tremend

UNIVERSITY OF BUCHAREST

VIRTUTE ET SAPIENTIA

Course 1 - Introduction in Software Engineering

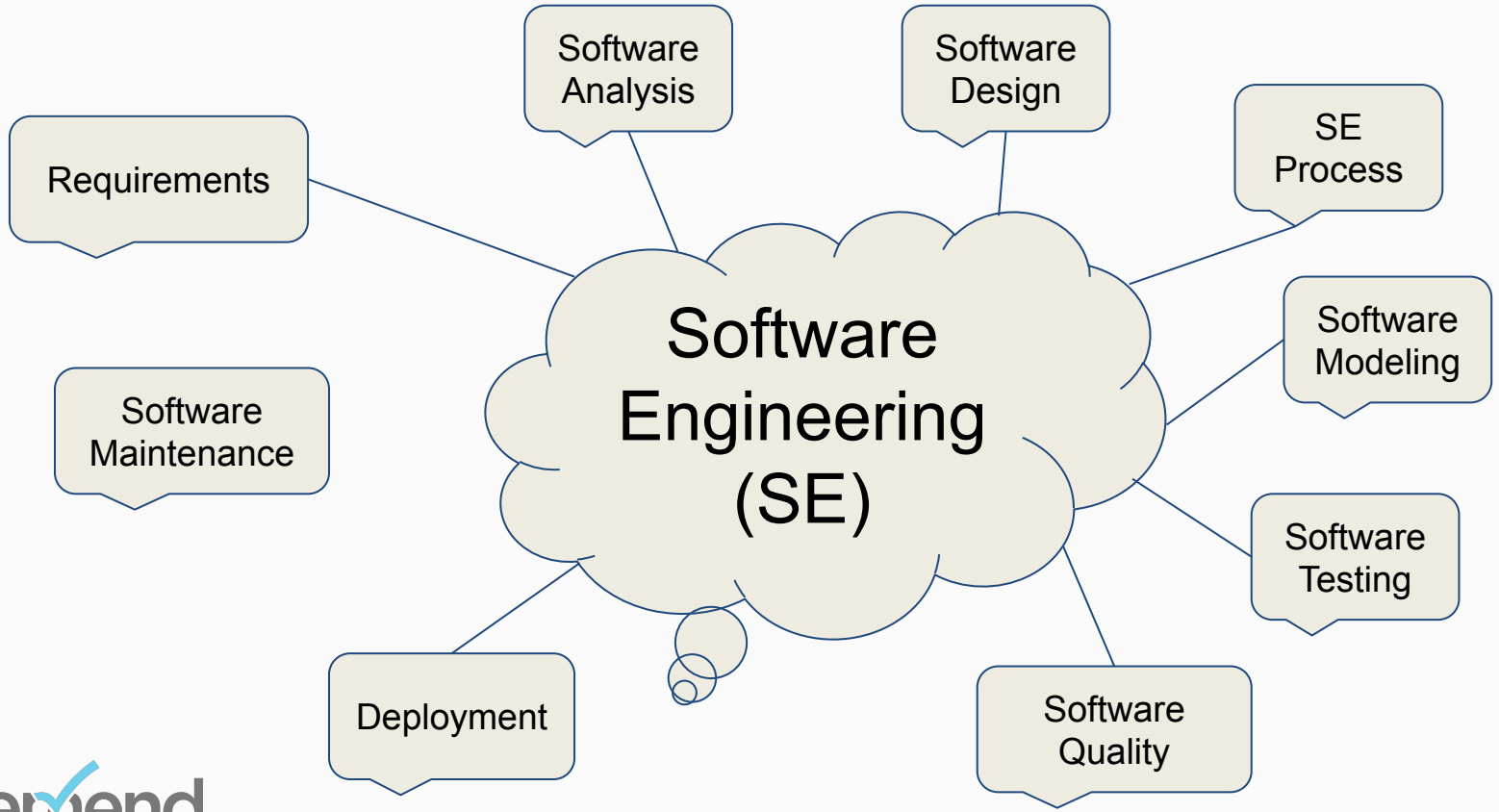
Inginerie Software

An universitar 2022 - 2023

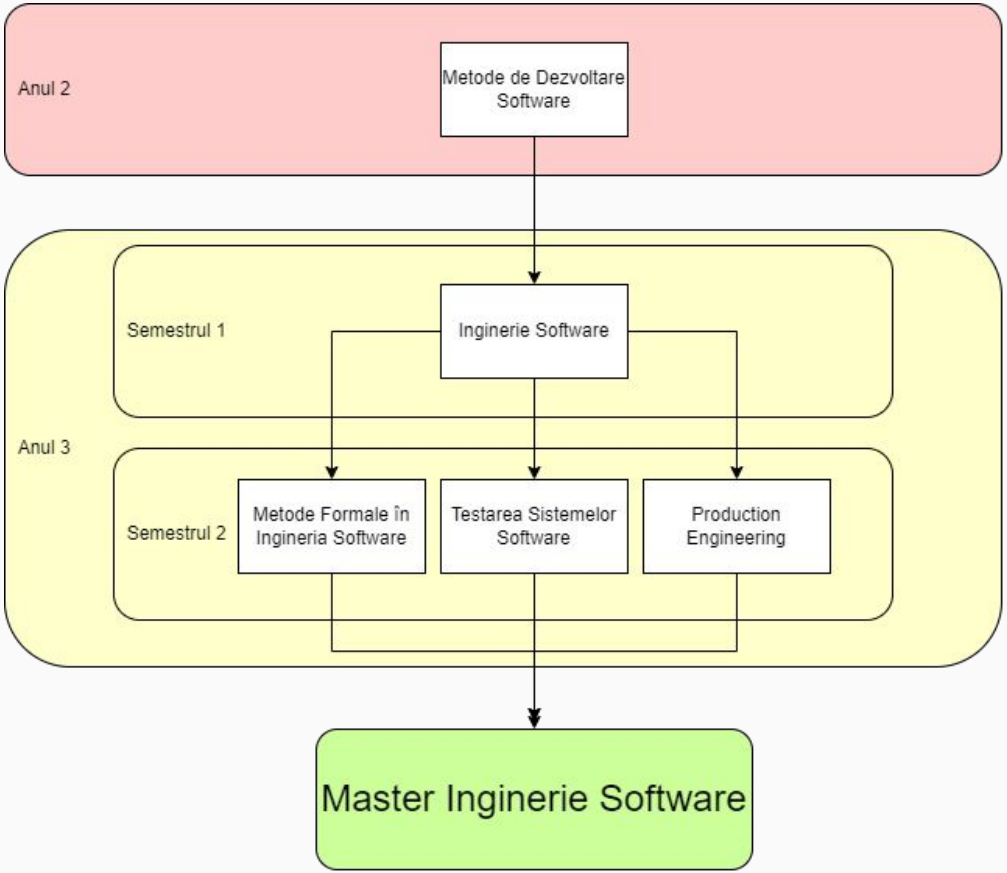
Agenda

1. *What is Software Engineering?*
2. *Ce vom invata*
   a. *Project definition and characteristics*
   b. *Product definition and attributes*
   c. *Software Development Lifecycle Overview*
3. *Organizarea didactica*
4. *Evaluarea*
5. *Q&A*

What is Software Engineering?

