



LECTURE 1

# SOFTWARE ENGINEERING

COURSE 2021/2022



## OUTLINE

1. Current state and a bit of SE History
2. Professional Software Development
3. Ethics in Software Engineering?

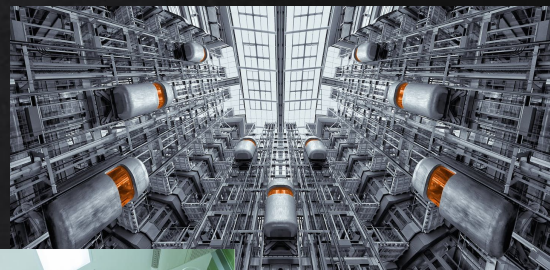


# CURRENT STATE AND A BIT OF SE HISTORY

Where are we?



# MODERN SOCIETY CAN NOT COPE WITHOUT SOFTWARE SYSTEMS



# SOFTWARE IS NOT TANGIBLE AND IS NOT CONSTRAINED BY THE LAWS OF PHYSICS

VERY QUICKLY THEY BECOME:

- LARGE AND COMPLEX
- DIFFICULT TO UNDERSTAND
- EXPENSIVE TO MAINTAIN

**STRINGENT NEED FOR:**

- FAST DELIVERY
- INCREASING NEW COMPLEX FEATURES
- LESS EXPENSIVE
- HIGH QUALITY
- DEAL WITH CHANGE
- MORE RECENTLY, SUSTAINABILITY



# SOFTWARE CRISIS



# SOFTWARE ENGINEERING FIRST PROPOSED IN 1968

At the NATO conference held to discuss what was then the “Software Crisis”

Individual approaches to program development did not scale-up to large and complex software systems

- unreliable

- costed more than expected

- were delivered late



## IN THE 80's

Frederic Brooks, Turing Award

"No Silver Bullet – Essence and Accident in Software Engineering"

**Accidental Complexity** – problems which engineers create and can fix (e.g. writing and optimizing code)

**Essential Complexity** – caused by the problems to be solved (e.g. if there is a list of desired functionalities for the system, this list must be implemented). This is unavoidable.





## IN THE 80's

Shrinking all the accidental activities to zero will not give the same order-of-magnitude improvement as attempting to decrease essential complexity

There is no silver bullet to solve the crisis

Also wrote the famous book: ***"The Mythical Man-Month: Essays on Software Engineering"*** on software engineering and project management with the central theme that **"adding manpower to a late software project makes it later"**

# THERE IS NO SILVER BULLET TO SOLVE THE CRISIS. BY THEN...

## Candidate Silver Bullets

- High-Level Languages Advancements
- Expert Systems
- Automatic Programming
- Graphical Programming
- Program Verification
- Environment and tools
- Workstations

## Candidate Silver Bullets for the Essence

- Buy versus build
- Requirements refinement and rapid prototyping
- Incremental development
- Great Designers

# THERE IS NO SILVER BULLET TO SOLVE THE CRISIS. BY THEN...

## Candidate Silver Bullets

- High-Level Languages And Frameworks
- Expert Systems
- Automatic Programming
- Graphical Programming
- Program Verification
- Environment and tools
- Workstations

**Abstraction**

**Code Generation**

**Abstraction/  
Modelling**

## Candidate Silver Bullets for the Essence

- Buy versus build
- Requirements refinement and prototyping
- Incremental development
- Great Designers

**Commercially  
off-the-shelf**

**Methodology  
And  
Incrementality**

**Legend – \*Nowadays we call it**



G. Booch



I. Jacobson



J. Rumbaugh

90's

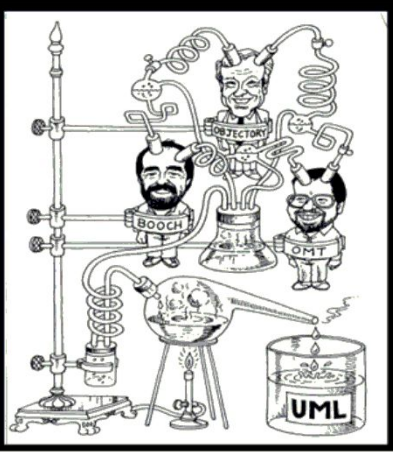


Introduced UML as a standard to visualize a system's architectural blueprints in a diagram

