

Software Engineering

João Caldeira

Invited Professor

Email. joaocarlos.caldeira@my.istec.pt

Mob. +351 917769544

October 27, 2022



About Me

João Caldeira

- BSc – FCT / UNL (1997)
- MSc – FCT / UNL (2009)
- PhD – ISCTE-IUL (2021)
 - Assistant Professor (Invited since 2019)

Univ. Europeia / IPAM / UA Invited Professor

- Data Mining
- PG in Data Science for Business
- Forecasting & Time Series Analysis
- PG in Blockchain Technologies (Coordinator)

Kiion – www.kiion.com

- Founder
- Process/Data Science Advisor

<https://www.linkedin.com/in/joao-carlos-caldeira/>



João Caldeira
Founder & PADS Advisor
Kiion Technologies

joao.caldeira@kiion.com
+351 91 7769544



Planning

Semester 1

Software Engineering

Class	Date	Week	Summary	Tasks/Tools/Bibliography
1	26/Oct	1	Introduction Software Engineering	Software Engineering 10th Edition, Ian Sommerville Software Engineering Body of Knowledge (SWEBOK)
2	2/Nov	2	Software Engineering Process	
3	9/Nov	3	Software Engineering Process	
4	16/Nov	4	Requirements Engineering	
5	23/Nov	5	Requirements Engineering	
6	30/Nov	6	Software Design	
7	7/Dec	7	Software Design 1 st Assessment Task	
8	14/Dec	8	Software Construction	
9	4/Jan	9	Software Construction	
10	11/Jan	10	Testing & Validation	
11	18/Jan	11	Testing & Validation	
12	25/Jan	12	2 nd Assessment Task	
13	1/Feb	13	To be defined	
14	8/Feb	14	To be defined	
15	15/Feb	15	To be defined	

Software Engineering

October 27, 2022



Software Engineering

Some History

- **1960s**

- Struggling, as it was difficult to keep up with the hardware which caused many problems for software engineers
- Software was over budget, exceeded deadlines, required extensive de-bugging and maintenance, and unsuccessfully met the needs of consumers or was never even completed
- **Software crisis: Powerful machines/inadequate programming methods**

- **1968**

- **NATO** held the first **Software Engineering** conference where issues related to software were addressed
 - **Guidelines and best practices for the development of software were established**

Software Engineering

Some History

- **1984**

- **Software Engineering Institute (SEI)** was established headquartered on the campus of Carnegie Mellon University in Pittsburgh, Pennsylvania, United States
- **Watts Humphrey** founded the SEI Software Process Program, aimed at understanding and managing the software engineering process
- The Process Maturity Levels introduced would become the **Capability Maturity Model Integration for Development**(CMMI-DEV), which has defined how the **US Government** evaluates the abilities of a software development team.

- **Last decades**

- Generally accepted best-practices for software engineering have been collected by the ISO/IEC JTC 1/SC 7 subcommittee and published
 - **Software Engineering Body of Knowledge (SWEBOK)**
 - Software engineering is considered one of major computing discipline

Software Engineering

A few other important names

- **Charles Bachman (1924–2017)** is particularly known for his work in the area of databases
- **László Bélády (1928–2021)** the editor-in-chief of the *IEEE Transactions on Soft. Eng.* (1980s)
- **Fred Brooks (born 1931)** best known for managing the development of OS/360
- **Peter Chen (born 1947)** known for the development of entity–relationship modeling
- **Edsger W. Dijkstra (1930–2002)** developed the framework for a form of structured programming
- **David Parnas (born 1941)** developed the concept of information hiding in modular programming
- **Michael A. Jackson (born 1936)** software engineering methodologist responsible for JSP method *(with John Cameron)*; Problem Frames approach for analysing/structuring software development problems
- **Richard Stallman**, created the GNU system utilities and championed free software

