Software Engineering

Lesson #01 - Lecture

Agenda: Lesson #01 - Software Engineering - Lecture

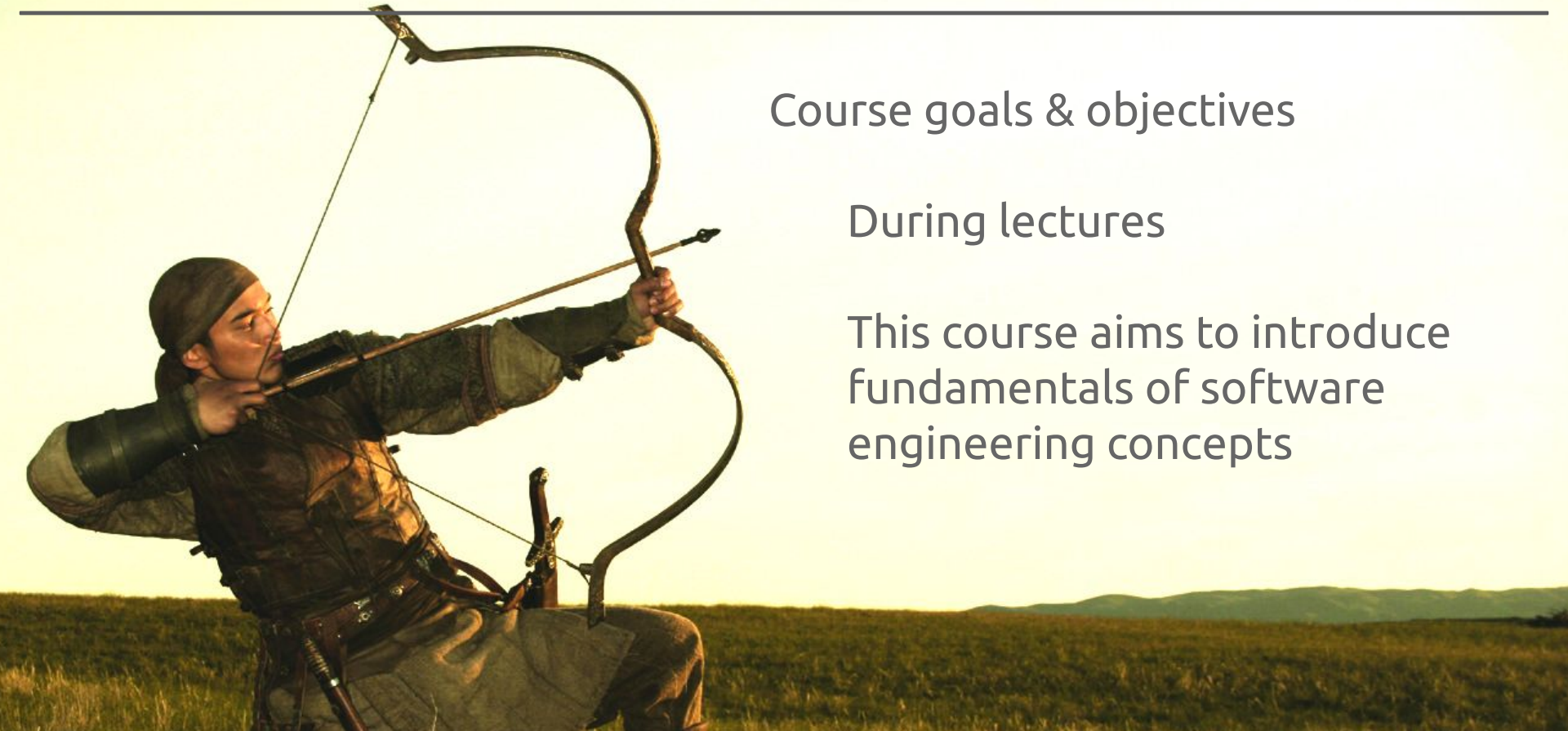
1	Course overview
2	Course organization
3	Course importance
4	Introduction to Software Engineering

Agenda: Lesson #01 - Software Engineering - Lecture

1	**Course overview**
2	Course organization
3	Course importance
4	Introduction to Software Engineering

Course overview

Course overview



Course goals & objectives

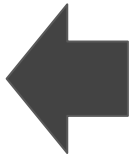
During lectures

This course aims to introduce fundamentals of software engineering concepts

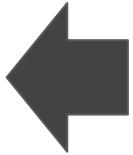
Course overview

Course Goals & Objectives

- Introduction to Software Engineering
- System Dependability and Security
- Advanced Software Engineering
- Software Management



Main focus of course



Short review -> Main focus on master degree study

Agenda: Lesson #01 - Software Engineering - Lecture

1. Course overview
2. **Course organization**
3. Course importance
4. Introduction to Software Engineering

Course organization

Course Details

- Academic Year: 2023 - 2024
- Fall 2023
- 4 months (September 2023 - December 2023)
- 3 hours in a week (2 h. lecture - 1h. practice)

Course organization

Course Code:

1cmqhus

Microsoft Teams

Course organization

Policy:

Standard KBTU Academic Policy is used

Course organization

ALDAMURATOV Jomart

E-mail: z.aldamuratov@kbtu.kz

+8 years experience in Education (KBTU & SDU)

~10 years experience in Enterprise IT
(Toyota Motor Kazakhstan LLP -
Almaty, Kazakhstan - 2008 - 2018)

Master of Computer Science
(Ritsumeikan University, Kyoto, JAPAN - 2005-2007)