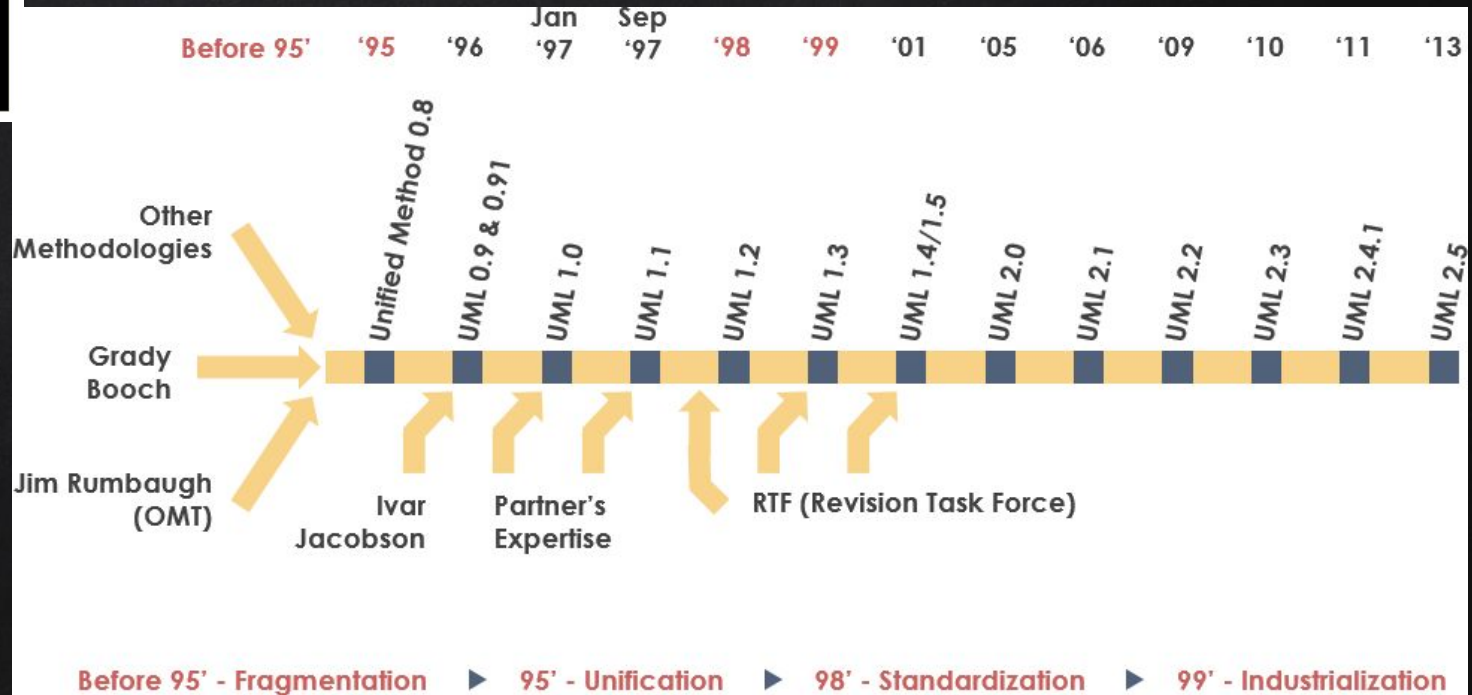
- Modelling for reasoning with systems abstractions







