



Co-funded by the
Erasmus+ Programme
of the European Union



SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI"
EST. 1888



ICT-TEX course on Digital skills

Topic 6: Introduction to Software Engineering

The course is developed under Erasmus+ Program Key Action 2:
Cooperation for innovation and the exchange of good practices [Knowledge Alliance](#)

ICT IN TEXTILE AND CLOTHING HIGHER EDUCATION AND BUSINESS

Project Nr. 612248-EPP-1-2019-1-BG-EPPKA2-KA

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.



Co-funded by the
Erasmus+ Programme
of the European Union



SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI"
EST. 1888



6.1. Software Engineering



These slides are part of the topic on
“Topic 6: Introduction to Software Engineering” of the
course on Digital skills in Textile and clothing industry.

Check also the other themes in this topic:

- 6.2. Requirements Engineering
- 6.3. Introduction to modeling and UML
- 6.4. UML Diagrams



Contents

1. Introduction to Software Engineering

- [Software Engineering](#)
- [Software](#)
- [Software processes](#)
- [Software development processes](#)
- [Software engineering method – definition](#)
- [Software process model](#)
- [Software process models examples](#)



Software Engineering

- Definition of Software Engineering

“Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.”

- Sommerville, I. *Software Engineering*. 10th edition, Published by Pearson Education, ISBN: 978-1-292-09613-1, 2016, pp. 21



Software Engineering

- Two important aspects of the definition of Software Engineering
 - **Software** (definition, types of software products, application in different areas, reuse, importance and so on.)
 - **Engineering** (the process of production of the software, including additional activities such as the development of tools, theories, and means to support the software development process; risk project management, and so on.)



Software Engineering

- **Software Engineering** represents a systematic approach to the development of **high quality software**, as well as to its **marketing, operation and maintenance**.
- **Software Engineering** is a discipline that deals with all aspects of the design and development of **high-quality** software.



Software

Definition of Software

“Software is: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately manipulate information, and (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.”

- *Pressman, R., Maxim, B. Software Engineering: A Practitioner's Approach. 9th edition, Published by McGraw-Hill Education, ISBN: 9781260548006, 2019, pp.5*



Application of software

The classification by R. Pressman, presents the *application of software as*:

- System software
- Application software
- Engineering/scientific software
- Embedded software
- Product-line software
- Web/mobile applications
- Artificial intelligence software

Pressman, R., Maxim, B. Software Engineering: A Practitioner's Approach. 9th edition, Published by McGraw-Hill Education, ISBN: 9781260548006, 2019, pp.7



Software as a product

The *software* that is intended to be sold (to derive financial benefits and so on) is called a **(software) product**.

Software products can be two main types:

- Generic software products
- Customized software products



Essential attributes of good software

The product characteristics of good software can be:

Acceptability

Dependability and security

Efficiency

Maintainability



Software processes

- *“Software Process as a framework for the activities, actions, and tasks required to build high-quality software.”*
- *Software Processes* describe and control the life cycle of a software product.
- The software product life cycle can include:
 - The Beginning. The emergence of the idea of creating a product and
 - The End. The moment when its use is discontinued.



Main activities at every software process

Specification - what the software should do and what its scope is.

Design and implementation - writing code, programming the software product is produced according to the specifications. It can also be called *Development process*.

Validation - check if the software is what the user wants.

Evolution - a change in the software as a result of changes in the requirements for it.



Software development processes

Software Development Processes means the set of activities related to the development of the software product from the beginning to the end of its life cycle.

Software Development Process is the entire process of setting the task, planning, implementation, and evaluation of a software and hardware application, including the tools used, methods, and necessary staff



Software engineering process framework

A Generic Process framework for Software Engineering - applicable to all software projects (no matter size or complexity)

Includes main Umbrella and framework process activities:

Communication

Planning

Modeling

Construction

Deployment



Software engineering process – Umbrella Activities

- The *Umbrella Activities* applicable during the development process of software projects
- Support the management, quality assurance, and risk management.