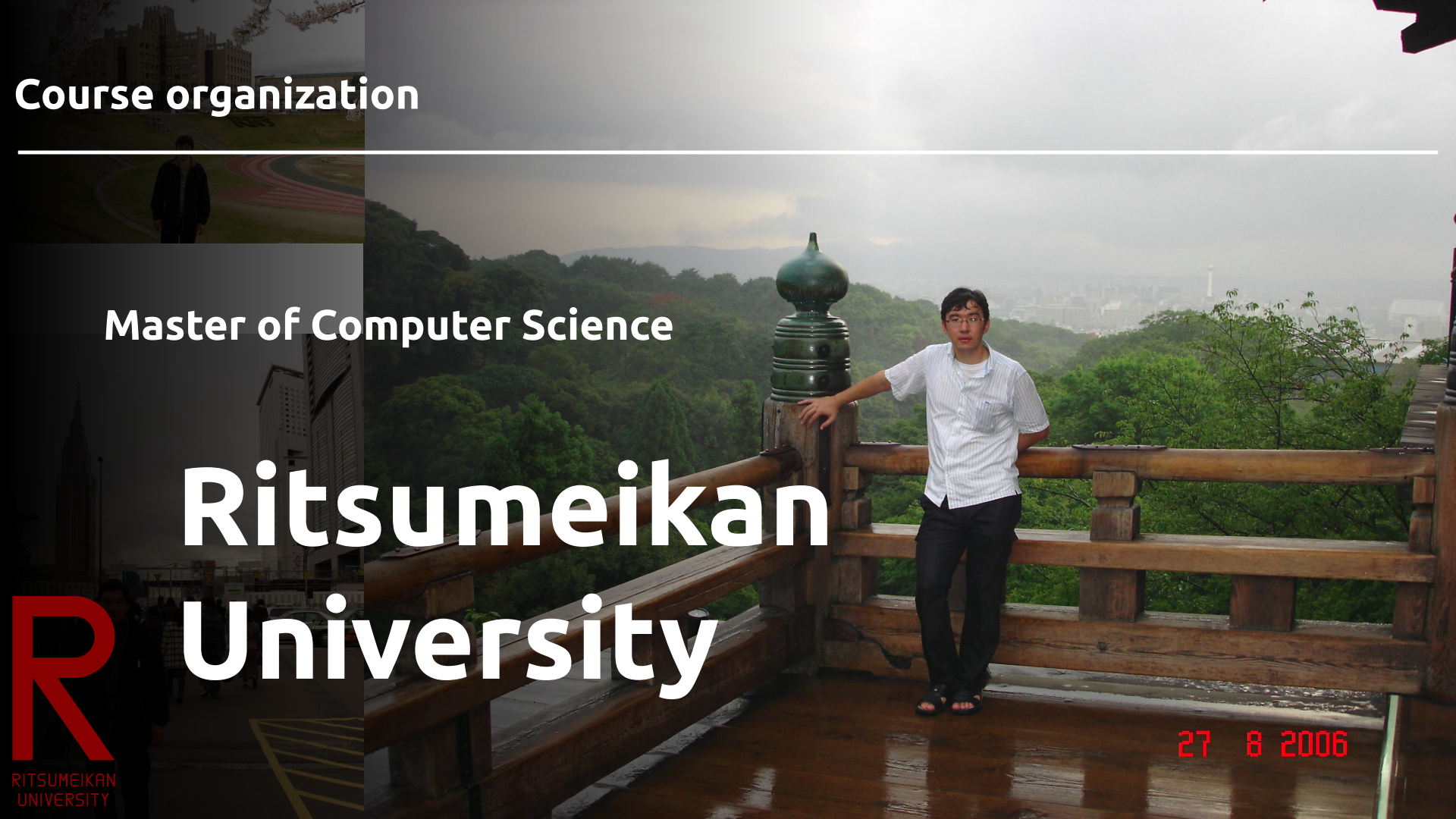
TOYOTA

RITSUMEIKAN UNIVERSITY
立命館大学

Course organization

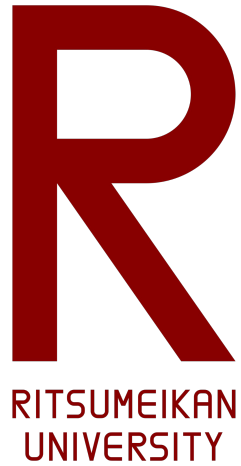
Master of Computer Science

Ritsumeikan University



RITSUMEIKAN UNIVERSITY

Instructor's Master Degree Thesis



- University: Ritsumeikan University
- Thesis Name: Negotiating method among alternatives (sub-goals) in Goal-Oriented Requirements Analysis (2007-06-21)
- Professor: prof. Atsushi Ohnishi
- Link: https://www.ieice.org/ken/paper/20070621eAVo/eng/
- Related Topics: Requirements Analysis / Requirements Elicitation / Goal-Oriented Requirements Analysis / Analytic Hierarchy Process /

Introduction to Software Engineering

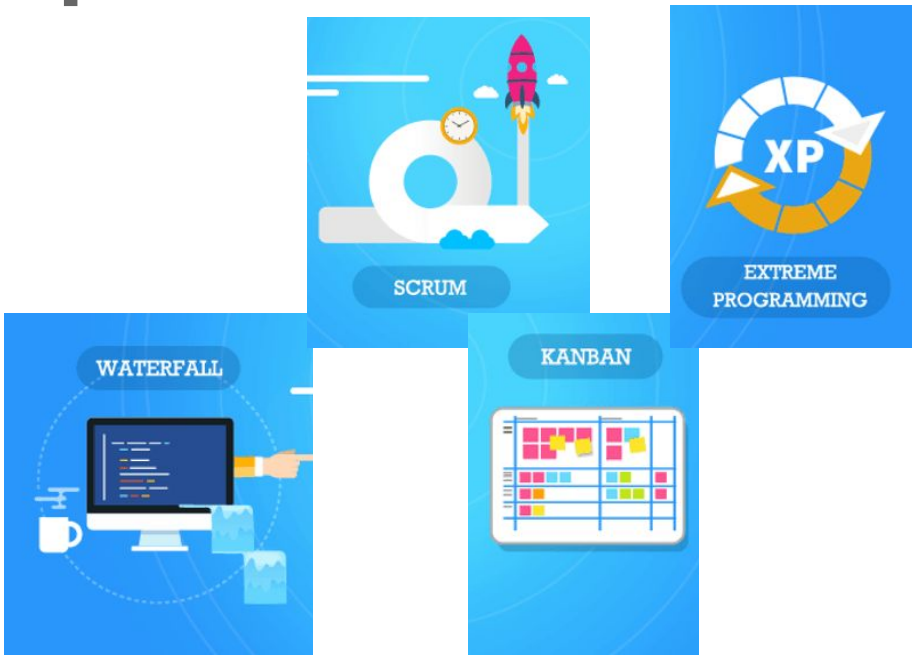
Part #01

Lesson 01: Introduction to Software Engineering

- Course overview
- Course organization
- Course importance
- Introduction to Software Engineering