90's



90's

CHAOS report – Standish Group Report

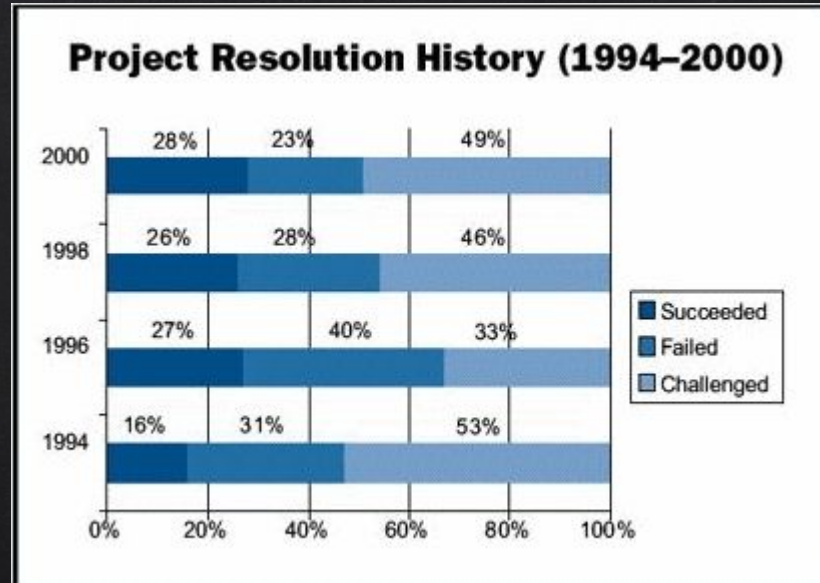
**Controversial** report that had impact in the SE community.

Observed industry software projects in the US in terms of:

- Size and cost
- Quality
- Delayed delivery

Highlights methodological/Process and  
project Management problems

# SINCE 1994 CHAOS REPORT – STANDISH GROUP



Success is defined as when the project has the following conditions: project teams are reasonably on time, on budget, and are building something to specification

# OO's

## Agile Development Process

- Manifesto 2001 by 17 experts





HOW ABOUT NOWADAYS?

# 2013 SURVEY 173 RESPONDENTS (ANALYSED BY DIFFERENT PARADIGMS)

How successful are software delivery teams?



**Successful** - A project is considered successful if a solution has been delivered and it met its success criteria within a range acceptable to your organization.

**Challenged** - A project is considered challenged if a solution was delivered but the team did not fully meet all of the project's success criteria within acceptable ranges (e.g. the quality was fine, the project was pretty much on time, but ROI was too low).

**Failed** - The project team did not deliver a solution.

Agile, lean and iterative strategies were superior on average compared to traditional and ad-hoc strategies.

Source: 2013 IT Project Success Rates Survey, [Amblysoft.com/surveys/success2013.html](http://www.amblysoft.com/surveys/success2013.html)  
Copyright 2014 Scott W. Ambler + Associates

SCOTT AMBLER  
+ Associates

\*This study rebuts some of the CHAOS report claims: <http://www.amblysoft.com/surveys/success2013.html>

Software Engineering 2021/2022

# 2015 CHAOS REPORT – STANDISH GROUP

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

**Successful:** delivered on time, on budget, with a satisfactory result

**Challenged:** late, overbudget, and/or with less than the required features

**Failed:** Canceled prior to completion, or delivered, but never used

The top 5 factors found in successful projects are:

1. **User Involvement**
2. Executive Management Support
3. Clear Statement of Requirements
4. **Proper Planning**
5. Realistic Expectations



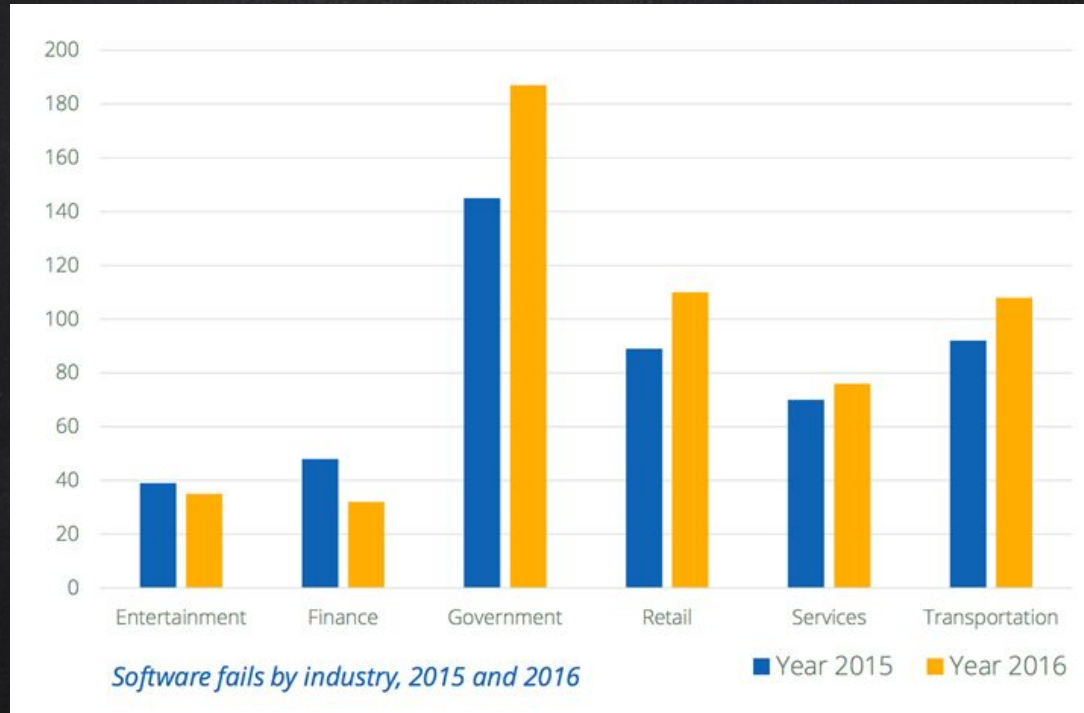
The top 5 indicators found in “Challenged” projects are:

