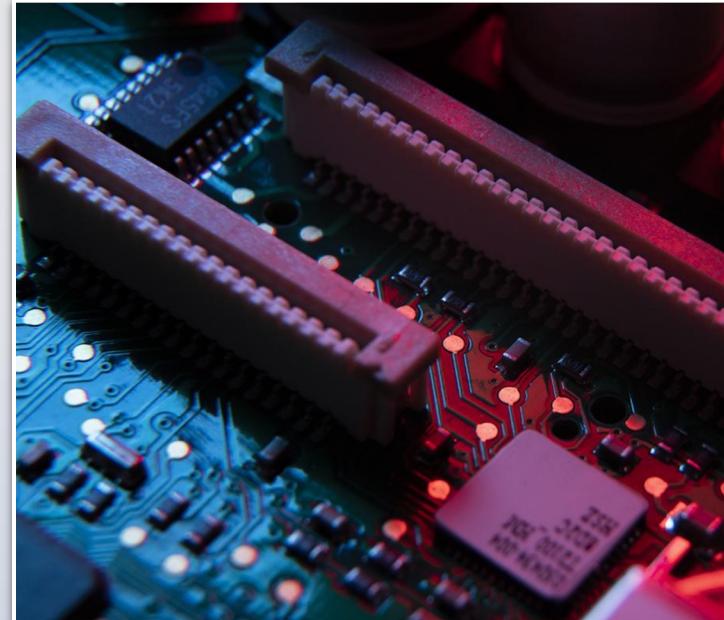




Introduction to Software Engineering

Software Engineering Course

2023 - 2024





Agenda

You can delete this slide when you're done editing the presentation

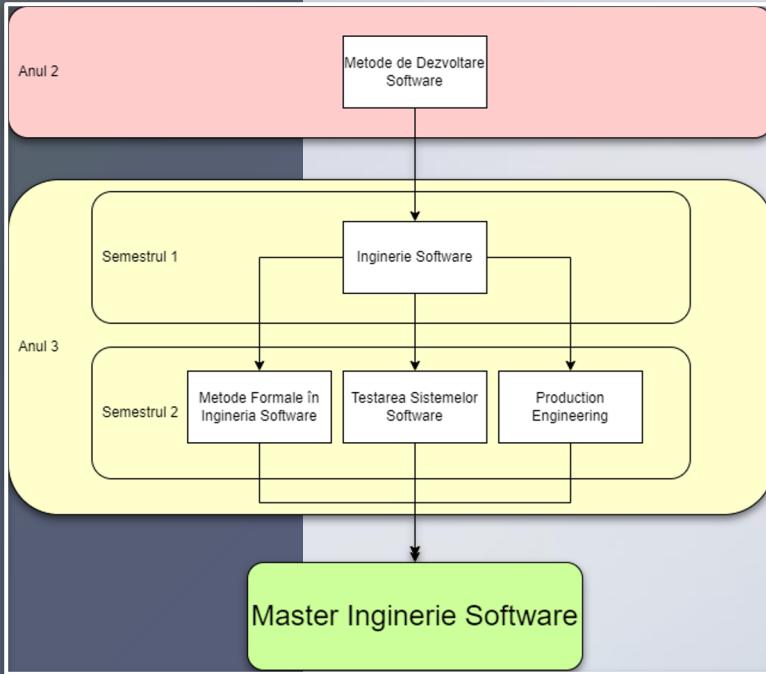
1.	What is Software Engineering?
2.	What we will study and why?
3.	Administrative information
a.	Assessment
4.	Q&A



01

What is Software Engineering?



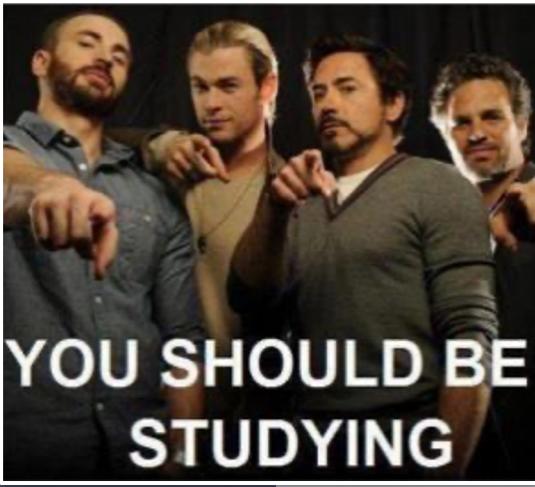


Software Engineering roadmap @ Informatics FMI



02

What are we going to study & Why?