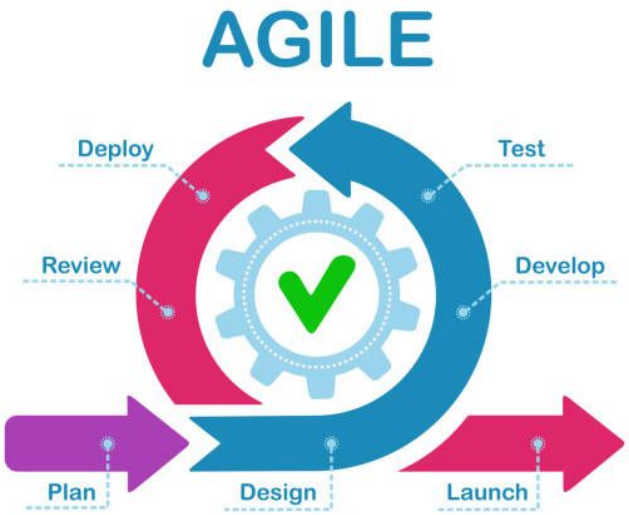
Lesson 02: Software processes & Agile software development

- Software process models
- Process activities
- Coping with change
- Process improvement



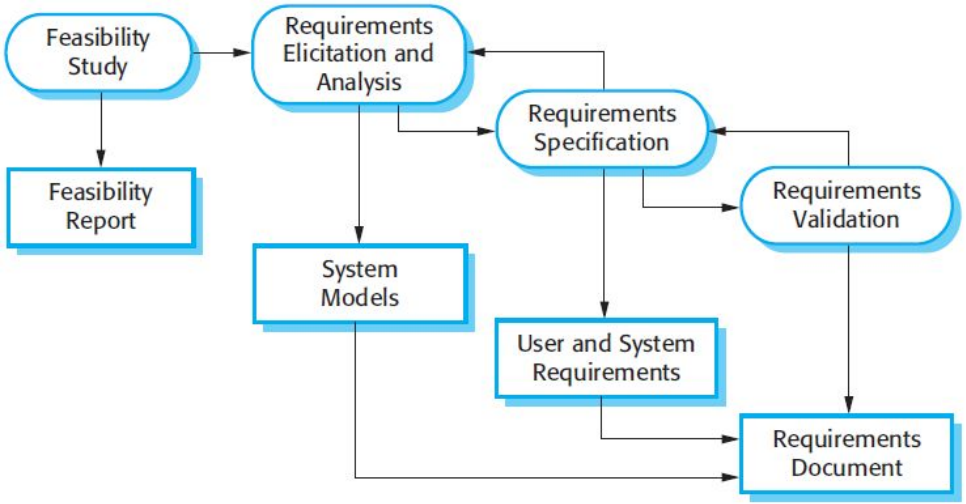
Lesson 02: Software processes & Agile software development

- Agile methods
- Agile development techniques
- Agile project management
- Scaling agile methods



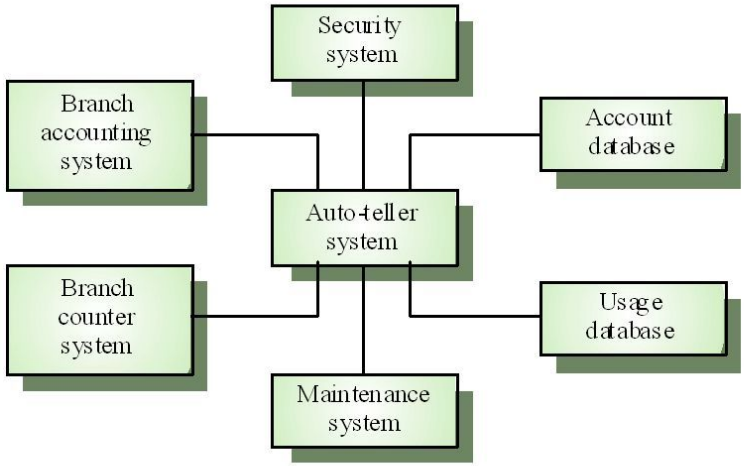
Lesson 03: Requirements engineering

- Functional & Non-functional requirements
- Requirements engineering processes
- Requirements elicitation
- Requirements specification
- Requirements validation
- Requirements change



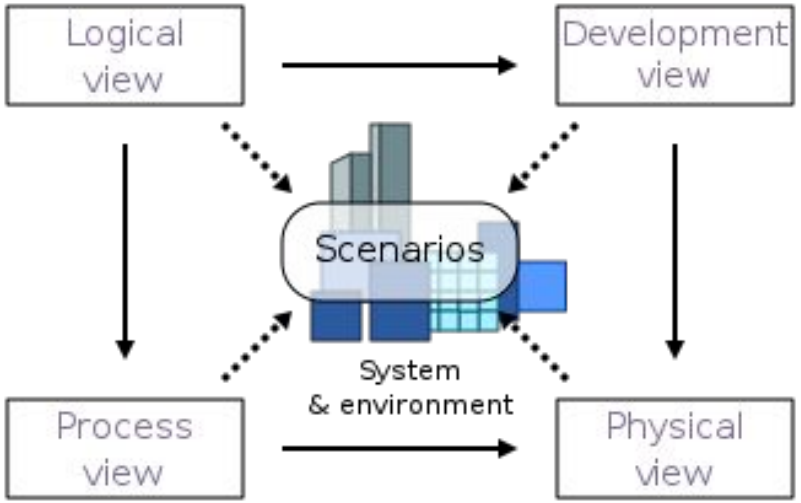
Lesson 04: Systems modelling

- Context models
- Interaction models
- Structural models
- Behavioral models
- Model-driven architecture



Lesson 05: Architectural design

- Architectural design decisions
- Architectural views
- Architectural patterns
- Application architectures



Lesson 06: Design and implementation

- Object-oriented design using UML
- Design patterns
- Implementation issues
- Open-source development