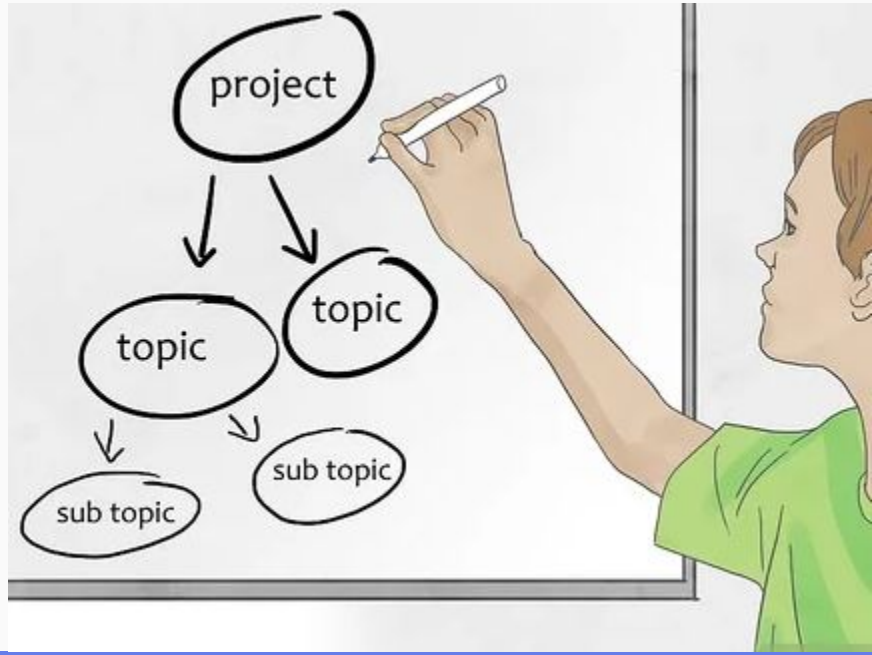


Software Engineering roadmap @ FMI



Project definition and characteristics

A project is defined as a specific, **finite activity** that produces an observable and measurable result under certain **preset requirements**. It is an attempt to implement desired change to an environment in a controlled way.



Project definition and characteristics

A project is a **set of tasks** which must be completed in order to arrive at a particular **goal or outcome**.

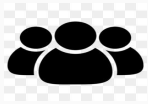
Depending on the size and scope of the project, these tasks may be simple or elaborate, but all projects can be **broken down into objectives** and what needs to be done to achieve them.

According to the **PMBOK®** Guide—Fourth edition (*PMI, 2008a, p. 434*) the definition of a project is "**a temporary** endeavor undertaken to create **a unique** project service or result."

Projects are **temporary** and **close down on the completion** of the work they were chartered to deliver.