Umbrella Activities classification by Pressman, R., Maxim, B. Software Engineering: A Practitioner's Approach. 9th edition, Published by McGraw-Hill Education, ISBN: 9781260548006, 2019, pp.11



Software engineering process – Umbrella

Activities

The main *Umbrella Activities* include:

- Software project tracking and control.
- Risk management
- Software quality assurance
- Technical reviews
- Measurement.
- Software configuration management
- Reusability management
- Work product preparation and production.



Software engineering method – definition

- *The method of software engineering* is a structured approach to software development that aims to facilitate the creation of cost-effective high-quality software.



Software engineering methods

- Software engineering methods include:
 - **Model description** - describes models of the software to be developed. For example: (Data flow model; Structural model; Object model; Incremental and Iterative model; Prototyping model, etc.)
 - **Rules** - restrictions applicable to system models
 - **Recommendations** - tips for good design practices.
 - **Description of the process** - what activities should be performed and in what order.



Software process model

- The abstract presentation of the software process from a certain perspective represents the **Model of the Software Process**.
- *A “Software Process Model is a simplified representation of a software process. Each process model represents a process from a particular perspective and thus only provides partial information about that process”*

- Sommerville, I. *Software Engineering*. 10th edition, Published by Pearson Education, ISBN: 978-1-292-09613-1, 2016, pp. 45



Software process model description

Description of Software Process Models

- Process flow – a description of the sequence of activities, actions, and tasks that are performed concerning sequence and time: (Linear process flow, Iterative process flow, Evolutionary process flow, Parallel process flow)
- Workflow - as a sequence of activities that are performed
- Data flow - the flow of information between the various stages in the process is considered
- Role/action - who does what.



Software process models examples

Waterfall Model

Evolutionary models - Incremental and Iterative Development, Prototype models, Spiral Model

V Model

Agile - Scrum, Extreme Programming - XP, Kanban, Feature-Driven Development (FDD), Dynamic systems development method (DSDM) and so on.