Lesson 07: Software testing

- Development testing
- Test-driven development
- Release testing
- User testing

Lesson 08: Software evaluation

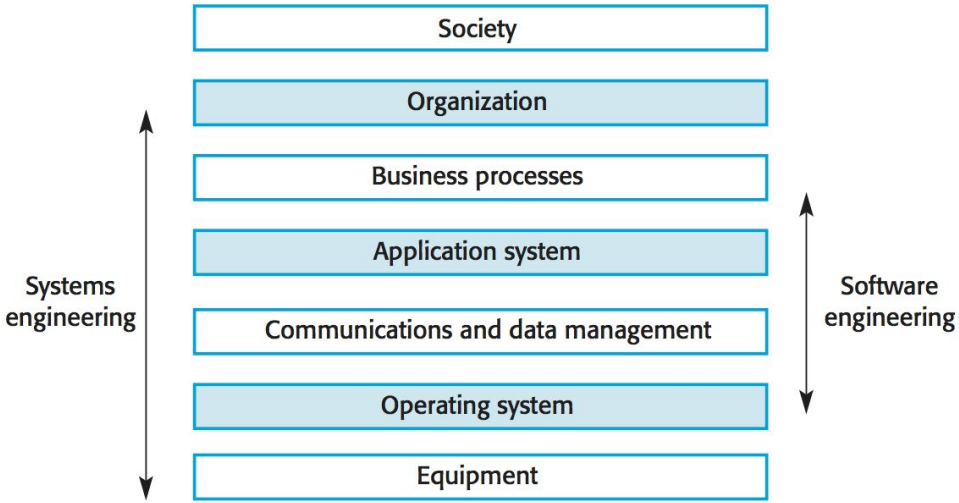
- Evolution processes
- Legacy systems
- Software maintenance

System Dependability and Security

Part #02

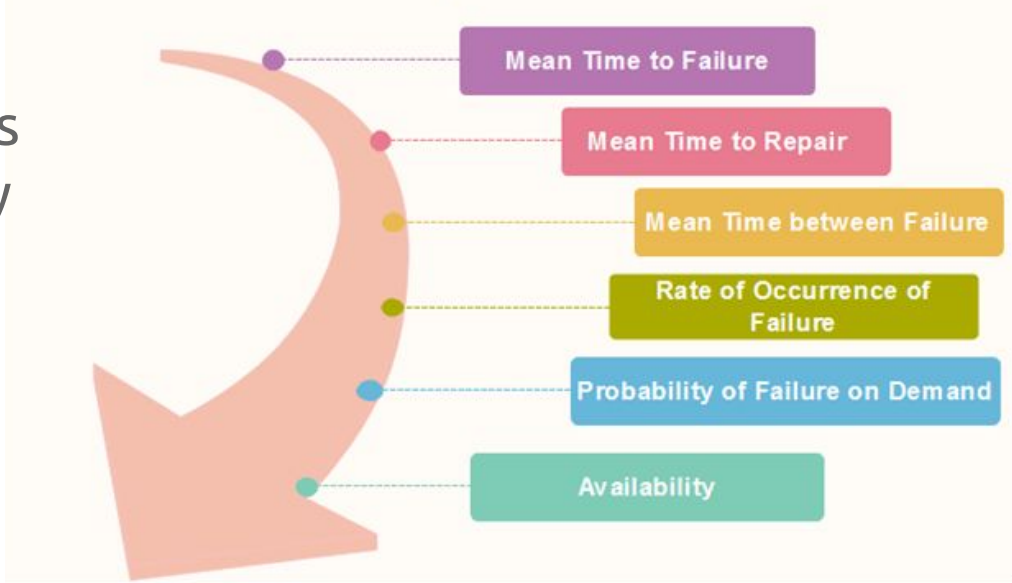
Lesson 09: Dependable systems

- Dependability properties
- Socio-technical systems
- Redundancy and diversity
- Dependable processes
- Formal methods and dependability



Lesson 10: Reliability engineering & Safety engineering

- Availability and reliability
- Reliability requirements
- Fault-tolerant architectures
- Programming for reliability
- Reliability measurement



Lesson 10: Reliability engineering & Safety engineering

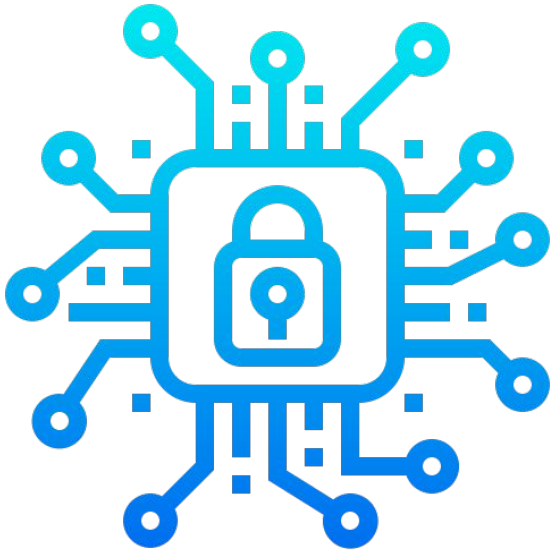
- Safety-critical systems
- Safety requirements
- Safety engineering processes
- Safety cases

Lesson 11: Security engineering & Resilience engineering

- Security and dependability
- Security and organizations
- Security requirements
- Secure systems design
- Security testing and assurance

Lesson 11: Security engineering & Resilience engineering

- Cybersecurity
- Socio-technical resilience
- Resilient systems design



Advanced Software Engineering

Part #03

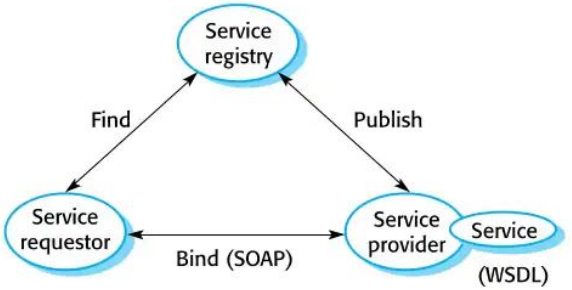
Lesson 12: Advanced Software Engineering

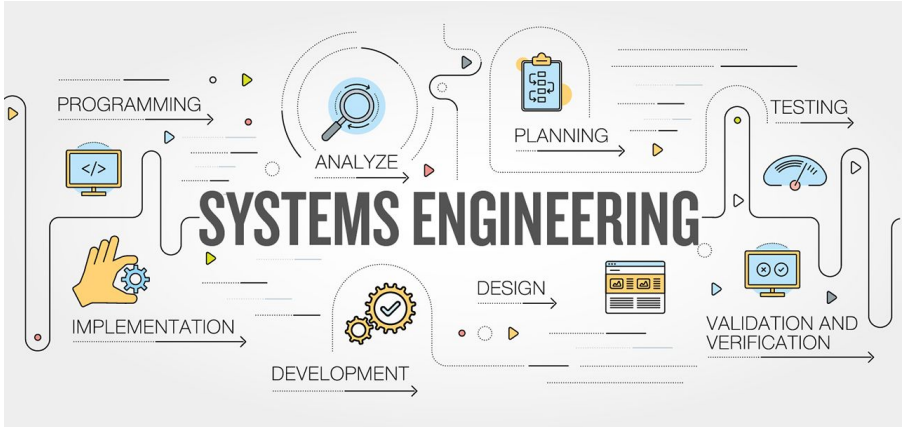
- Software reuse
- Component based software engineering
- Distributed software engineering