Lifecycle

Scope



Stakeholders

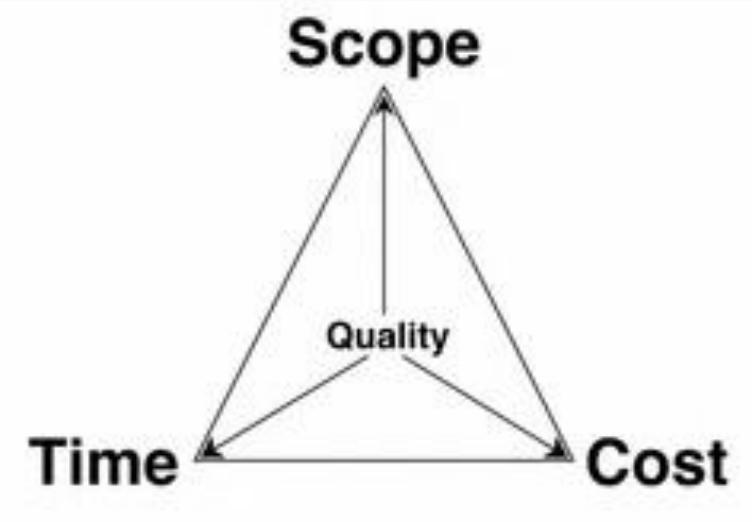
Deliverables

Milestones

Resources

Dependencies

Project definition and characteristics

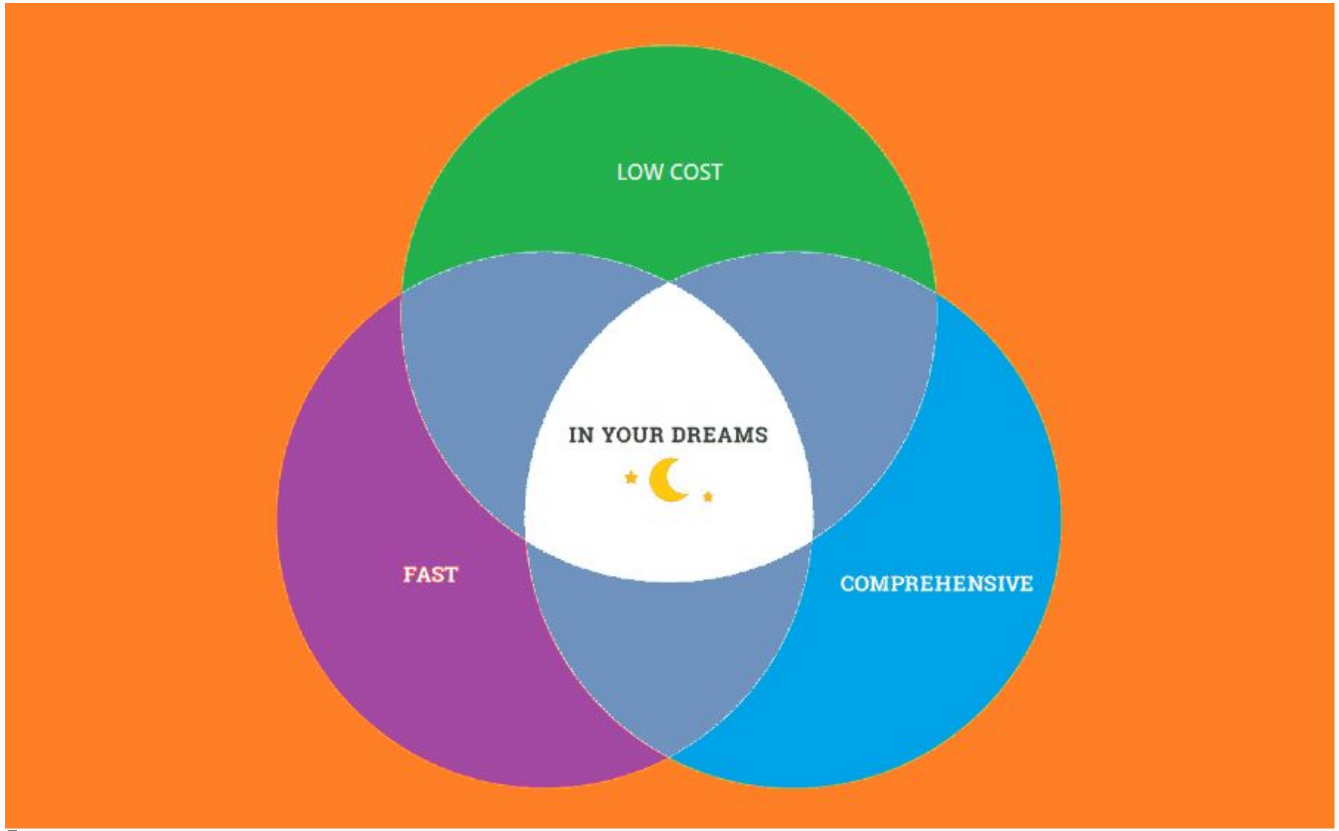


- **A single definable purpose**, end-item or result. This is usually specified in terms of cost, schedule and performance requirements.
- Every project is **unique**.
- Projects are **temporary in nature** and have a definite beginning and ending date.
- Projects are completed when the **project goals are achieved** or it's determined the project is no longer viable
- A successful project is one that **meets or exceeds the expectations of the stakeholders**.

