Software Development Methods (MDS) 2019/2020

Vasco Amaral
vma@fct.unl.pt

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
Departamento de Informática

Module 1
Introduction

Vasco Amaral
vma@fct.unl.pt

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
Departamento de Informática

Acknowledgments

The current course materials are the result of evolving collective effort of former lecturers of the Software Development Methods (MDS) course. The slides had significant rework by Prof. Miguel Goulão last year and will be updated this year.

Vasco Amaral | Ana Moreira | João Araújo | Miguel Goulão

Team

Vasco Amaral, regent, t.1, p.1
P2/13
vma@fct.unl.pt
Office hours:

Catarina Gralha, p.6, p.7
123-A
acg.almeida@campus.fct.unl.pt
Office hours:

João Araújo, p.2, p.3, p.4, p.5
P2/4
p191@fct.unl.pt
Office hours:

Schedule

	2ª	3ª	4ª	5ª	6ª	Sábado
8:00 9:00						
9:00 10:00	**MDS** p.1 Ed 2: Lab 114/Ed.II	**MDS** p.3 Ed 2: Lab 112/Ed.II	**MDS** t.1 Ed 7: 1D/Ed.VII	**MDS** p.7 Ed 2: Lab 121/Ed.II	**MDS** p.2 Ed 2: Lab 121/Ed.II	
10:00 11:00	**MDS** t.1 Ed 7: 1D/Ed.VII	**MDS** p.4 Ed 2: Lab 112/Ed.II				
11:00 12:00			**MDS** p.6 Ed 2: Lab 120/Ed.II		**MDS** p.5 Ed 2: Lab 121/Ed.II	
12:00 13:00						
13:00 14:00						
14:00 15:00						
15:00 16:00						
16:00 17:00						
17:00 18:00						
18:00 19:00						
19:00 20:00						
20:00 21:00						
21:00 22:00						
22:00 23:00						
23:00 24:00						
	2ª	3ª	4ª	5ª	6ª	Sábado

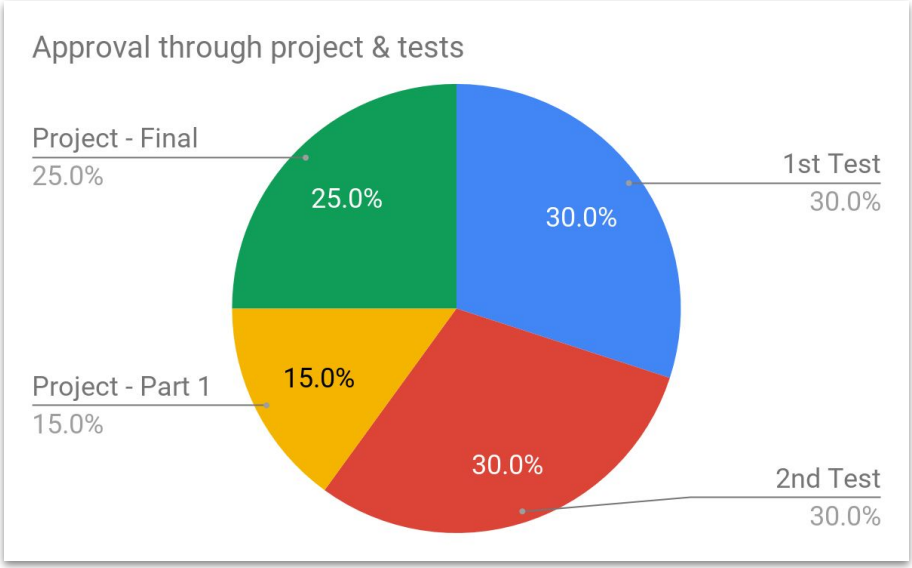
Evaluation rules (normal season)

Theory (60%, minimum of 9.5 for passing)

- 1st test (30%)
- 2nd test (30%)

Practice (40%, minimum of 9.5 for passing)

- Project part 1 (15%)
- Project part 2 (25%)



See fully detailed description on clip!