Source: Wikipedia

Software Engineering

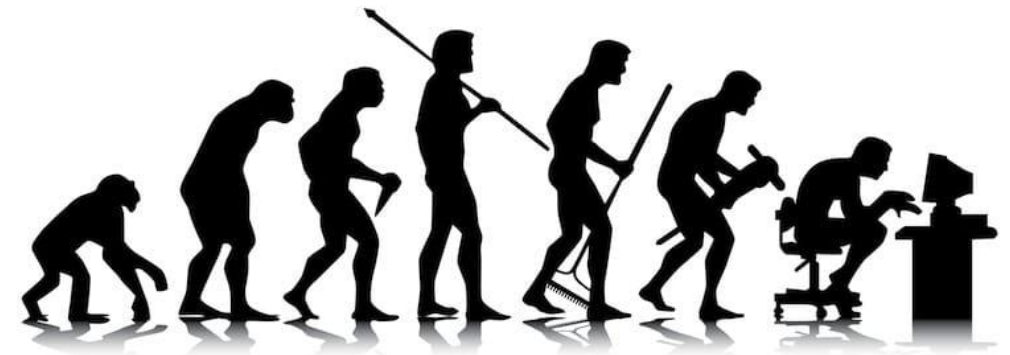
A Definition

- **What is software engineering ?**
 - ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB) defines software engineering as “the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software
 - In summary, it’s the **application of engineering to software**

Software Engineering

Evolution At-a-Glance

- ***“The application of engineering to software”***
- **It’s the management of a Process (Change / Evolution)**



Software Engineering

About Processes

“...All things -from the tiniest virus to the greatest galaxy- are, in reality, not things at all, but processes...”
--Alvin Toffler(1928-2016)

- **Software Engineering is a “Process”. Why ?**
 - It’s a structured set of activities for managing the creation of a new software product or maintain an existing one, including:
 - Requirement collection, analysis, design, coding, testing, and maintenance
- **What is a process ?**
 - A series of [**activities | steps | actions**] taken sequentially or in parallel in order to achieve a particular [**end | result | outcome**]*

Software Engineering

About Software Development

- Software systems are abstract and intangible
- Not constrained by the properties of materials, nor are they governed by physical laws
- Software can quickly become extremely complex, hard to understand, and expensive to change
- There are no universal notations, methods, or techniques for software engineering
- Different types of software require different approaches
 - Organizational information system or data warehouse
 - A controller for a scientific instrument
 - Graphics-intensive computer game

Software Engineering

Software projects often fail

- **Increasing system complexity**

- Systems have to be built and delivered more quickly
- Larger, even more complex systems are required
- Systems have to have new capabilities that were previously thought to be impossible

- **Failure to use software engineering methods**

- It's easy to write programs without using software engineering methods and techniques
- This software is often more expensive and less reliable than it should be
- **It requires better software engineering education and training**

Software Engineering

Professional Software Development

- **Generic products**

- Stand-alone systems produced by a development organization
- Sold on the open market
 - Apps for mobile devices, software for PCs such as databases, word processors, drawing packages, and project management tools
- Includes “vertical” applications designed for a specific market (SAP, Feedzai, Talkdesk, etc)

- **Customized software**

- Commissioned by and developed for a particular customer
- A software contractor designs and implements the software especially for that customer

Software Engineering

Software Diversity

- **Stand-alone applications**
- **Interactive transaction-based applications**
- **Embedded control systems**
- **Batch processing systems**
- **Entertainment systems**
- **Systems for modeling and simulation**
- **Data collection and analysis systems**
- **Systems of Systems**

Software Engineering

Internet Software Engineering

- 1. Software reuse became the dominant approach for web-based systems**
- 2. It is impractical to specify all the requirements for such systems in advance**
 - Web-based systems are always developed and delivered incrementally
- 3. Software may be implemented using service-oriented software engineering**
 - The software components are stand-alone web services
- 4. Interface development technology such as AJAX and HTML5 were adopted**
 - Support the creation of rich interfaces within a web browser (client side)

Software Engineering

Last decade trends

- **Mobile App Industry**
- **Continuous Integration/Continuous Deployment (CI/CD)**
- **DevOps**
- **Service Oriented Architecture (SOA)**
- **Low Code / No Code Paradigms**
- **Global Software Development (GSD)**

Software Engineering Process

A good software product

Characteristic

Acceptability

Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use

Dependability and Security

Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Software has to be secure so that malicious users cannot access or damage the system

Efficiency

Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, resource utilization, etc

Maintainability

Software should be written in such a way that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment

Software Engineering

Ethics in Software Engineering

- **Confidentiality**
- **Competence**
- **Intellectual property rights**
- **Computer misuse**

Software Engineering Process

October 27, 2022



Software Engineering

Set of activities

- **Requirements**

- End-User / customer business needs

- **Specification**

- The functionality of the software and constraints on its operation must be defined

- **Design & Implementation**

- The functionality of the software and constraints on its operation must be defined

- **Verification & Validation**

- The software must be validated to ensure that it does what the end-user / customer wants

- **Evolution**

- The software must evolve to meet changing customer needs

Software Engineering Process

Generic Software Process Models

- **The waterfall model**

- The fundamental process activities of specification, development, validation, and evolution are represented as separate process phases such as requirements specification, software design, implementation, testing, etc...