1. Lack of User Input
2. Incomplete Requirements & Specifications
3. **Changing Requirements & Specifications**
4. Lack of Executive Support
5. Technical Incompetence

# Top factors found in “Failed” projects

1. Incomplete Requirements
2. Lack of user involvement
3. Lack of Resources
4. Unrealistic Expectations
5. Lack of Executive Support
6. Changing Requirements & Specifications
7. Lack of Planning
8. Didn't Need it Any Longer
9. Lack of IT management
10. Technical Illiteracy

# WE FAIL TOO OFTEN...



An increasing number of software fails is making it to the news. Source:  
<http://www.softwaretestingnews.co.uk/biggest-software-fails-2016>

# Serviços de Registo tiveram "falha geral" nos sistemas informáticos

17 de setembro de 2021

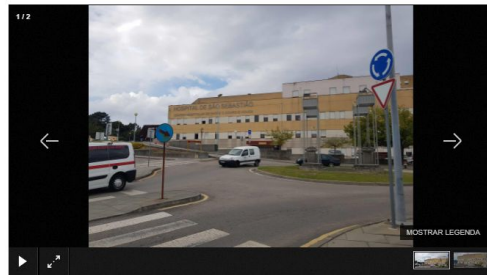


Os serviços de Registo sofreram hoje "uma falha geral" nos seus sistemas informáticos, o que "condicionou o efetuado pelos profissionais e atrasou ainda mais o atendimento dos utentes", revelou a Plataforma Sindical dos Registos, falando num "cenário torçoso e mundista".

IN Direto Nacional Local Justiça Mundo Economia Desporto Pessoas Inovações

Santa Maria da Feira

## Falha de rede informática no Hospital S. Sebastião obriga a reagendar consultas



Salomão Rodrigues  
17 Setembro 2020 às 16:29



O sistema de rede informática do Hospital S. Sebastião, na Feira, está, desde ontem, com intermitências no funcionamento obrigando a que algumas consultas externas fossem reagendadas. Os serviços Partilhados do Ministério da Saúde



Portugal Mundo Noticiários Desporto Futuro Programas

JUSTIÇA

## Falhas no sistema informático dos tribunais provocam adiamento de julgamentos

O Sindicato dos Funcionários Judiciais adianta que vários julgamentos estão a ser adiados devido a uma falha informática. As alterações ao sistema têm sido feitas em várias fases.



Últimas Mercados Obsessões Prova dos 9 Opinião Advocatus Pessoas ECOseguros

Segurança Social

## Falha informática provoca corte no apoio a 1.800 desempregados


ECO  
18 Junho 2021





Falha informática provocou corte no apoio extraordinário ao rendimento dos trabalhadores a 1.800 beneficiários. Há também 2.800 pessoas que receberam valores acima do previsto, que terão que devolver.



# LAST WEEK ...




VIDEOLIVESHOWSCORONAVIRUS



## UK airports report delays as IT failure shuts e-gates

*Travelers arriving at U.K. airports are facing delays after electronic passport gates have failed, forcing passengers to wait in line for border officers to inspect their documents*

By The Associated Press  
24 September 2021, 15:14 • 1 min read



LONDON -- Travelers arriving at U.K. airports faced delays Friday after electronic passport gates failed, forcing passengers to wait in line for border officers to inspect their documents.