Every project has to manage four basic constraints:

- **Time**
- **Cost**
- **Scope**
- **Quality**

Project definition and characteristics



Example of projects and NOT a project

Example of projects:

- **Building a Light Rail Transit System:** Building of the light-rail transit system to connect Bucharest city with Henry Coanda Airport.
- **Mobile App Development**
- **3D Game Development:** the team is working on a cool 3D game to be released this year on Oculus Rift. That is definitely a project with a strict deadline.
- **Building of Egypt's Pyramids:** A remarkable achievement of a bygone era, that we marvel even today. These were some of the very first projects that we know of today.

Examples of non-projects:

- Running the **daily operations** of a company
- Doing **regular maintenance** work on the Light Rail Transit system that connects Bucharest city center to the main airport
- Making breakfast everyday
- Daily **governance of the Kingdom of Egypt**

Product definition and attributes

A product represents a collection of business capabilities valuable to a defined customer segment. A product may be just software and data.

A product is a good, service, platform, application, system, created, generally for sale, to meet customer and business needs.

Product definition and attributes

A well-engineered software product should possess the following essential characteristics:

Efficiency

The software should not make wasteful use of system resources such as memory and processor cycles.

Maintainability

It should be possible to evolve the software to meet the changing requirements of customers.

Dependability

It is the flexibility of the software that ought to not cause any physical or economic injury within the event of system failure. It includes a range of characteristics such as reliability, security, and safety.

Adaptability

The software system should have the ability to get adapted to a reasonable extent with the changing requirements.

In time

Software should be developed well in time.

Within Budget

The software development costs should not overrun and it should be within the budgetary limit.

Functionality

The software system should exhibit the proper functionality, i.e. it should perform all the functions it is supposed to perform.

Product vs Project

Lifecycle

The product life cycle goes beyond a project life cycle.

Each product may have many projects associated to it during its life cycle.

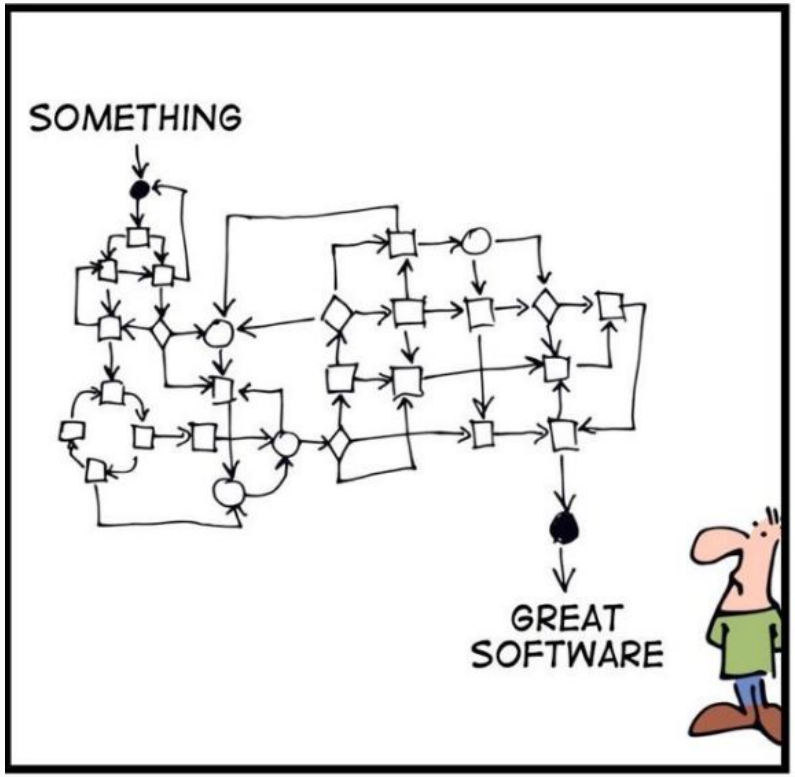
PRODUCT	PROJECT
Facets of the product life cycle are often run as a project.	Occurs in one or more phases of the product life cycle. *Ex: the retirement of a product can be a project on it's own.*
Longer lifetime	Defined lifetime
Adaptive roadmaps	Defined roadmaps *[at high level scope]*
Defined phases (development, introduction, growth, maturity, retirement)	Overlapping phases (initiation, planning, execution, monitoring and controlling, closure)