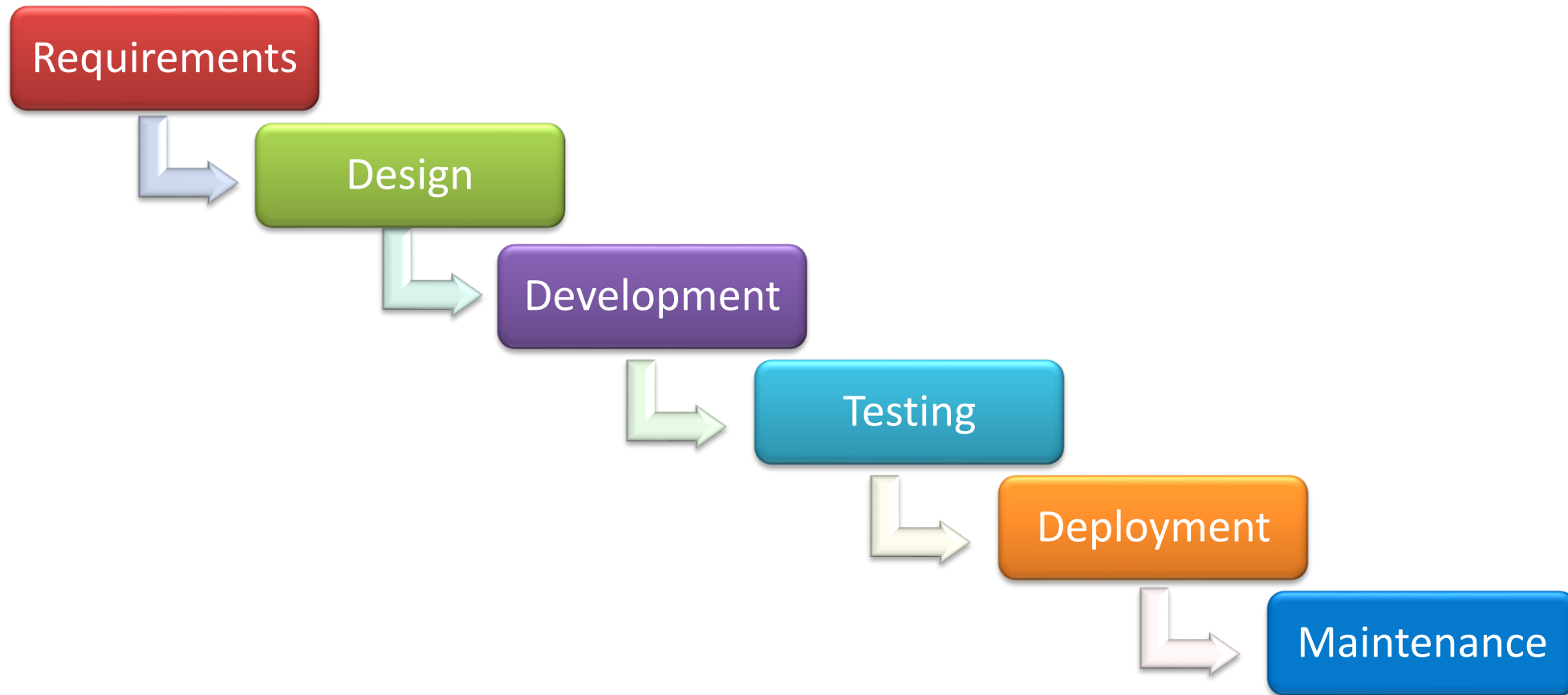


Waterfall model

- The oldest method of software development is based on the so-called Waterfall model. It consists of a sequence of steps (stages), each stage having to be completed before the next one starts. At the end of each stage, as a result, we have certain documents, data, code, and other artifacts.



Waterfall model





Waterfall model stages

The stages in the Waterfall model are as follows:

- Requirements
- Design
- Development
- Testing
- Deployment
- Maintenance



Waterfall model stages

Requirements

- What the software should do and what its scope is. Intensive communication with the client and stakeholders. The specification must be elaborated in detail and focused on the hardware, software, requirements of the contracting authority, and with priority to implementation. At the end of this stage, the *Software Requirement Specification document (SRS)* is prepared.



Waterfall model stages

Design stage

- The design creates a detailed product design, models at different levels of abstraction, and different parts or aspects of the project. As a result, a description of the software architecture is used, various system components and connections between them are identified, the necessary hardware and software resources for goal setting, etc. A schedule for the implementation of the project is made, they are distributed and the edition of their implementation is determined. The risk and the necessary resources (time, people, equipment, money) are assessed



Waterfall model stages

Development stage

- At this stage, the coding of the model in selected programming languages is used.

Testing

- This is a process of checking the written code. Testing can be performed on modules as well as on the whole system. The process is very long and accompanies implementation and integration. Testing is a key step in correcting errors and omissions. Thanks to testing, it is possible to adapt the entire system to the specification. When errors are detected - return to the previous stages.



Waterfall model stages

Maintenance stage

- At this stage, the components of the system are assembled into a new system and its properties are revealed. It is mandatory to test the new system by the developer and the contracting authority - whether it meets the specification. Upon successful completion of the tests, the new system reaches its end users and starts working.