Lesson 13: Advanced Software Engineering

- Service-oriented software engineering

- Systems engineering

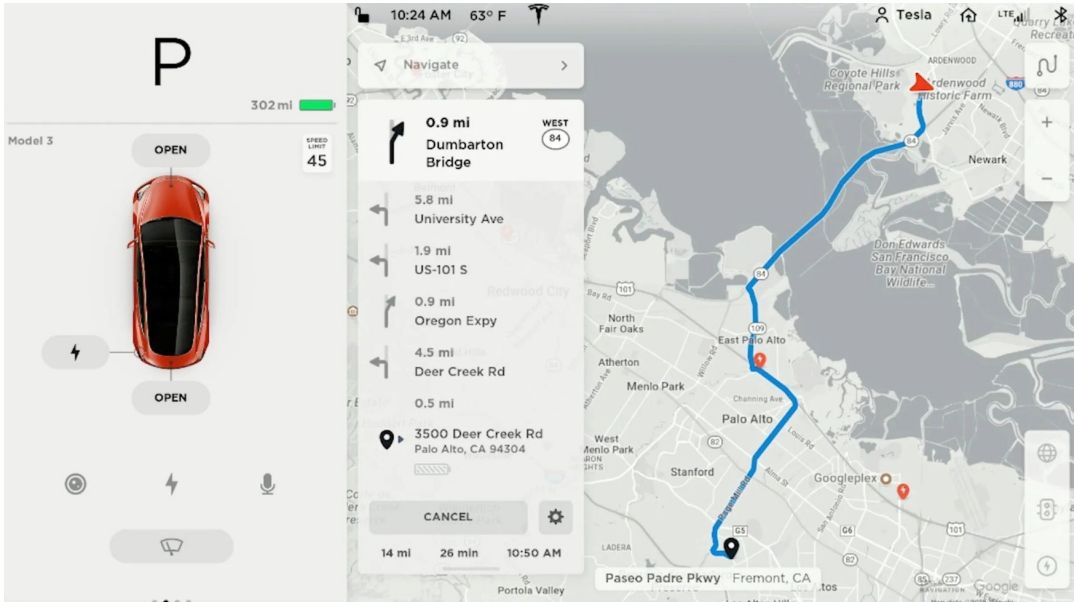




Course organization

Lesson 14: Advanced Software Engineering

- Systems of systems
- Real-time software engineering

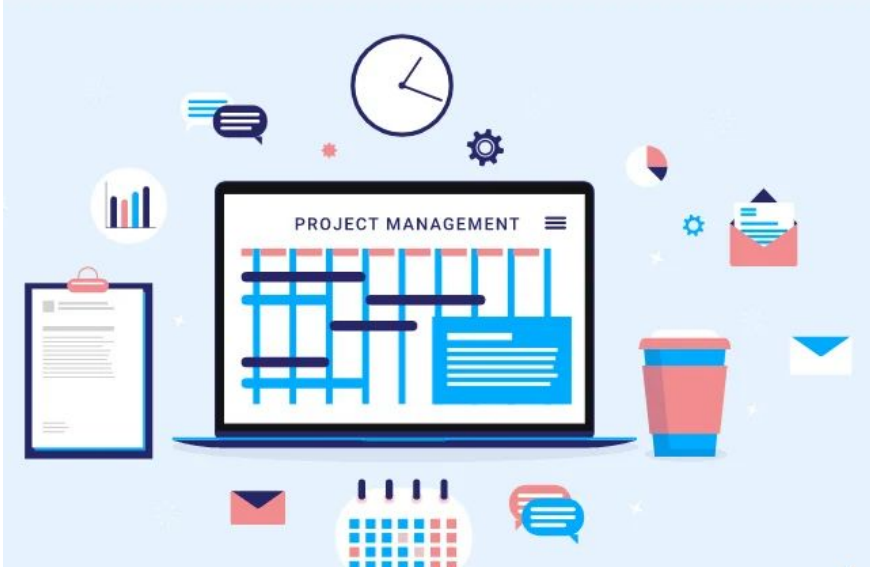


Software Management

Part #04

Lesson 15: Project management

- Project management
- Project planning
- Quality management
- Configuration management



Course organization

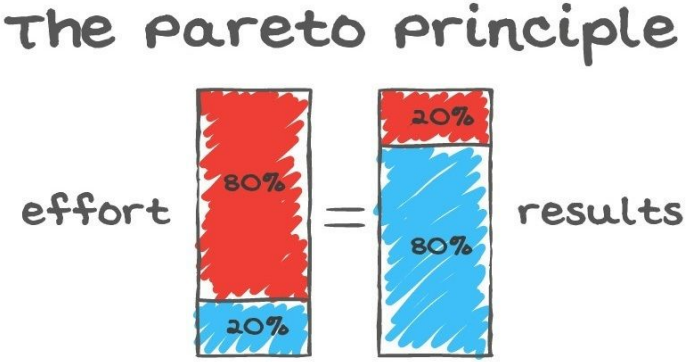
Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ES	MP
Lectures	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		**15**
Practices	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		**15**
Assignments		2		3		3		3		3		4		4			**22**
Midterm Exams							8										**8**
Final Exam																**40**	**40**
MP for EC	2	4	2	5	2	5	10	5	2	5	2	6	2	6	2	**40**	
																TOTAL	**100**

MP-Max. Points for the semester; MP for EC-Max. Points for each class; ES-examination session;

References:

Attendance is important, because

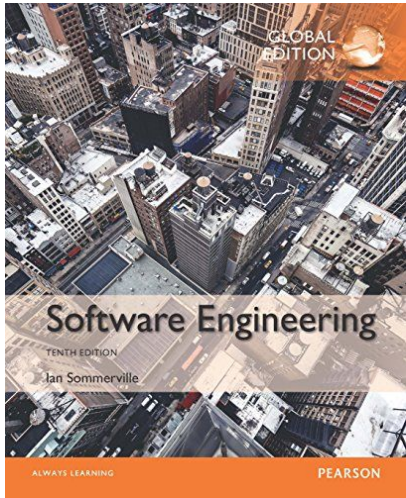
- only 20% of knowledge is available in the Slides
- remaining 80% of knowledge will be delivered through discussion, explaining, reviewing the topic