Software Development Lifecycle Overview

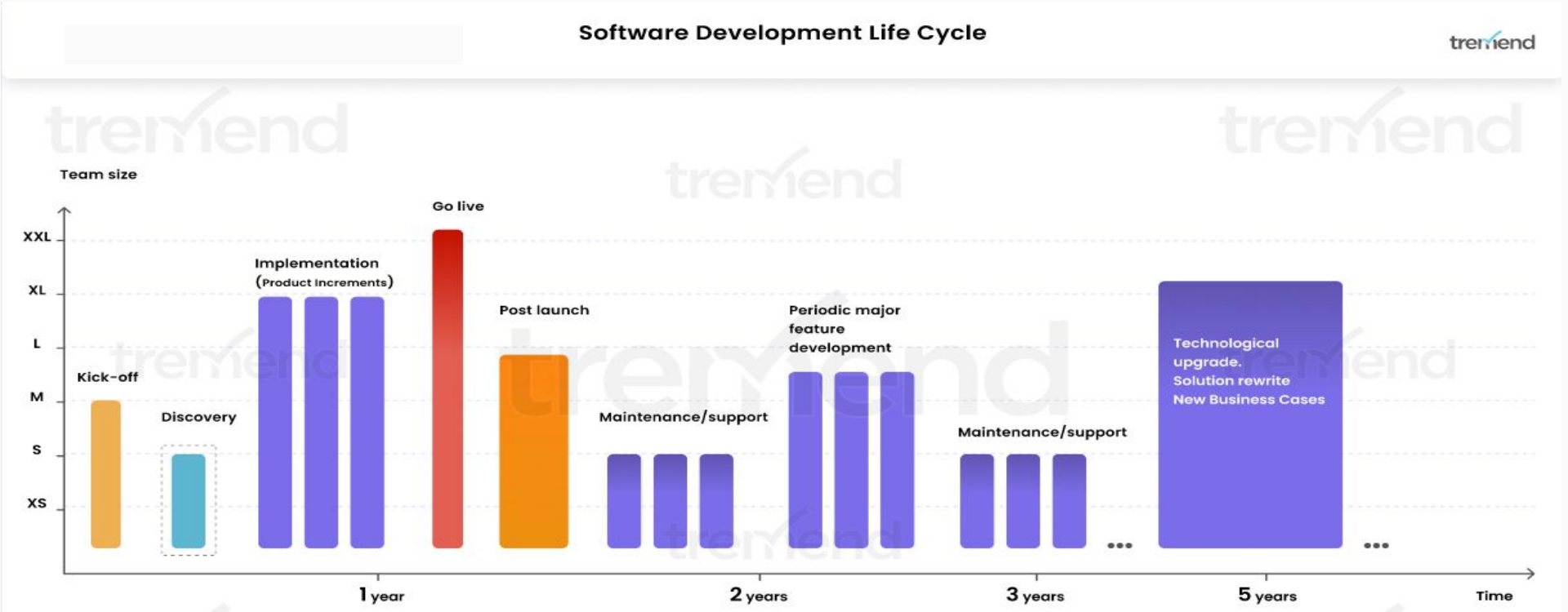
Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software.

SDLC is a process followed for a software project, within a software organization.

It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.



Software Development Lifecycle Overview



Software Development Lifecycle Overview

Stage 6: Maintenance

The maintenance phase of the SDLC occurs after the product is in full operation. Maintenance of software can include software upgrades, repairs, and fixes of the software if it breaks.

Stage 5: Testing

In the fifth stage, all the pieces of code are tested to verify and validate a software product. This is done to check the correspondence between the real and expected behavior of a requirement.

Stage 1: Analysis

During this software development lifecycle phase, the specialists meticulously collect precise requirements from the customer to present a solution fine-tuned to their needs. The analysis phase also gathers business requirements and identifies any potential risks.