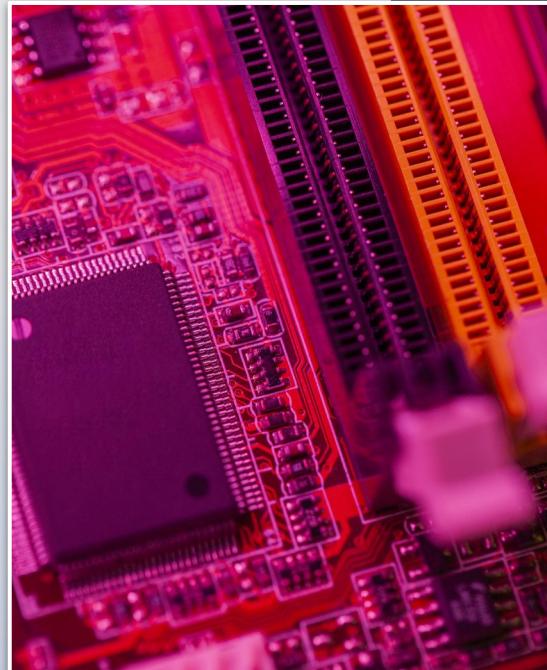
•••

Software Development Lifecycle (SDLC)

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 (SDLC)

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.



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.

Stage 1: Analysis

The needs of the software project are identified, the problem or opportunities stated, the goals are aligned with business requirements.. Market research is carried out, and the needs of the target audience who will use the product are determined to implement a project plan.

Stage 2: Planning

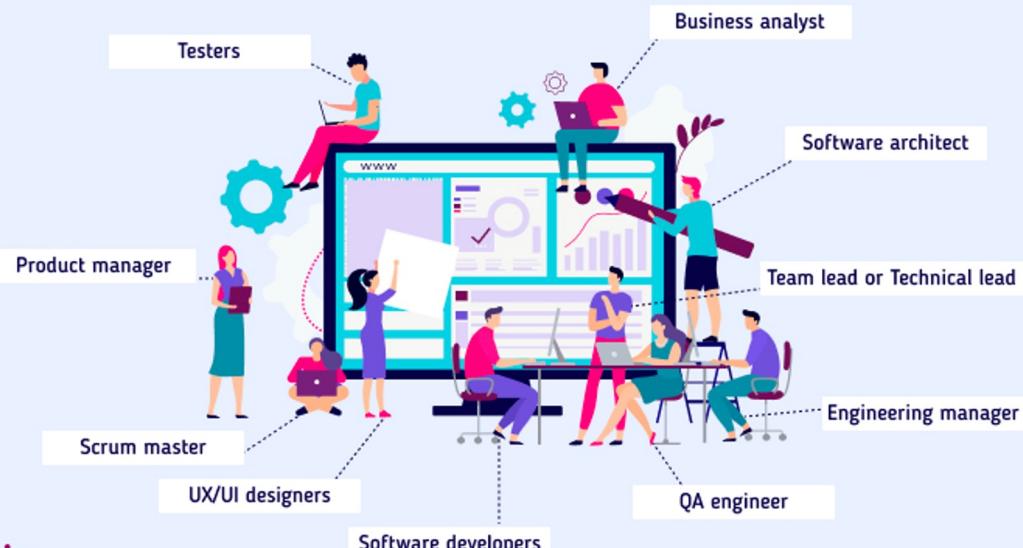
The purpose is to outline the scope of the problem and identify solutions, also to evaluate the feasibility of the project. Resources, costs, time, and other aspects should be considered here. The project plan is developed that identifies, prioritizes, and assigns the tasks and resources required to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

Stage 3: Design

Focuses on converting the information gathered during the planning and analysis phase into clear requirements for the development team, defining one or more designs through which to achieve the project result. Architecture of the software product is also defined in this phase.