Course organization

References:

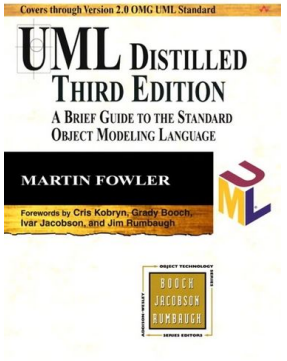
- Software Engineering, Global Edition, 10th edition, Ian Sommerville, 2016, Pearson;



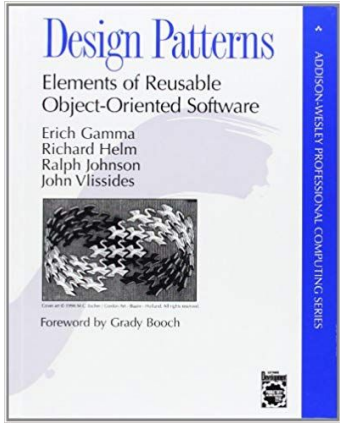
Course organization

References:

UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Martin Fowler, 2004, Addison-Wesley Professional;

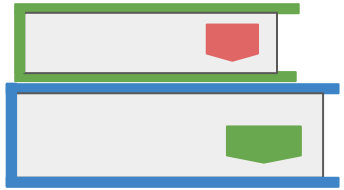


Design Patterns: Elements of Reusable Object-Oriented Software, 1st Edition, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 1994, Addison-Wesley Professional;

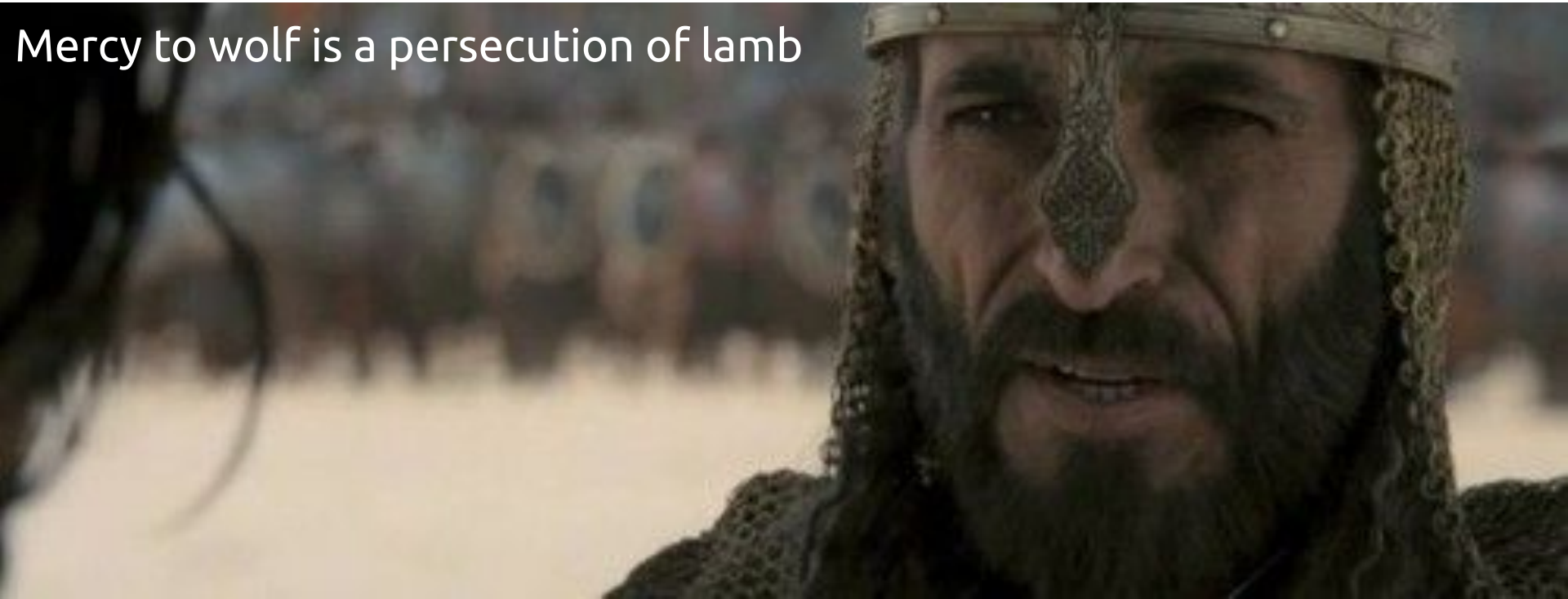


References:

- Several articles related to topics and online materials will be listed on the UNINET/WSP



Course organization



Course organization

- Ара сияқты болу

Course organization

- Ара сияқты болу

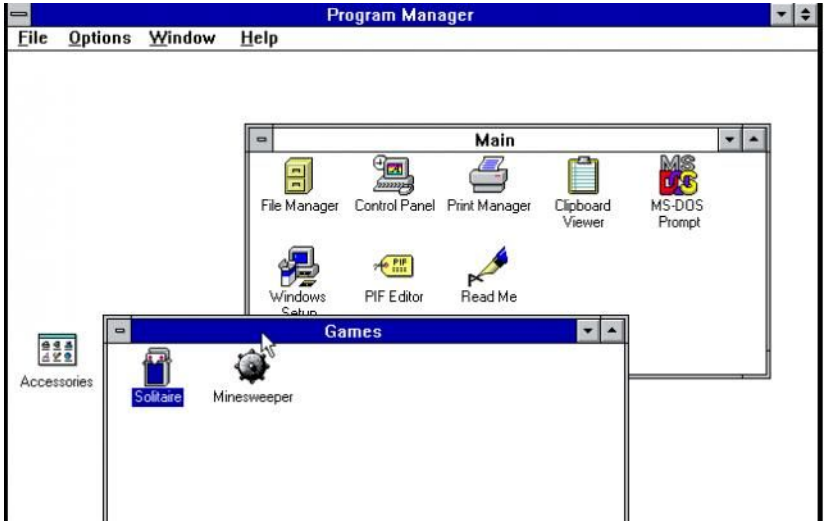
Agenda: Lesson #01 - Software Engineering - Lecture

1. Course overview
2. Course organization
3. **Course importance**
4. Introduction to Software Engineering

Course importance

PC Generation

- ~ 40 years old
- 1990 - 2000 years



Course importance

Internet Generation

- ~ 30 years old

- 2000 - 2010 years

Smartphone Generation

- ~ 20 years old

- 2010 - 2020 years

Internet of Things Generation

- ~ 10 years old

- 2020 - 2030 years

Course importance

Internet of Things Generation

- Ubiquitous & Cloud computing

Course importance

Internet of Things Generation

- DARQ

You are Engineer!