Stage 2: Planning

The purpose of the second stage is to outline the scope of the problem and identify solutions. Resources, costs, time, and other aspects should be considered here. The planning phase of the SDLC is also when the project plan is developed that identifies, prioritizes, and assigns the tasks and resources required to build the structure for a project.

Stage 3: Design

In the design phase, one or more designs are created to achieve the project result. Architecture of the software product is also defined in this phase.

Stage 4: Development

The Development Phase includes several activities that are the responsibility of the developer. The developer places the outputs under configuration control, performs change control, documents and resolves problems and non-conformances found in the software products.

Organizare

Organizarea cursului și laboratoarelor

		Course		Laboratory									
		Name	Lecturer	331	332	333	334	341	342	343	344	351	352
Week 1	03.10-07.10	Introductory Course	**ALL**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Week 2	10.10-14.10	Agile foundation and fundamentals	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 3	17.10-21.10	Agile Frameworks	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 4	24.10-28.10	Agile Project Management	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 5	31.10-04.11	Introduction in Business Analysis	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 6	07.11-11.11	Requirements Analysis and Design Definition	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 7	14.11-18.11	Requirements Management	**Tremend**	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend	Tremend
Week 8	21.11-25.11	Software Architecture and Design Patterns	**Rareș**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Filler Week	28.11-29.11												
Week 10	05.12-09.12	Software Testing and Verification I	**Rareș**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Week 11	12.12-16.12	Software Testing and Verification II	**Rareș**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Week 12	19.12-23.12	DevOps	**Rareș**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Holiday 1													
Holiday 2													
Week 13	09.01-13.01	Sustainability, Ethics and Professional Practice in SE	**Rareș**	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș	Rareș
Week 14	16.01-20.01	SDLC Recap	**ALL (fizic)**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**	**Online consultations**
Sesiune 1	23.01-27.01												
Sesiune 2	30.01-03.02												
Sesiune 3	06.02-10.02												

- Cursul are loc de două ori în fiecare săptămână, laboratorul este săptămânal pentru fiecare grupă.
- De anunțat ce vom face în săptămâna 28-29.11 în funcție de orarul stabil.
- De stabilit împreună prezentarea din sesiune după ce vă explic organizarea proiectului.

Organizarea proiectului

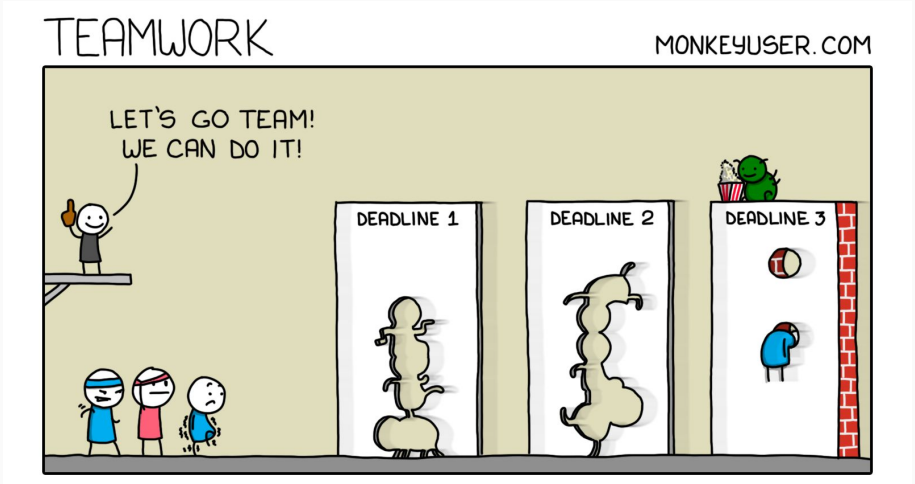
Examenul pentru Inginerie Software va consta în realizarea unui proiect software în echipă.

Echipele vor fi formate din 4-6 studenți. Vă construiți singuri echipele. Membrii echipelor pot fi cross-grupe, și cross-serii.