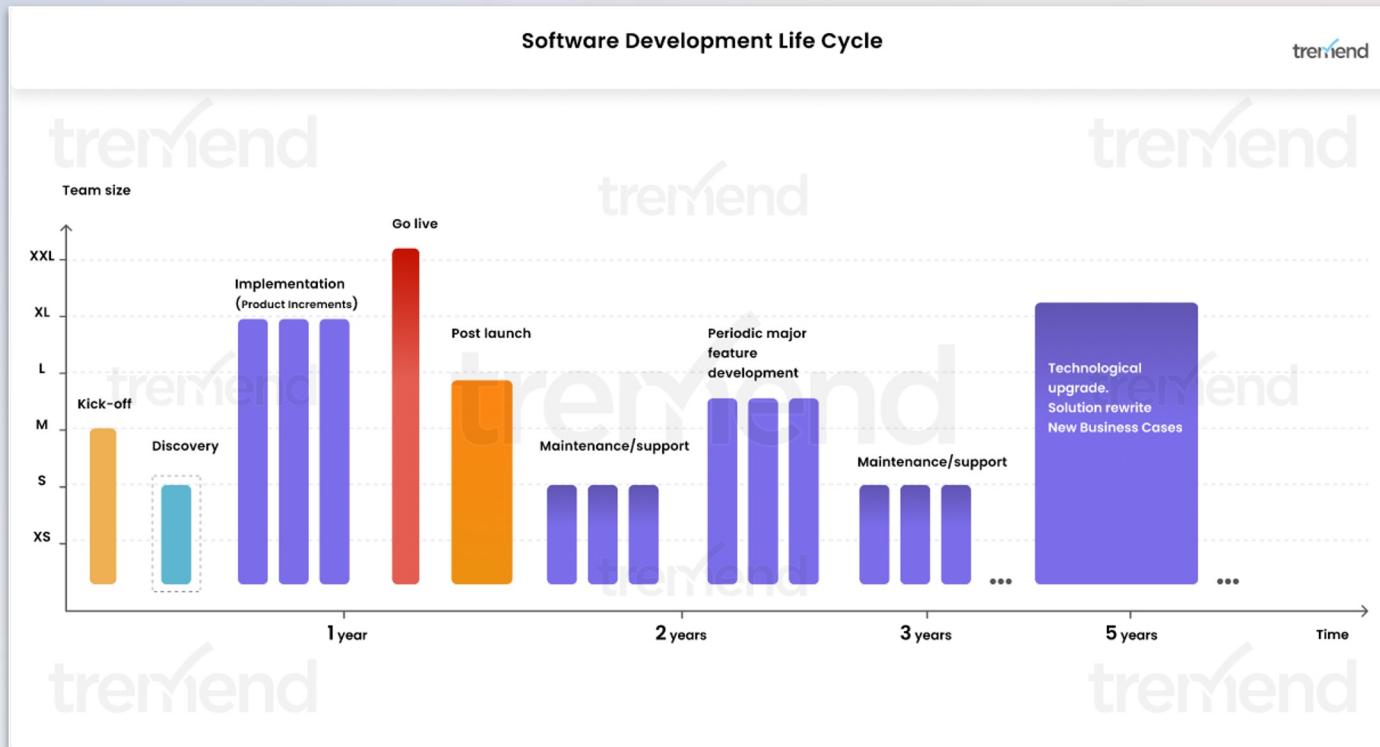
Roles in SDLC

Your perfect development team



<https://alcor-bpo.com/recruitment-news/10-key-roles-in-a-software-development-team-who-is-responsible-for-what/>

Software Development Lifecycle (SDLC)



Why do we write code?

Software is not a goal, but the means to an end!

" Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.

1st Agile principle

"Identifying Value" where "Value is what the customer is willing to pay for"; in other words "Precisely specify value by product"

1st Lean principle

"Take an economic view", "Organize around value"
SAFe Lean-Agile principles



Efficient software development life cycles

"Working software is the primary measure of progress",
7th Agile principle

"Make Value Flow without Interruptions"

SAFe Lean-Agile 6th principle

Software is more about people than it is about machines

" The best architectures, requirements, and designs emerge from self-organizing teams."

11th Agile principle

"Pursue Perfection"

5th Lean principle



Project definition and characteristics

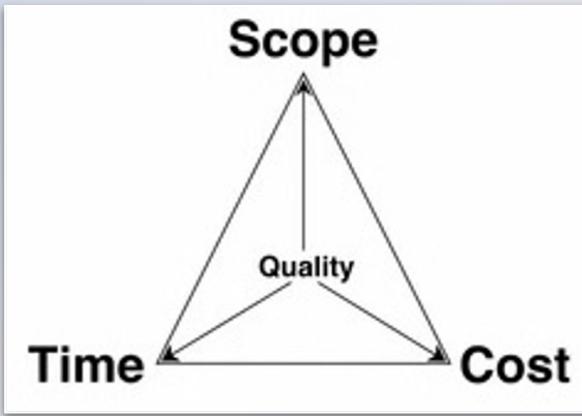
A project is a **set of tasks** which must be completed in order to arrive at a particular **goal** or **outcome**.

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.

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



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**



Projects and "Not projects"

Projects

- **Building a Light Rail Transit System:** Building of the light-rail transit system to connect Bucharest city with Henri 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.

Not 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

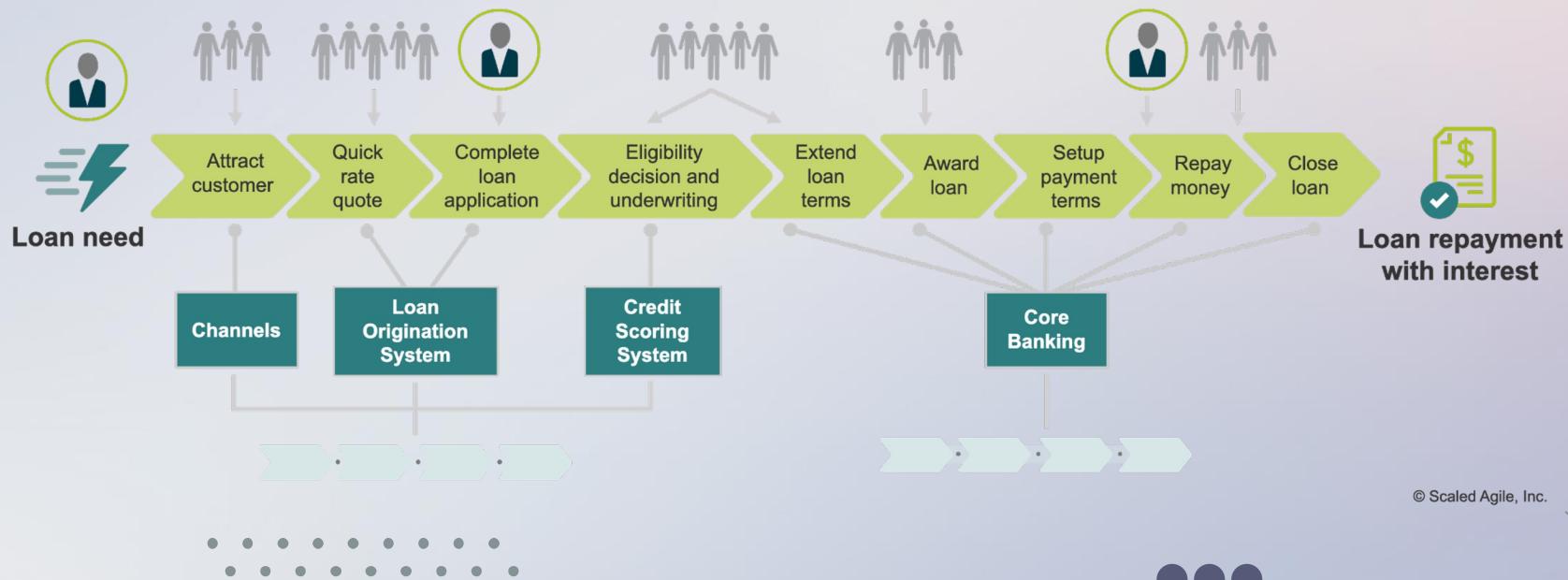
A product represents a **collection of business capabilities** valuable to a defined customer segment. A product may be just software and data.