Evaluation rules (appeal exam season)

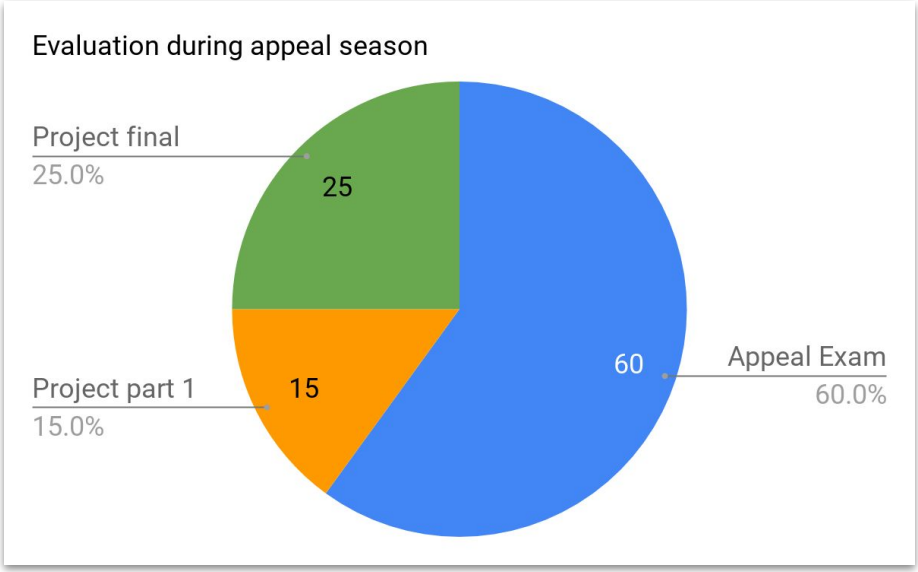
(also applicable to students who were approved this semester and wish to improve their grade)

Theory (60%, minimum of 9.5 for passing)

- Appeal exam (60%)

Practice (40%, minimum of 9.5 for passing)

- Project part 1 (15%)
- Project part 2 (25%)



See fully detailed description on clip!

What about students with a valid frequency obtained in a previous semester?

- Frequencies from the last 2 editions of the course are accepted
- In the end, the final grade is either the tests mean, or the exam grade)
- The project grade obtained in previous years is not used

Project evaluation

- Work in groups of 4 students
- Project elaborated outside the classes, using a software tool for project specification
  - **No manually written/drawn elements will be accepted for evaluation**
- The project is to be **incrementally enriched and improved throughout the whole semester**
- The final project is discussed in the last week of classes and during the evaluations week (during the labs hours).
- **No assignments are accepted beyond the established deadlines!**

Working students

- Working students must communicate their status and that they plan to use that status. To benefit of their status, they must officially inform the academic services.
- Working students must participate in exactly the same tests/exams and projects as full-time students

Important dates (temporary, to be fixed by next week)

01	Project assignment	2018/09/23-28(?)
02	Assignment first deadline	2018/10/23(?)
03	First test	2018/11/09
04	Assignment final deadline	2018/12/04(?)
05	Second test	2018/12/14

Course web page

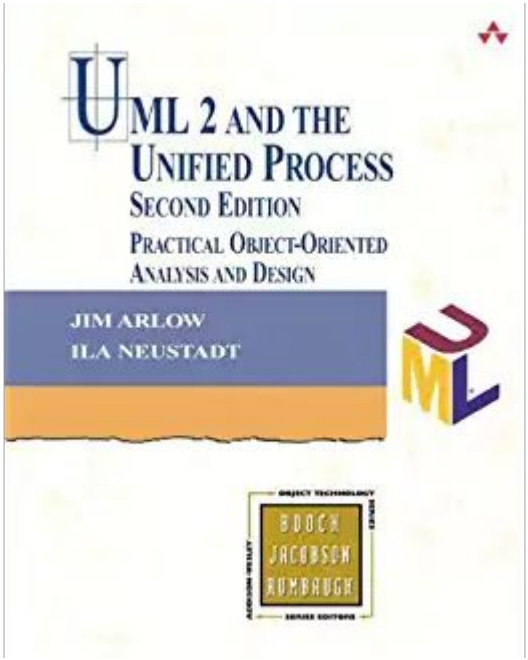
All course materials will be made available through Moodle.