Etapa 1

Team Formation - Pentru această etapă aveți 3 săptămâni de pe 03.10 până pe 21.10. Tot în această perioadă va trebui să vă alegeți o temă de proiect (care e subiectul proiectului vostru). Odată ce ați stabilit componența echipei, și tema proiectului, unul dintre membrii echipei va completa formularul:

https://forms.office.com/r/9A5qbxfJyw



Organizarea proiectului

Etapa 2

Project Analysis - veți face setup-ul al proiectului (repoul de GitHub, tool-urile de dezvoltare, eventual să vă asigurați că puteți pe cât posibil să lucrați toți pe toate componentele proiectului).

Vom folosi și classroom.github.com dar pentru care va voi da linkul de join în săptămâna 2. Deci, vă rog nu creați un repo până atunci.

De asemenea veți parcurge etapa de discovery al proiectului. Aceasta etapa este definita sub forma a doua sprinturi.

Livrabil intermediar - explicat pe slide-urile următoare

Deadline: 29.11.2022 ora 12:00

Organizarea proiectului

Etapa 3

Project Development - veți dezvolta aplicația propriu-zisă. Veți defini sprinturi de dezvoltare, în care veți distribui backlogul dezvoltat în etapa anterioară.

- Vom evalua Sprint-ul 1 și 2 pentru toată lumea ca parte a livrabilului intermediar.
- Dintre sprinturile de DEV(3-6), la final alegeți 3 cele mai „reușite” sprinturi ale voastre spre a fi evaluate.
- Am definit sprintul 5 ca fiind opțional - la alegerea voastră dacă vreți să lucrați în vacanța de iarnă, dar dacă veți rezolva issue-uri în această perioadă, le veți asocia acestui sprint.
- Pe timpul sesiunii în funcție de cum poziționăm examenul vom defini un Hardening Sprint.

	Project							
	Team up	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sesiune
03.10-07.10	Team up							
10.10-14.10	Team up							
17.10-21.10	Team up							
24.10-28.10	Team up							
31.10-04.11		Project Setup						
07.11-11.11		Discovey						
14.11-18.11			BA					
21.11-25.11			"Planning"					
28.11-29.11				DEV				
05.12-09.12				DEV				
12.12-16.12					DEV			
19.12-23.12					DEV			
						DEV - optional during Holiday		
						DEV - optional during Holiday		
09.01-13.01							DEV	
16.01-20.01							DEV	
23.01-27.01								(Hardening Sprint)
30.01-03.02								(Hardening Sprint)
06.02-10.02								

Examen

Examenul va consta în evaluare proiectului pe care l-ați realizat

- Livrabil Intermediar (40 puncte sub rezerva submiterii la timp a livrabilului) **Deadline: 29.11.2022 ora 12:00**
  - Problem statement
  - Functional decomposition
  - Non-functional requirements list
  - 1 Activity/State diagram
  - Prioritized product backlog (stories identified)
  - Project charter document: project objectives, scope, vision, team, and their responsibilities and stakeholders, budget.
  - Roadmap: High level (epic/ feature level)
  - Definition of Done and Definition of ready: Story level and Sprint level

Examen

- Proiectul final (60 puncte)
  - 1 deliverable per sprint (select best 3 sprints):
    - sprint backlog (snapshot after planning),
    - Sprint report,
    - Retrospective outcome,
    - Review session (register application Demo)
    - User stories & acceptance criteria
  - Software Architecture report
  - Testare
    - Scalability testing (?)
    - User acceptance testing
    - Raport cu rezultatele acestor teste.
  - CI pipeline
    - Puteți face un demo offline cu rularea pipeline-ului și să adăugați linkul în Wiki-ul / README.md-ul proiectului
  - Prezentarea aplicației finale într-un demo live
    - Validarea funcționalității aplicației
    - Adresarea considerațiilor de etică și de sustenabilitate

Course 1 - Introduction in Software Engineering

Q&A