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

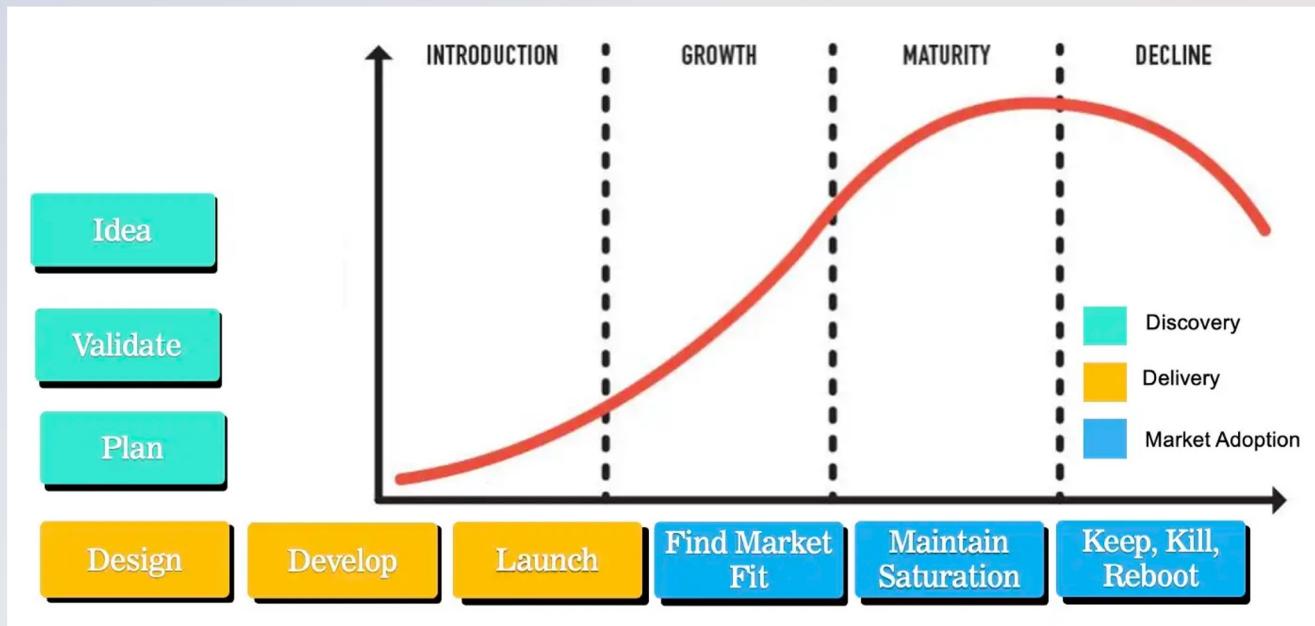


Product example

Consumer loan by © Scaled Agile, Inc.



Product lifecycle



<https://www.abtasty.com/blog/great-product-lifecycle-manager/>





Product vs. Project

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. <i>Ex: the retirement of a product can be a project on its own.</i>
Usually the product lifetime is not limited or it takes several years	Defined lifetime, usually less than 1,2 years
Adaptive roadmaps	Defined roadmaps (<i>at a high level scope</i>)
Defined phases (development, introduction, growth, maturity, retirement)	Overlapping phases (initiation, planning, execution, monitoring and controlling, closure)



Types of companies (*in digital era*)

Product based company	Consultancy/Service company
Has its own product to sell to consumers through the market.	Offers services to its clients.
Focuses on product development, innovation and quality.	Focuses on employee expertise, solutions to their clients and customization.
Limited or no interaction with the customer	Ongoing, close interaction with clients
Product companies work for years keeping one technology/product in mind	More frequent change of project, technology, domain.
e.g. Amazon, Google, SAP, IBM, Microsoft, Adobe, Salesforce	Accenture, McKinsey, IBM, Cognizant, Endava, Tremend (Publicis Sapient)





