

CSC 7003 : Foundations of Software Engineering

J Paul Gibson, D311

paul.gibson@telecom-sudparis.eu

<http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC7003/>

Introduction to Software Engineering

<http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC7003/L0-Introduction.pdf>

Evaluation :

- To be decided

Objectives :

- Capacity to identify and describe the software life cycle, roles, artefacts, and activities.
- Understand the concepts of software "best practices" and when they apply.
- Be able to adapt a software development process to ones needs and select an appropriate set of best practices that will guide you in completing a software development project.

Keywords : Discipline, Professionalism, Understanding, Fundamentals

Prerequisites : None (except basic programming experience)

Program :

To understand the true nature of software and appreciate that software should be engineered in a disciplined fashion, following professional standards.

To realise that engineering software has similarities to other engineering disciplines, but that there are aspects of engineering software which are unique to that discipline.

Place current (and future) technologies in software engineering into the context of software development techniques and tools that have appeared throughout the history of the discipline.

Provide the ‘big picture’ of software engineering so that students can progress to studying specialist techniques/tools/methods

Material to Cover

- History of Software Engineering
- The Nature of Software : modelling and abstraction
- Software Process Lifecycle : analysis, requirements, design, implementation, testing, maintenance
- Rigour and formality : Specification, validation, verification and testing
- Software Quality and Software Process Improvement : international standards
- Project Management : roles in software development teams
- Ethics and Software Engineering as a Profession
- Software Case Studies – successes and failures
- Current state of the art in software development and Leading Research Projects



Teaching-CSC7003 for Dr J. Paul Gibson, INFormatique (INF), Telecom SudParis, France.

Foundations Of Software Engineering

The material will be uploaded dynamically: the teaching approach is based on PBL and much of the learning will be through interaction/group work during the assigned lecturing time. (Please check the website for updates before every lecture.)

Assessment

TO BE DECIDED

Books

There are no course texts but the following books have inspired some of the material included in the course and provide additional information that you may find useful.

[Fundamentals of Software Engineering](#), Ghezzi, Jazayeri and Mandrioli, Prentice Hall

[Software Engineering](#), Ian Sommerville, Addison Wesley.

[Software Engineering: Principles and Practice](#), Hans van Vliet, John Wiley & Sons.

[The Mythical Man-Month](#), Frederick P. Brooks, Jr., Addison-Wesley Longman Publishing Co.

Sessions

Sessions are a mix of problem-based learning, group project work, directed practicals, interactive lectures and traditional lectures. There is no preset format - the lecturer organises the style and content of each session depending on the needs of the class.

Session 1: Wednesday 4th January (13h45, B07) - Introduction

Lecture Slides

Introduction To Software Engineering [pdf](#),

Software Engineering and Ethics: when code goes bad [pdf](#),

Some PBL : The balance problem [pdf](#),

Additional reading material

Software Reuse and Plagiarism: A Code of Practice, J Paul Gibson, 2009: [pdf](#)

Useful Links

[What makes a Great Software Engineer?](#) [Software Engineering Awards](#) [Software Engineering Institute \(SEI\)](#) [Joel On Software](#)

[Wikipedia - Balanced ternary](#) [Doug Jones - Ternary Arithmetic](#) [Rosetta Code - Balanced ternary](#) [History of ternary](#)