https://moodle.fct.unl.pt/course/view.php?id=5462

Enrollment key : mds2020

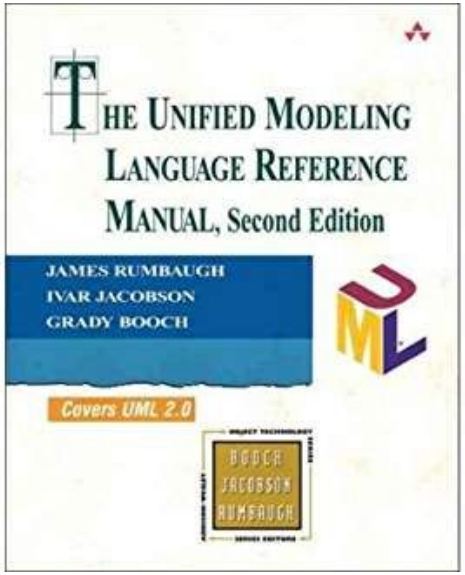
Bibliography [1/5]

UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (Second Edition),
Jim Arlow, **Ila Neustadt**,
Addison-Wesley, 2005.



Bibliography [2/5]

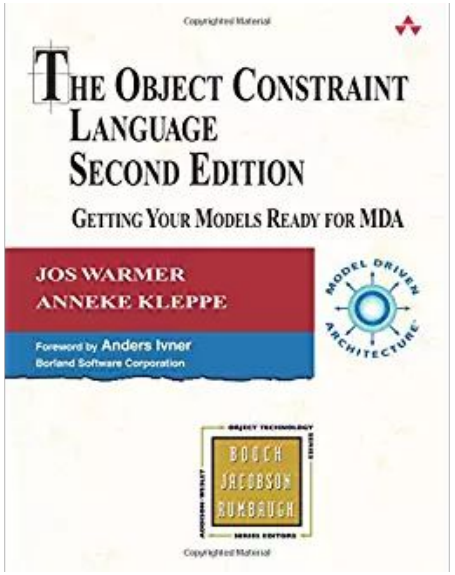
The Unified Modeling Language Reference Manual,
G. Booch, J. Rumbaugh & I. Jacobson,
Addison-Wesley, 2004.



Bibliography [3/5]

The Object Constraint Language: Getting Your Models Ready for MDA (2nd Ed.),

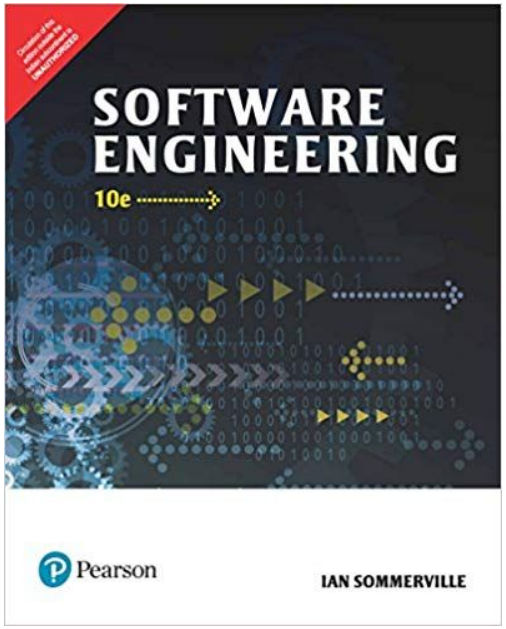
Anneke Kleppe, Jos Warmer,
Addison-Wesley, 2003.



Bibliography [4/5]