Types of services

Types of services:

- Strategy
- Consulting in digital, technological, operational areas
- Software development, etc

Type of software development services:

- Custom software development services
- Mobile software development services
- Web development services (CMS, Platforms - PHP, JAVA, .NET, Cloud computing)
- Embedded systems development

Business model in consulting

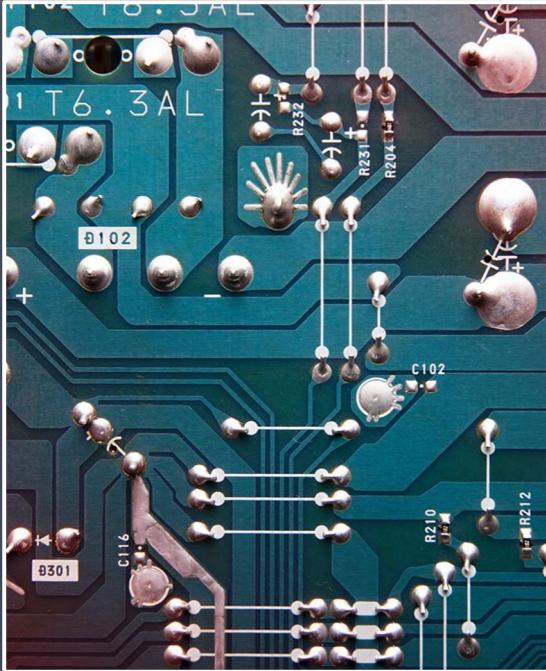
- Team augmentation - where a person or team might join an existing team for a short time to help accelerate some development
- Outcome-based consulting - where a team will identify product-market fit or problem-solution fit and build the solution





03

Organizare

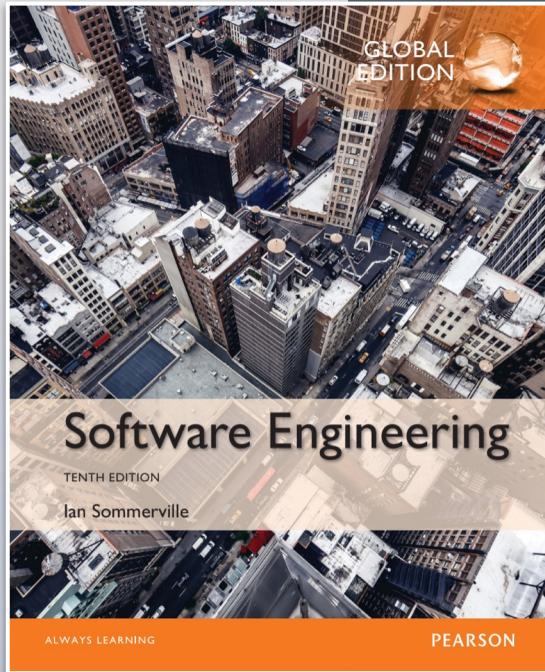




Materiale

Veți avea următoarele materiale didactice:

- Principalul material vor fi prezentările asociate fiecărui curs.
- Una dintre cele mai bune cărți de Software Engineering este cea din imagine. Veți avea la fiecare curs asociate câteva pagini din cartea aceasta pentru aprofundare.
- Pentru anumite subiecte veți avea și niște research papers
- Pentru anumite subiecte veți avea și referințe non-academice, în special grafice.



Calendar

		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Holiday 1	Holiday 2	Week 13	Week 14	Sesiune 1	Sesiune 2	Sesiune 3
		02.10-06.10	09.10-13.10	16.10-20.10	23.10-27.10	30.10-03.11	06.11-10.11	13.11-17.11	20.11-24.11	27.11-29.11	04.12-08.12	11.12-15.12	18.12-22.12			08.01-12.01	15.01-19.01	22.01-26.01	29.01-02.02	05.02-09.02
Course	Name	Introduction Course	Business Analysis	Requirements Analysis	Requirements Management	Agile Planning / Team Management	UI/UX	Containerization with Docker	Software Architecture		CI/CD	Gaming Development	Extended Testing			Legislation Course				
Lecturer	ALL	Tremend	Tremend	Tremend	Tremend	Invitat	Rareş		Invitat	Ciprian	Rareş				Alex Moraru + Rareş	Recap				
Live Lab				Tremend - BA+Req BA	Tremend (Req Mngmt)											Rareş+				
Online Consultations						Tremend (Req) Tremend (UX)				Invitat					Rareş	Rareş				
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Filler Week	Week 10	Week 11	Week 12	Holiday 1	Holiday 2	Week 13	Week 14	Sesiune 1	Sesiune 2	Sesiune 3
		02.10-06.10	09.10-13.10	16.10-20.10	23.10-27.10	30.10-03.11	06.11-10.11	13.11-17.11	20.11-24.11	27.11-29.11	04.12-08.12	11.12-15.12	18.12-22.12			08.01-12.01	15.01-19.01	22.01-26.01	29.01-02.02	05.02-09.02
Project planning	Team up	Team up	Team up																	
Sprint 1				Project Setup	Discovery															
Sprint 2						BA	UX													
Sprint 3								PLANNING + DEV												
Sprint 4									DEV	DEV										
Sprint 5										DEV	DEV									
Sprint 6															DEV - opti DEV - optional during Holiday					
Sesiune																DEV	DEV			
Evaluation								LIVRARE										LIVRARE		

- Cursul are loc de două ori în fiecare săptămână.
- Sunt două săptămâni în care are loc laboratorul fizic la facultate. (*Live Lab*)
- Veți avea săptămâni cu *Online Consultations* în loc de Laborator - pe subiectul anunțat în săptămâna respectivă puteți să cereți feedback din partea noastră pe ce ați făcut în proiect.