London Heathrow, Manchester and Edinburgh airports all reported that a technology failure disrupted gates that allow passengers with biometric passports to quickly pass through border controls without speaking to immigration officers.



# PROFESSIONAL SOFTWARE DEVELOPMENT

Where are we?

# WHY DO WE FAIL COMPARING TO OTHER ENGINEERING?

## Software Engineering

- Project Planning and Software Modelling is still not common practice
  - (there is still the attitude program first and make questions afterwards)
- There is no physical product, therefore It is expected to evolve as the context/requirements change too often along the life-cycle

## Other Engineering Disciplines

- Blueprints for detailed pre-design
- The product is physical, therefore evolution is rare
  - (We don't see weekly upgrades to a bridge)

# WHY DO WE FAIL COMPARING TO OTHER ENGINEERING?

Besides...

The Software Engineer has to deal with

Problem Domain complexity (typically multidisciplinary)

Solution Domain Complexity  
(technological stack involved to implement the solution)





# WHAT IS SOFTWARE?

SOFTWARE\* = Code + Specification + configuration + manuals + libraries  
+ support websites + maintenance

\*we are only interested in software that is not just for yourself  
(amateur) but for other people to use and engineers to change or  
maintain

# WHAT ARE THE ATTRIBUTES OF GOOD SOFTWARE?

- Delivers the required functionality
- Performance to the user
- Maintainable
- Dependable
- Usable
- Sustainable

# WHAT IS SOFTWARE ENGINEERING? (WHAT TO DEAL WITH)

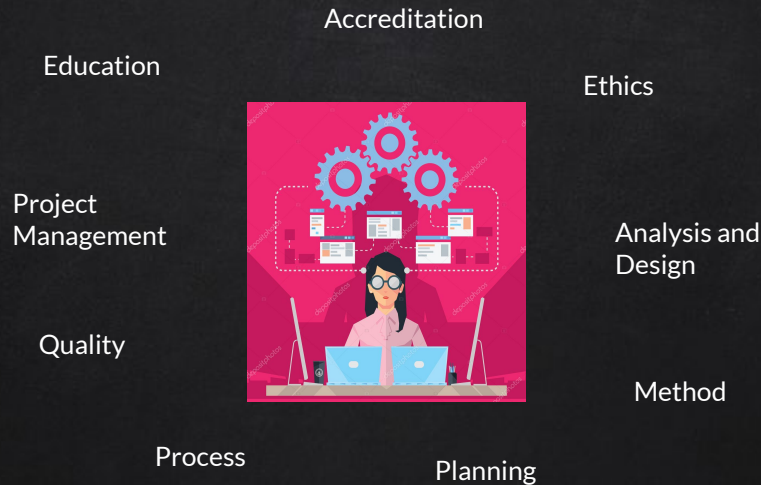
The discipline concerned with all aspects of software production from initial conception looking to cope with:

- increasing diversity
  - complexity of the problem domain (e.g. Finance, Government)
  - complexity of the solution domain (e.g. technological stack and integration)
- demands for reduced delivery times and cost (including operation, maintenance and evolution)
- developing trustworthy software

# WHAT IS A SOFTWARE ENGINEER? (HOW TO DEAL WITH)

Is the software professional that **applies the scientific method and engineering discipline** for

- Analysing and present solutions to (large scale and/or complex) engineering problems
- Using methodologies, technical knowledge, processes, and project management skills





# WHAT ARE THE MAIN SOFTWARE ENGINEERING ACTIVITIES?

- **Software Specification**
  - Customers and engineers define the software to be produced and constraints on its operation
- **Software Development**
  - Software is designed and programmed
- **Software Validation**
  - Software is checked to ensure that it is what the customer requires
- **Software Evolution**
  - Software is modified to reflect changing customer and market requirements

# CLARIFYING

USES

USES

## Systems Engineering\*

Concerned with all the aspects of development and evolution of complex systems where software plays a major role.