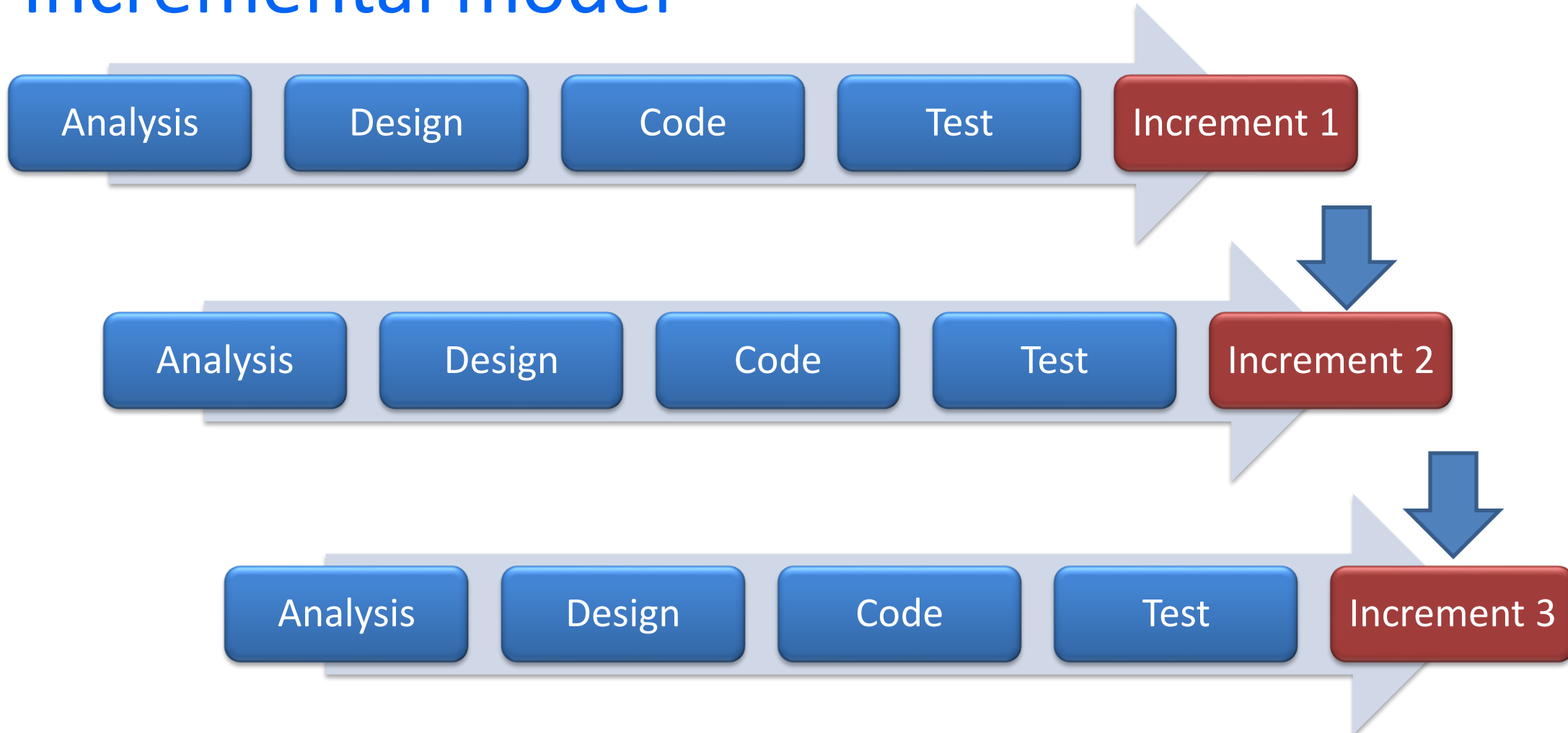
Waterfall model stages

Deployment stage

- This stage represents the activity of making changes and improvements in integrated software by establishing good communication with the client and the users of the system. The modification of the software is in a corrective, adaptive direction and with subsequent improvement of the product, by improving existing or adding new functionalities.