Software Engineering,
Ian Sommerville,
Addison-Wesley/Pearson

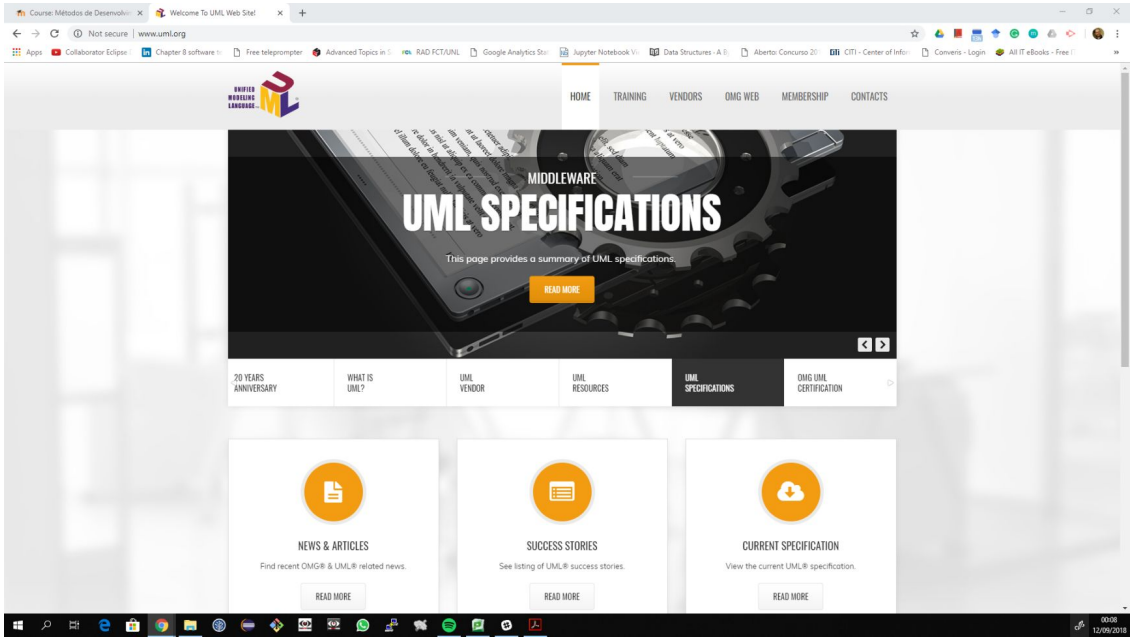
8th, 9th or 10th edition



Bibliography [5/5]

… and many, many more resources, including the official UML web site:

www.uml.org



Online UML quizzes

Business Informatics Group, Vienna University of Technology:

http://elearning.uml.ac.at/

Rules while in class

Lectures are supposed to be interactive.

Do ask questions. Stay focused. Learn.

Your mission

It is up to you to keep up with lectures and labs (in case you miss any of these, it is your job to find out what happened).

The class's contents, important messages provided during classes, and so on will not be repeated elsewhere.

Before emailing us...

Go to clip and look up the info you need.

Go to moodle and look up the info you need.

Make sure you actually were in class, paying attention, just in case the info you need was discussed there before.

If you still did not find what you need, please email us. :-)

Emailing us...

When contacting elements of the teaching staff by email, please **make sure your subject starts with [MDS] followed by the subject of the message**, and that the contents of the message contain your name, student number lab, and the subject you wanted to ask us about.

The answer may be provided by email, or in the next class, if it is of general interest to other students.

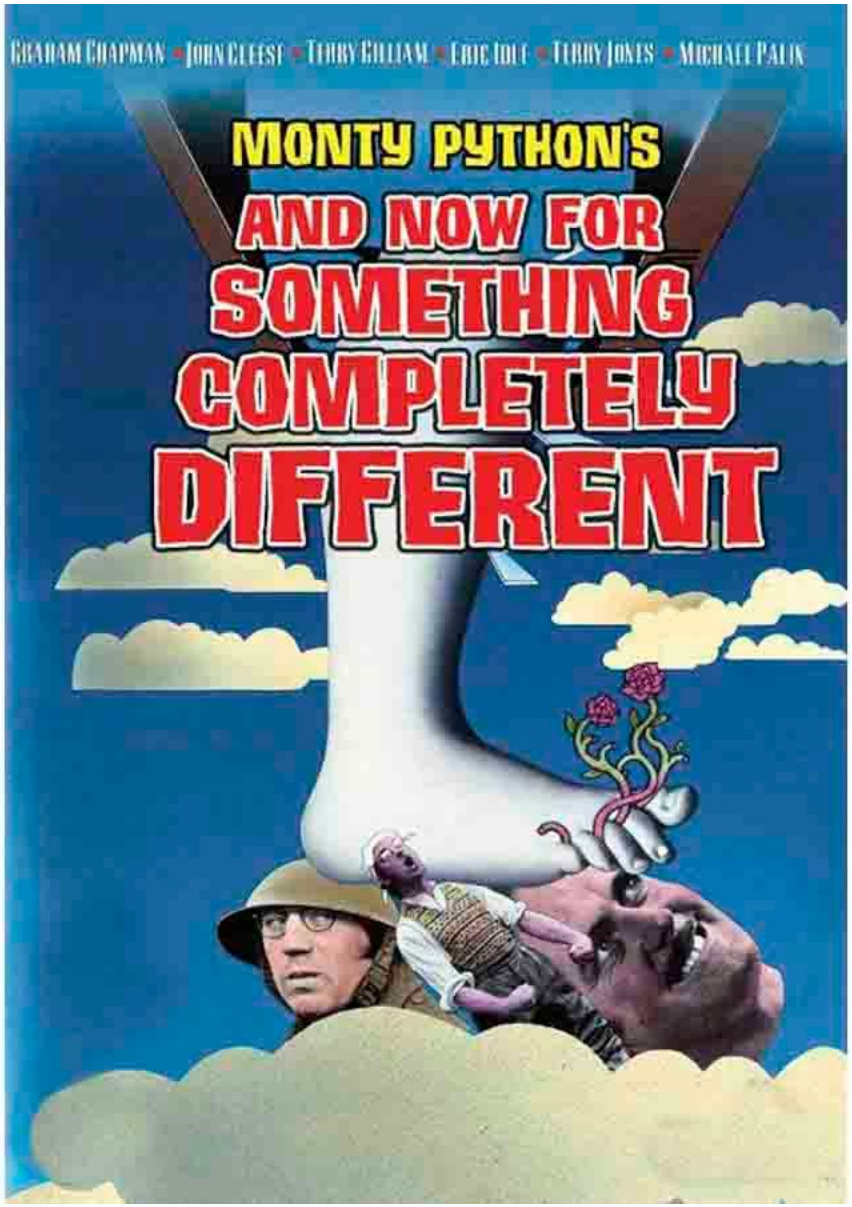
Even better than emailing us...