Evaluarea

Evaluarea disciplinei va se axează în jurul unui **proiect software în echipă**.

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

Proiect Nou

You must define a project that delivers a software based solution to an **identifiable need of potential users**. You will engage in **requirements gathering, design an application, development, testing and delivery**. A working demo of the application must be presented at the end of the semester.

sau

Continuați proiectul MDS

You must define a new project for your MDS application. You need to create a **document that explains what is already done** in the application then identify new **requirements** from the potential users, **design the features, develop, test and deliver** the new features. A working demo of the application as a whole, and the new features must be presented at the end of the semester.

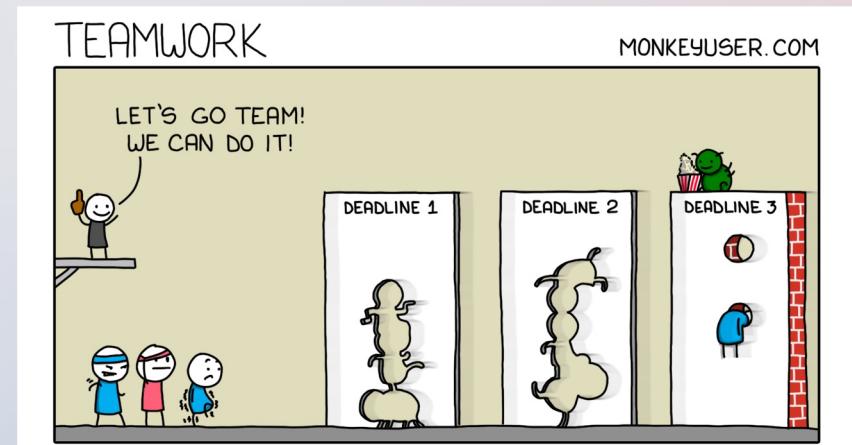




Evaluarea

Etapa 1 – Formarea echipei

- Până pe **15.10 23:59** va trebui să vă stabiliți echipa de proiect (4-6 studenți).
- Odată stabilită echipa, veți identifica o temă de proiect, sau feature-urile adiționale pentru proiectul vostru.
 - Care este problema pe care ați identificat-o? Nevoia pe care o generează? Soluția pe care o propuneți, și beneficiarii soluției?
- Până pe **15.10 23:59** trebuie ca un membru al echipei să completeze următorul formular:
 - [Formular](#) (se va deschide din 7.10)





Evaluarea

Etapa 2 – Analiza, planificarea și proiectarea proiectului

- Se desfășoară în perioada **16.10 - 10.11**
- Cursurile din Etapa 1 și o parte din Etapa 2 au ca rol să vă învețe **cum să faceți** Analiza, Planificarea și Proiectarea proiectului.
- Până pe **24.11 23:59** veți avea încărcate în GitHub toate artefactele **Livrabilului Intermediar**

Checklist livrabil intermedier:

- Problem statement/product vision
- Product features & functionalities
- Non-functional requirements
- Either one activity diagram or a state diagram
- A set of user stories
- Prioritized product backlog
- Roadmap
- UI mockups





Evaluarea

Etapa 3 – Implementarea

- Se desfășoară în perioada **13.11 - 29.01 (ziua de prezentări din sesiune)**
- Cursurile din 13.11 - 22.12 au ca scop să înțelegeți ce trebuie să livrați la final în special pe partea de arhitectură, testare, livrare.
- Săptămânile de vacanță sunt definite ca un sprint, dar puteți să nu faceți nimic, fiind vacanță. Dacă alegeti să lucrați, nu uități să realizați artefactele din checklist.

Checklist livrabil final:

- Fiecare 2 săptămâni vor fi definite ca un Sprint.
- Puteți să nu lucreați în sprint-ul de vacanță.
- Pentru fiecare sprint de dezvoltare veți crea:
 - Sprint backlog at the beginning of the sprint
 - Sprint specific user stories & acceptance criteria at the beginning of the sprint
 - Sprint report at the end of the sprint
 - Retrospective outcome at the end of the sprint
 - Review session at the end of the sprint
- Software Architecture report la prezentare
- Proiectul va implementa mai multe paradigme de testare (detaliat la momentul cursului)
- Definirea și rularea unui CI pipeline pentru proiect
- Prezentarea aplicației finale într-un demo live
 - Validarea funcționalităților propuse ale aplicației