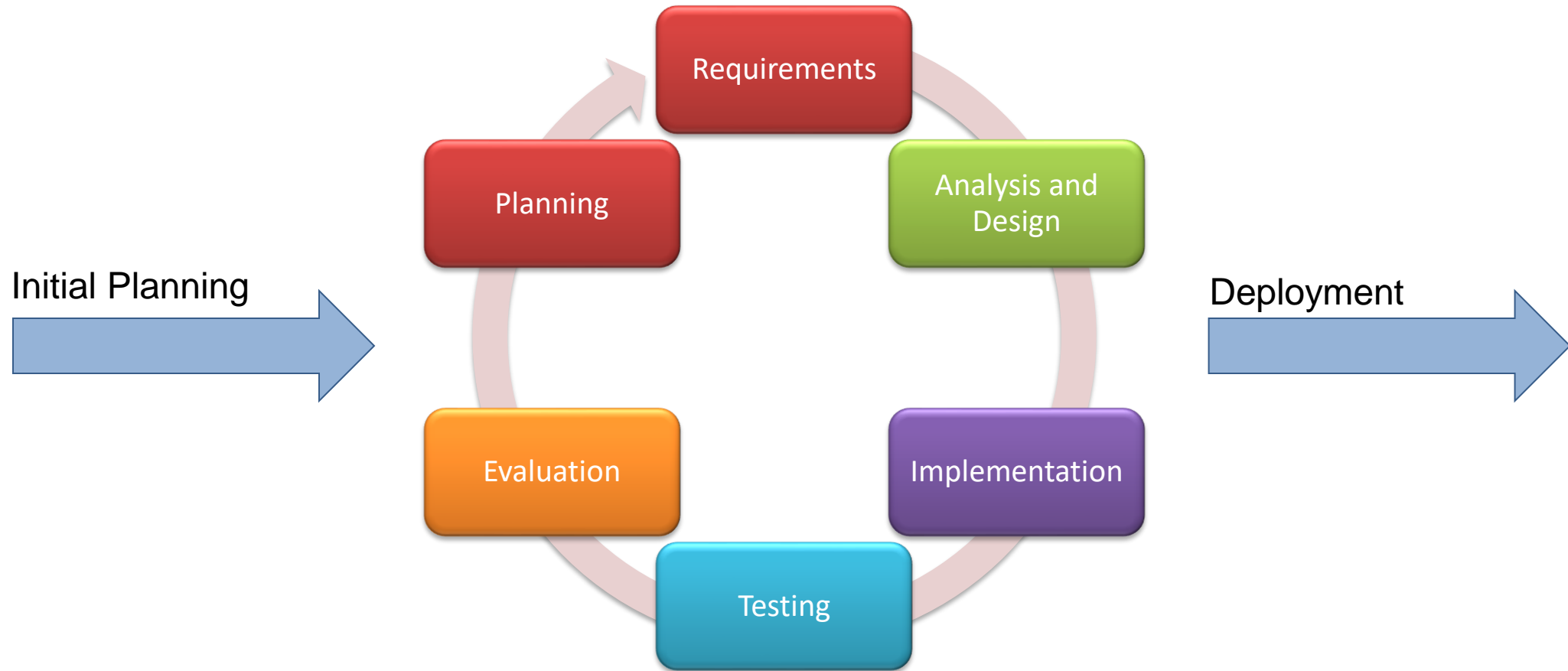


Incremental model





Iterative model





Agile model – 12 Principles





Agile models

Learn more about:	At Web page:
Scrum	https://www.scrum.org/resources/what-is-scrum
Scaled Agile Framework (SAFe)	https://www.atlassian.com/agile/agile-at-scale/what-is-safe
Dynamic Systems Development technique (DSDM)	https://www.geeksforgeeks.org/dynamic-systems-development-method-dsdm/
Crystal	https://airfocus.com/glossary/what-is-the-crystal-agile-framework/



Agile models

Learn more about:	At Web page:
Extreme Programming (XP)	https://ronjeffries.com/xprog/what-is-extreme-programming/
Kanban	https://kanbanize.com/kanban-resources/getting-started/what-is-kanban
Lean	https://medium.com/kayvan-kaseb/using-lean-in-software-development-1b01bbb98d6e
Feature-Driven Development (FDD)	https://www.lucidchart.com/blog/why-use-feature-driven-development



References

- Sommerville, I. Software Engineering. 10th edition, Published by Pearson Education, ISBN: 978-1-292-09613-1, (2016)
- Pressman, R., Maxim, B. Software Engineering: A Practitioner's Approach. 9th edition, Published by McGraw-Hill Education, ISBN: 9781260548006, (2019)
- Dick J., Hull E., Jackson K. Introduction. In: Requirements Engineering. Springer, Cham. https://doi.org/10.1007/978-3-319-61073-3_1, ISBN: 978-3-319-61073-3, (2017)

CONTACTS

Coordinator:

Technical University of Sofia

Project coordinator:

assoc. prof. Angel Terziev, PhD
aterziev@tu-sofia.bg

Web-site: ICT-TEX.eu

Author:

Assistant professor Yavor Dankov
Sofia University "St. Kliment Ohridski"

Email: yavor.dankov@fmi.uni-sofia.bg

ResearchGate: <https://www.researchgate.net/profile/Yavor-Dankov>

Scopus: <https://www.scopus.com/authid/detail.uri?authorId=57202891597>



Co-funded by the
Erasmus+ Programme
of the European Union

KNOWLEDGE ALLIANCE

ICT-TEX

ICT IN TEXTILE AND CLOTHING
HIGHER EDUCATION AND BUSINESS

These slides and the materials included in these slides (including references) are for educational purposes only. The use of slides should be done with correct citation and only for educational purposes.

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.