Everything that wasn't invented by God was invented by an engineer

Top 3 Questions from this course

1. WHY?

Why am I studying?

Top 3 Questions from this course

2. WHAT?

What am I studying?

Top 3 Questions from this course

3. HOW?

How am I studying?

Agenda: Lesson #01 - Software Engineering - Lecture

1. Course overview
2. Course organization
3. Course importance
4. **Introduction to Software Engineering**

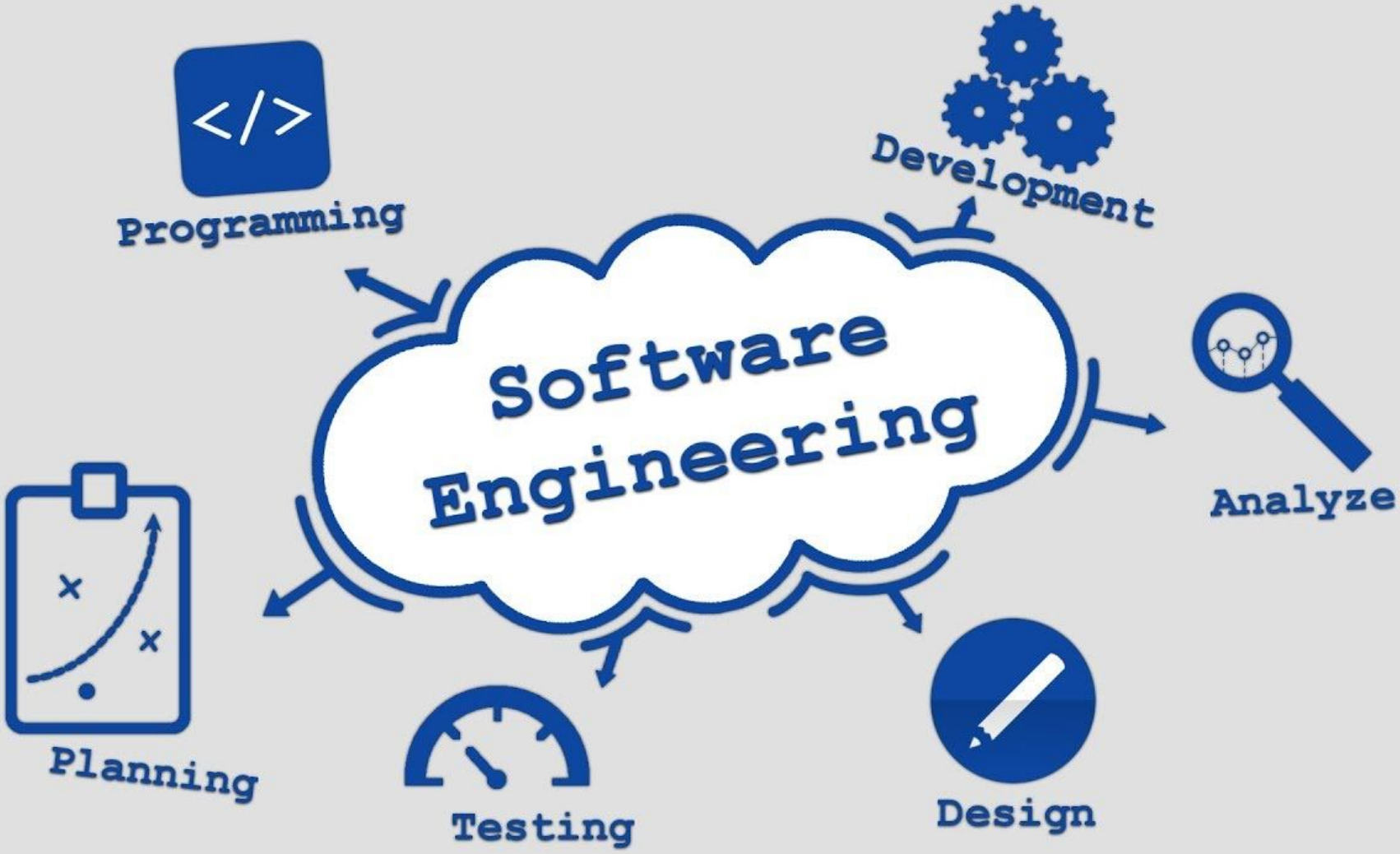
Introduction to Software Engineering

Introduction to Software Engineering

Part #01

Chapter 1 - Introduction

Part #01


Programming
Development
Software Engineering
Analyze
Planning
Testing
Design

Software engineering

Software engineering is essential for the functioning of government, society, and national and international businesses and institutions. We can’t run the modern world without software

Software engineering

National infrastructures and utilities are controlled by computer-based systems, and most electrical products include a computer and controlling software

Software engineering

The economies of all developed nations are dependent on software

More and more systems are software controlled

Software engineering is concerned with theories, methods and tools for professional software development

Expenditure on software represents a significant fraction of GNP in all developed countries