Evaluarea

Punctarea

Puteți lua maximum 110 puncte din 100

- 80 de puncte pentru proiect
 - 30 de puncte pentru livrabilul intermediar
 - 50 de puncte pentru livrarea finală
- 30 de puncte pentru quiz-urile care au loc la curs și laboratoarele fizice
 - $12 \text{ cursuri} + 2 \text{ laboratoare live} = 14 \text{ quiz-uri}$. Fiecare quiz va valora 2.5 puncte. Încă 2 puncte se mai acordă când ai 3 quiz-uri rezolvate la rând (adică ai fost prezent la 3 quiz-uri consecutive cursuri și laboratoare. De exemplu, dacă între două cursuri e un lab la care n-ai fost, se pierde „streak”-ul)
 - Punctajul se acordă doar dacă luați măcar 45 de puncte din proiect.

Alte informații

- Pentru comunicare vom folosi și Microsoft Teams - Team-ul creeat special pentru acest curs.
- Pentru toate livrabilele vom utiliza platforma **classroom.github.com**. Este neapărat. Pentru link, mergi pe slide-urile următoare.
- Pentru toate livrabilele urmărim să existe un timestamp (când a fost redactat / încărcat în repo).





Cum vă alăturați în GitHub Classroom

După ce te autentifici în contul de GitHub, îți găsești numele în lista aceasta și tă-l asociezi.

The screenshot shows the GitHub Classroom interface. At the top, it says "Join the classroom: **inginerie-software-2023-2024-classroom**". Below this, there is a message: "To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email)." There is also a link "Can't find your name? [Skip to the next step →](#)". A table titled "Identifiers" lists several GitHub accounts:

Identifiers
adela-gabriela.danescu
adela-nicoleta.corbeanu
adrian-ionut.sefcic
adrian-octavian.patrascu
adrian-valentin.nafornita
albert-ioan.itichi
albert-ionut.balaute-amarcialei



Cum să alăturați în GitHub Classroom

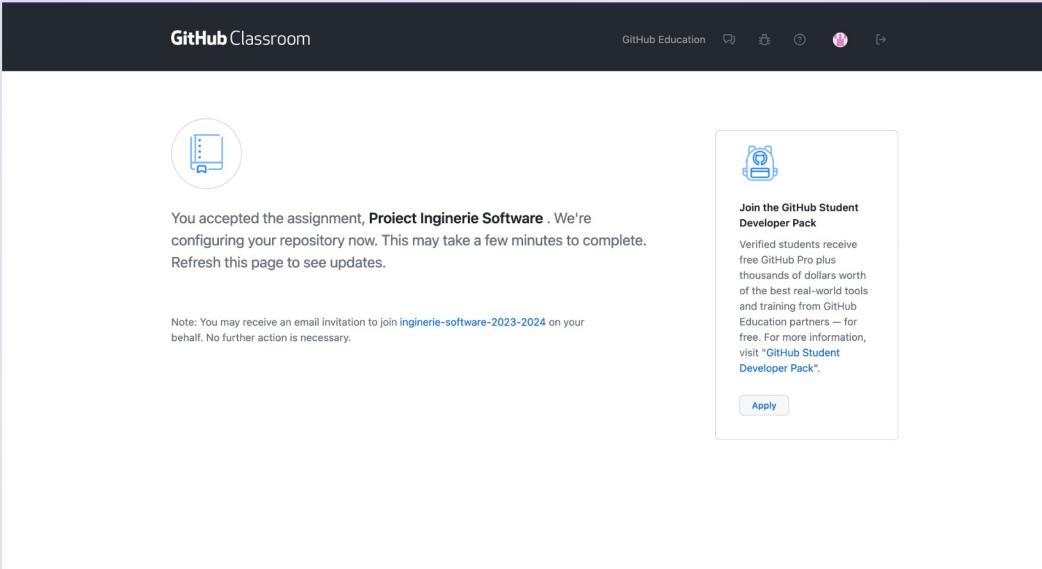
Dacă ești primul membru al echipei care intră pe classroom, vei crea echipa. Dacă a fost deja creată echipa, poți să o găsești în listă, și să te alături.

The screenshot shows the GitHub Classroom interface. At the top, there's a dark header with the GitHub Classroom logo and navigation links for GitHub Education, notifications, and user profile. A green notification bar at the top states: "Your account is linked to rares.cristea2 on the roster. If this is wrong, please reach out to your instructor." Below this, the main content area has a title: "inginerie-software-2023-2024-classroom" followed by "Accept the group assignment — Project Inginerie Software". A message below the title says: "Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later." Underneath, there's a section titled "Join an existing team" with a button for "rares-echipa-test 1 Join". Below that, there's a link "OR Create a new team" with a "Create a new team" button.



Cum să alăturați în GitHub Classroom

După ce accepți assignmentul te vei alătura repo-ului sau se va crea, după caz.



The screenshot shows a GitHub Classroom assignment acceptance page. At the top, it says "GitHub Classroom". Below that, there's a circular icon with a book and a pencil. The main message reads: "You accepted the assignment, **Project Inginerie Software**. We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates." A note below states: "Note: You may receive an email invitation to join [inginerie-software-2023-2024](#) on your behalf. No further action is necessary." To the right, there's a box titled "Join the GitHub Student Developer Pack" which says: "Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. For more information, visit ["GitHub Student Developer Pack"](#)". An "Apply" button is at the bottom of this box. The bottom of the page has three decorative dots.