The primary means of communication is in person, during classes, office hours (please send an email in advance - **till the day before** - so that we know you are coming to see us there)

E-mails about technical

Warning

Any kind of fraud detected in a project, test, or exam, results automatically in failing the MDS course in this edition and cancels the frequency, regardless of whether it was gained in this edition of MDS, or during a previous edition. This failure prevents the student from going to the appeal exam, or any special season exam this year.

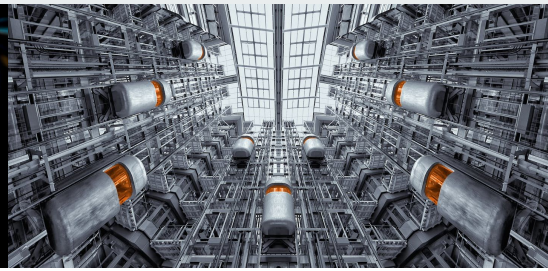

GRAHAM CHAPMAN • JOHN CLEESE • TERRY GILLIAM • ERIC IDLE • TERRY JONES • MICHAEL PALIN
MONTY PYTHON'S
AND NOW FOR
SOMETHING
COMPLETELY
DIFFERENT

Why do we have a Software Development Methods course?

Why are we here?

Read Chapter 1 of the Book
“Software Engineering”,
10th edition,
by Ian Sommerville



Modern society cannot cope without software systems

- Immersion of Software in all areas of human activity
  - Economy, Banking, Health, Aeronautics, Space Industry, R&D, etc…




Modern society cannot cope without software systems

- The great majority of electrical systems rely on computers and control software

Modern society cannot cope without software systems

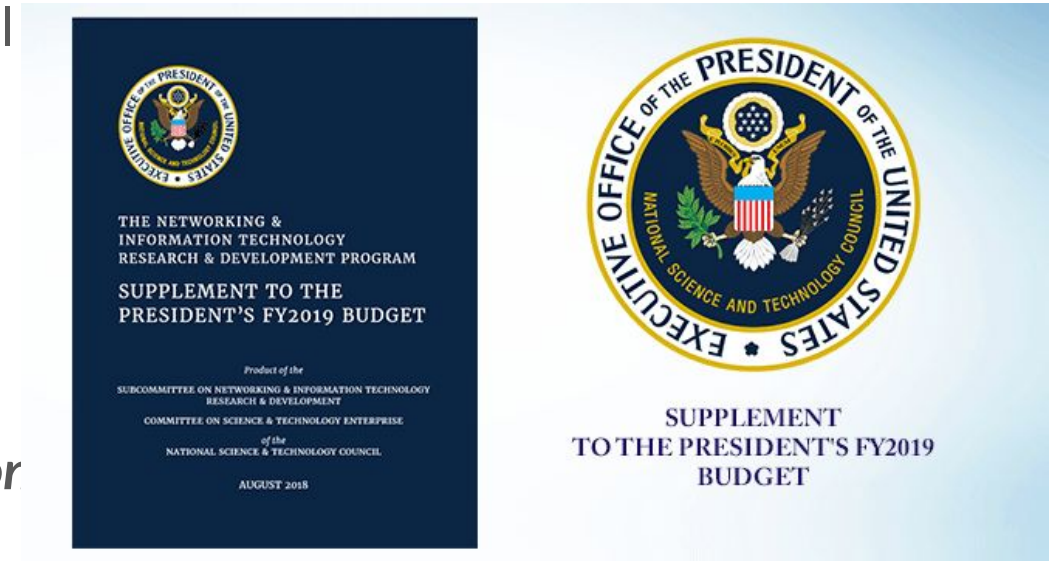
- Systems are getting **larger** and **more complex**
  - need to be delivered more **quickly**
  - must have new capabilities (thought to be impossible)
  - be **less expensive**
  - **reliable**