Software costs

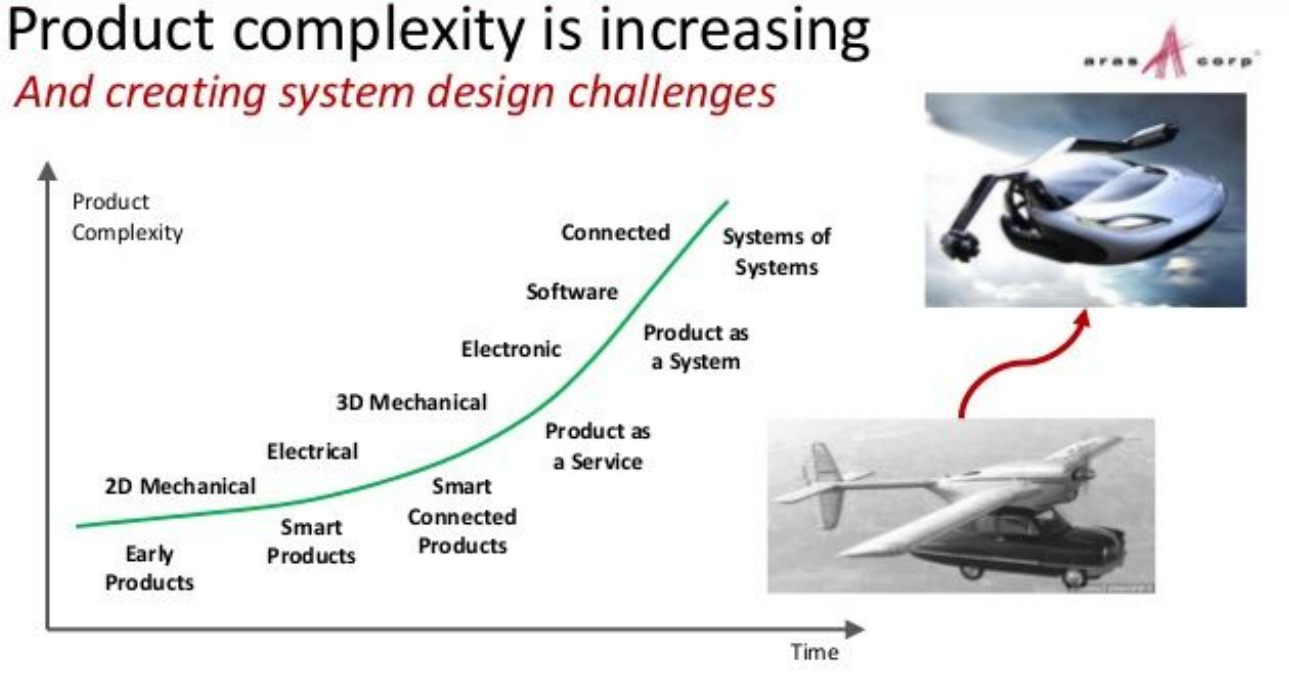
Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware costs

Software costs more to maintain than it does to develop software. For systems with a long life, maintenance costs may be several times development costs

Software engineering is concerned with cost-effective software development

Software Project failures

Increasing system complexity



Software Project failures

Failure to use software engineering methods

FAQ about software engineering

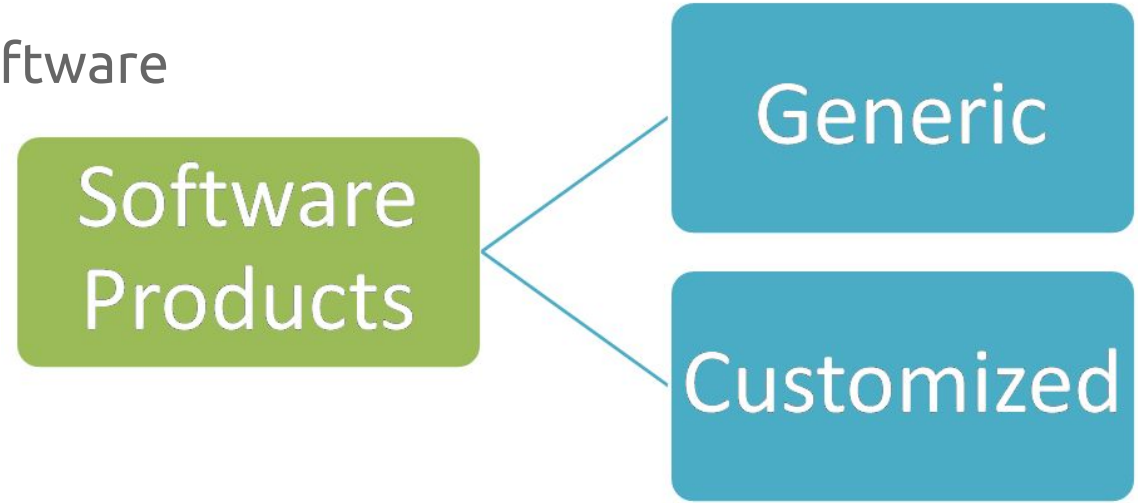
10 Questions to Introduce Software Engineering

https://www.youtube.com/watch?v=gi5kxGslkNc

Two kinds of software product

General Products/Software

Customized Products/Software



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

Essential attributes of Good Software

- Acceptability
- Dependability and security
- Efficiency
- Maintainability

Software Engineering diversity

- Stand-alone applications
- Interactive transaction-based applications
- Embedded control systems
- Batch processing systems

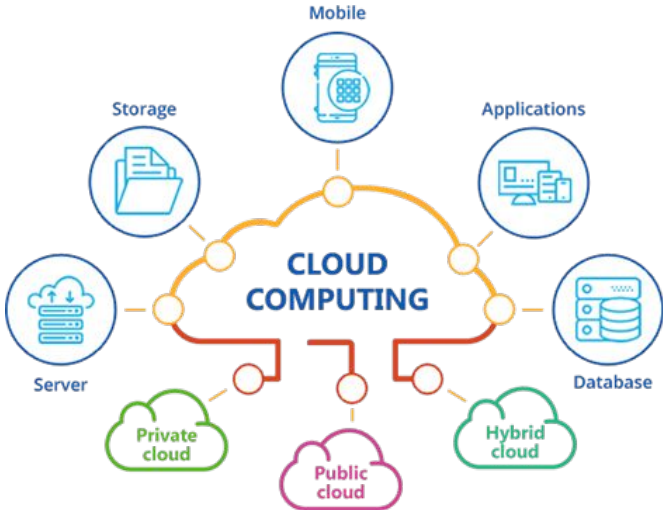
Software Engineering diversity

- Entertainment systems
- Systems of modeling simulation
- Data collection and analysis systems
- Systems of systems

Internet software engineering

The web is the platform for running the applications and organizations are increasingly developing web based systems rather than local systems

Cloud computing



Software engineering ethics

- Confidentiality
- Competence
- Intellectual property rights
- Computer misuse

Why software engineering

Why software engineering

https://www.youtube.com/watch?v=R3NzTt0BTWE

Agenda: Lesson #01 - Software Engineering - Lecture

Q & A