Cum vă alăturați în GitHub Classroom

Iar acesta este link-ul. Fiecare membru al echipei trebuie să îl acceseze.

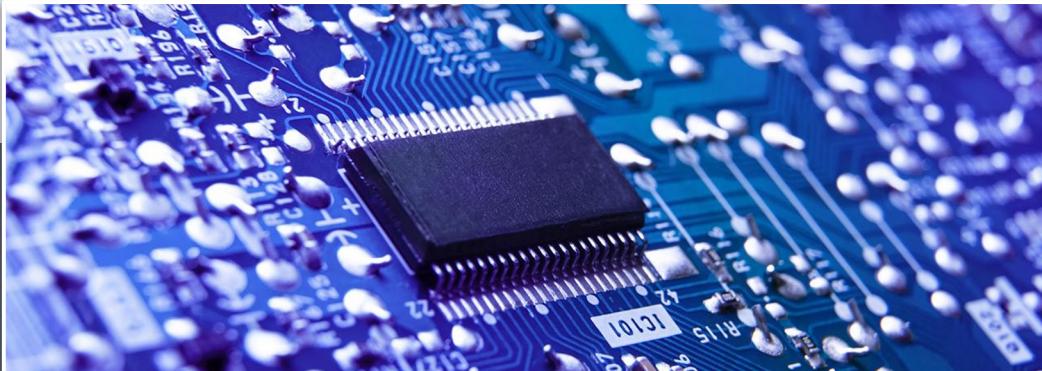
<https://classroom.github.com/a/oyyoaT98>

Nu uitați că până pe **15.10 23:59** un membru al echipei trebuie să completeze următorul Formular





Quiz!





Q&A

(dacă ne rămâne timp vă arăt și feedback-ul din anii
trecuți)





Feedback din anul anterior

01



Industrie

„Mi-a placut faptul ca a invitat la curs persoane din industrie care sa ne explice materia aplicata pe realitatea din domeniu si experienta proprie”.

„Cursul este poate cel mai interesant si util dintre toate cursurile de la facultate. De asemenea, Rares a fost foarte intrelegator si a adus multi oameni din industrie.”
„speakeri externi din industrie”

02



Prezență

„Din pacate, ceva (poate ora cursului) a facut ca prezenta la curs sa fie mica.”
„Prezenta scazuta vorbeste de la sine, ceva nu pare sa functioneze”



Feedback din anul anterior

03



Content și Predare

„Materia a fost de actualitate.”

„As imbunatati modul de predare, cateodata se simtea o atmosfera foarte formala si devinea plictisitor.”

„Profesorul a fost mereu disponibil pentru intrebari si clarificari.”

„am apreciat foarte mult faptul ca s-a tinut cursul si cu 2 studenti si s-a depus interes”

„materiale bune si clare”

„As imbunatati timpul de raspuns pentru intrebarile adresate pe Teams.”

„Mi-a placut respectul si firea relaxata fata de studenti.”

„Explicatii neclare asupra materiei, avand in vedere obiectivul final.”

„Materia a fost in rest foarte interesanta.”

04



Notarea

„Notarea s-a facut putin in defavoarea studentilor.”

„notare fair”

„puncte bonus pe prezenta sau 1p/prezenta sau mai mult pentru a creste prezenta la curs”

„Modul in care a fost corectat Proiectul final a fost mult prea drastic, mai ales pentru felul in care s-au desfasurat materia pe tot parcursul semestrului.”

„Ar trebui mai accentuată importanța documentației in notarea proiectului față de codul scris in sine, au fost colegi care nu au înțeles acest lucru și s-au concentrat pe aspectul programării, luând note mai mici.”



Feedback din anul anterior

05



Organizare

„un principal aspect care a scăzut prezența la curs cel puțin pentru seria 35 este faptul că s-a evitat din start tinerea cursului la ora stabilită; nicio problema dacă trebuia să venim cu o alta serie, însă ora cursului la alta serie se suprapunea cu un alt curs la noi; nu spun că dacă era un interval orar liber veneau toți, dar cel puțin pentru mine asta a fost un mare dezavantaj care m-a facut să cred că seria 35 nu este importantă (a se nota că era într-adevar și cursul de la 8-10 un interval liber, dar la cum se face naveta prin București, trezitul la 6 nu este o opțiune pentru toată lumea)“

„curs la ore mai bine alese“

Absolut inutil acest curs, nu inteleag **care era diferența dintre el și cursul de Metode de Dezvoltare Software**. Trebuia să vina cei de la Tremend să ne predea **informatii pe care le putem gasi în 5 min pe internet?** Din internshipul meu la Tremend nici ei nu fac mai nimic din ce predau. **Mai bine invatam de la inceput lucruri mai folositoare: docker, jenkins, kubernetes, lucruri de dev ops**, nu cum să facem project charter de parca ne angajam ca manager sau ne cautam singuri clienti.