- **Incremental development**

- This approach interleaves the activities of specification, development, and validation
- The system is developed as a series of versions (increments), with each version adding functionality to the previous version

- **Reuse-oriented software engineering**

- This approach is based on the existence of a significant number of reusable components
- Focuses on integrating components into a system rather than developing from scratch

Software Engineering Process

Product vs. Process

Renoir's Process



Renoir's Product



Pollock's Process



Pollock's Product

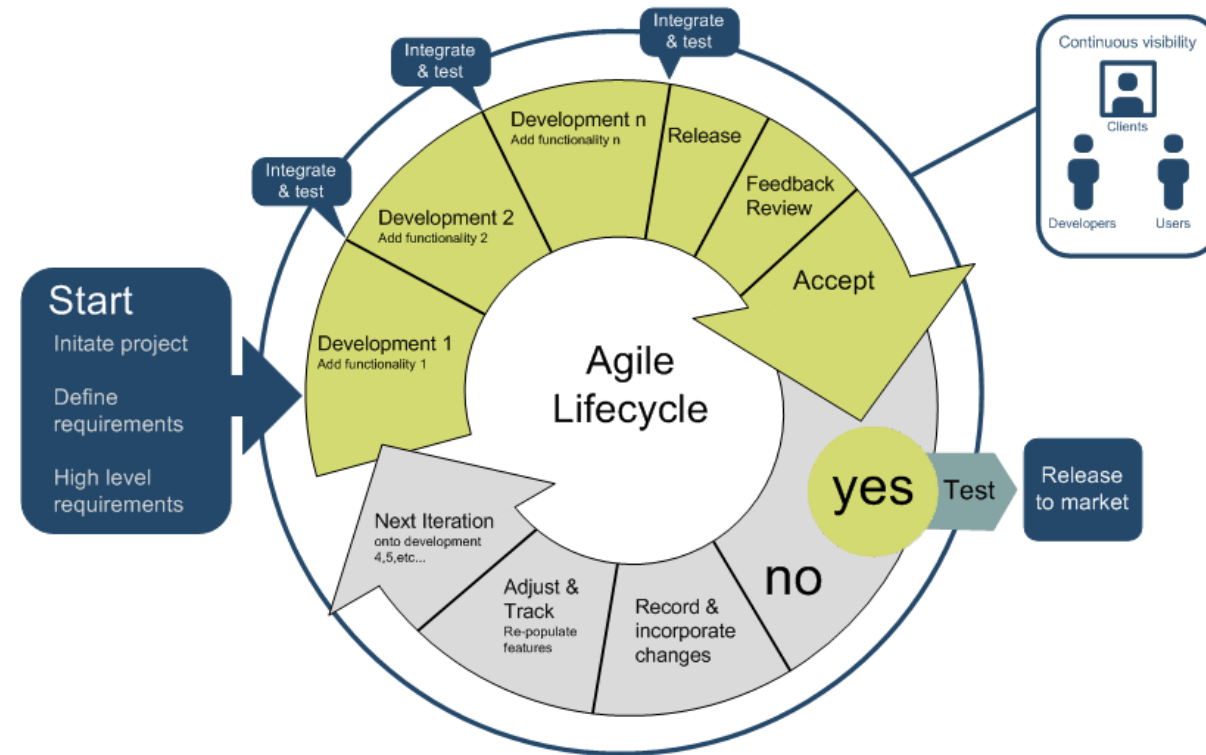


Pollock's Process (like) Product

Software Engineering Process

Software Development Metodologies - Agile

- **It's all about agility**
- **Fast implementation**
- **Succeed/Fail quickly**
- **Assess results & start again**



Software Engineering Process

Product vs. Process



Quick Overview

	Product	Process
Concept	The product is the final result of a development cycle	The process is a sequence or set of steps that should be followed to create a product
Focus	Product development focus is on final outcome	The process focuses on each step to be followed during software product development
Life	A product life cycle tends to be short term	A process life cycle is generally long term
Goal	Goal of product development is to finish the work and deliver the product successfully	The main goal of a process is to make a good quality products

Software Engineering

João Caldeira

Invited Professor

Email. joaocarlos.caldeira@my.istec.pt

Mob. +351 917769544

October 27, 2022