## Software Engineering

Concerned with practical problems of producing software

## Computer Science

Concerned with the theories and methods underlying computers and software systems

\*Also includes hardware development, policy and process design and deployment.  
Concentrates on the overall architecture and less on the system components (hardware, software, etc)

## MANY DIFFERENT TYPES OF SOFTWARE

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

# SO, ULTIMATELY... PROGRAMMER VS. SOFTWARE ENGINEER?

**Programmer** is tasked with creating the code that makes a program run

**Software Engineer** is responsible for designing, developing and implementing (or follow the software lifecycle) the software solutions programmers create.



3.



# ETHICS IN SOFTWARE ENGINEERING?

# A MATTER OF BEING RESPECTED AS SOFTWARE ENGINEER

Your job involves wider responsibilities that go beyond the scope of technical skills within a social and legal framework.

You must behave morally and ethically responsible way to be respected as professional engineer.

# DEONTOLOGY

Regulates the relations and behavior of the professionals with:

- Society
- Employer
- Clients
- Colleagues

# ACM/IEEE-CS JOINT TASK FORCE ON SOFTWARE ENGINEERING ETHICS AND PROFESSIONAL PRACTICES

1. PUBLIC – Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER – Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

3. PRODUCT – Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.

8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.



# PROFESSIONAL SOCIETIES AND INSTITUTIONS HAVE AN IMPORTANT ROLE TO PLAY IN SETTING ETHICAL STANDARDS.

ACM (Association for Computing Machinery) – <http://acm.org>

IEEE (Institute of Electrical and Electronic Engineers) – <http://ieee.org>

British Computer Society – <http://bsc.org>

Ordem dos Engenheiros – <http://ordemdosengenheiros.pt>

...

SEE YOU IN THE NEXT LECTURE



# REFERENCES

Software Engineering, by Ian Sommerville, Pearson, (either 9th or 10th edition)  
see chapter 1

'No silver bullet: Essence and accidents of software engineering'.

Introduction to the problems of software engineering.

F. P. Brooks, IEEE Computer, 20 (4), April 1987.

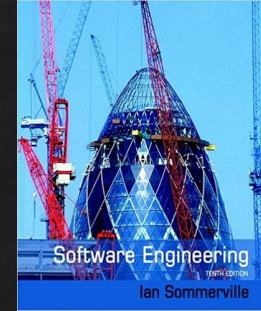
<http://doi.ieeecomputersociety.org/10.1109/MC.1987.1663532>.

'Software engineering code of ethics is approved'.

ACM/IEEE Code of Ethics

Comm. ACM, D. Gotterbarn, K. Miller, and S. Rogerson, October 1999.

<http://portal.acm.org/citation.cfm?doid=317665.317682>.



# PROFESSIONAL ENGINEERING ASSOCIATIONS / ORDERS

## PURPOSE

While the mission of Universities is to certify adequate education foundations for future professionals, the professional associations (or Professional Orders) certify that a given professional, properly trained by accredited institutions, has already demonstrated practical use of his engineering knowledge, with due quality, in real life projects and is conducting his engineering activity guided by engineering principles (updated longlife).

Besides, the professional associations warrant that the accepted members have deontological/ethical conduct and quality to address problems and can take responsibility within the scope of certain recognized acts for a given engineering discipline regulated by law.



# PROFESSIONAL ENGINEERING ASSOCIATIONS/ORDERS

## DEALING WITH MISCONDUCT

The professional orders have internal control mechanisms to sanction professional misbehaviour of their members, which include, in the limit, the legal means to expel members.

In many countries worldwide, a professional can not claim to have the title of Engineer and officially practise (or be responsible for) certain professional acts without being recognized by a professional association.

The fact that a person uses the title, is considered to be misleading for the society, as a person is claiming trust (as part of a group of professionals with a defined identity) certified by the professional association when not recognized as such. Not doing so, the professional can incur into legal infractions ultimately leading to court.

# PROFESSIONAL ENGINEERING ASSOCIATIONS/ORDERS SOCIETY

Other activities of the professional orders, besides contributing to the discussion of the education curricula, include to use the technical knowledge to contribute to improve society and serve as an independent technical voice (promoting technical discussion) to society's strategic decisions.

It also serves as consultant to courts, and takes actions to promote professional lifelong education/training, as well as promoting scientific discussion towards disciplinary advancement.

For more information, please consult:

[www.ieee.org](http://www.ieee.org)

[www.acm.org](http://www.acm.org)

[www.ordemdosengenheiros.pt](http://www.ordemdosengenheiros.pt)