Care este maturitatea proiectului;



Air Control Traffic System

vs



Blog app

Unelte software

- Analiza cerintelor
 - Enterprise Architect, IBM Rational
- Dezvoltare
 - Visual Studio, NetBeans, IntelliJ
- Testare
 - HP Quality Center, Sonar
- Configurare software
 - Jenkins, Puppet, Chef
- Managementul proiectelor
 - Microsoft Project, Jira



Ecosistemul Java



An example from **Dynamic Software** (http://www.dynamiquesoftware.com/Dynamique_Software_Enterprise_Java.php)

Rolul inginerului software

Un rol mai puțin definit formal

- Software Engineer vs Software Developer
- Software Engineering nu înseamnă doar implementarea (dezvoltarea) de software
 - Este un tester un inginer software ?

In general, oricine aplică tehnici de inginerie în analiza, proiectarea, implementarea, testarea sau mentenanța unui sistem informatic este un inginer software.

Ingineria software si managementul proiectelor

- Managementul proiectelor este o activitate umbrela
- Ingineria software se aplica pe toata durata existentei unui proiect