Software systems are politically under focus

According to USA Presidential Information Technology Advisory Committee (PITAC) Report 1999:

"Software is the new physical infrastructure of the information age"



Societal point of view

European Software Strategy

a NESSI Position paper 2008

flexibility, intelligence and security to all complex systems and equipment that support and control the ***key infrastructures of our society***

Economic point of view

European Software Strategy

a NESSI Position paper 2008

Increases ***productivity and competitiveness in all business activities***

Technologic point of view

European Software Strategy

a NESSI Position paper 2008

Traditional slip hardware/software will disappear *with blurred frontier between the computer, the network and the application*

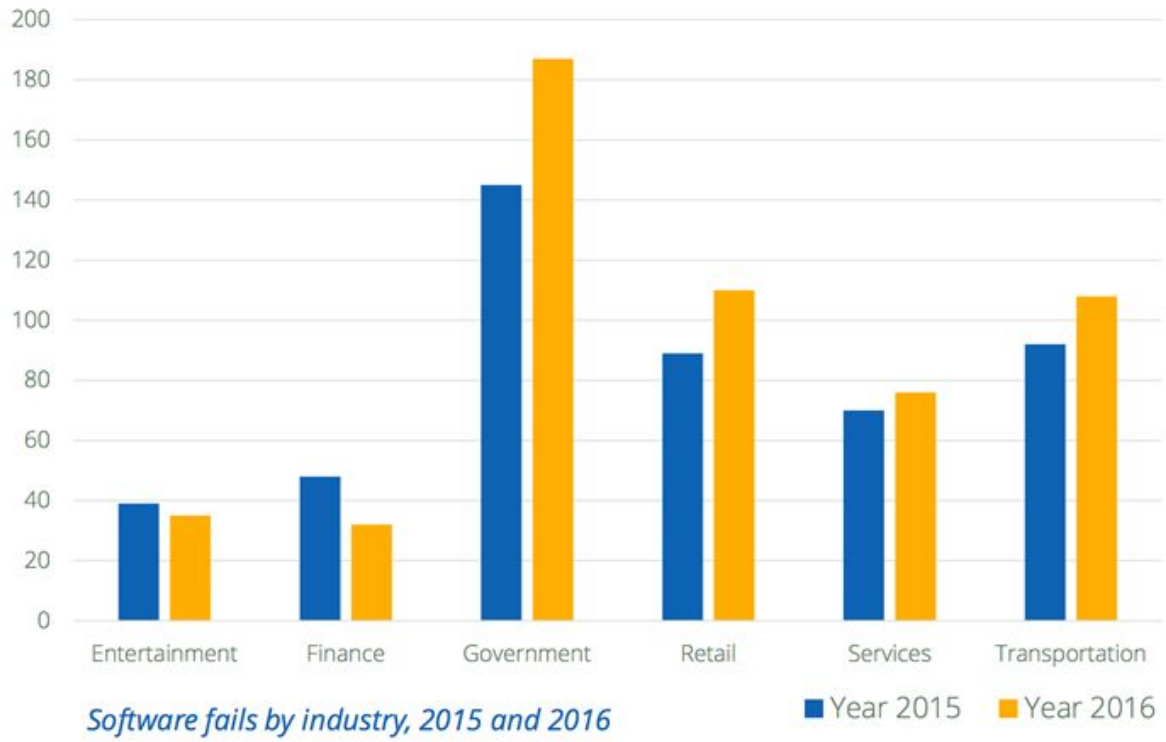
Software is really important

European Software Strategy

a NESSI Position paper 2008

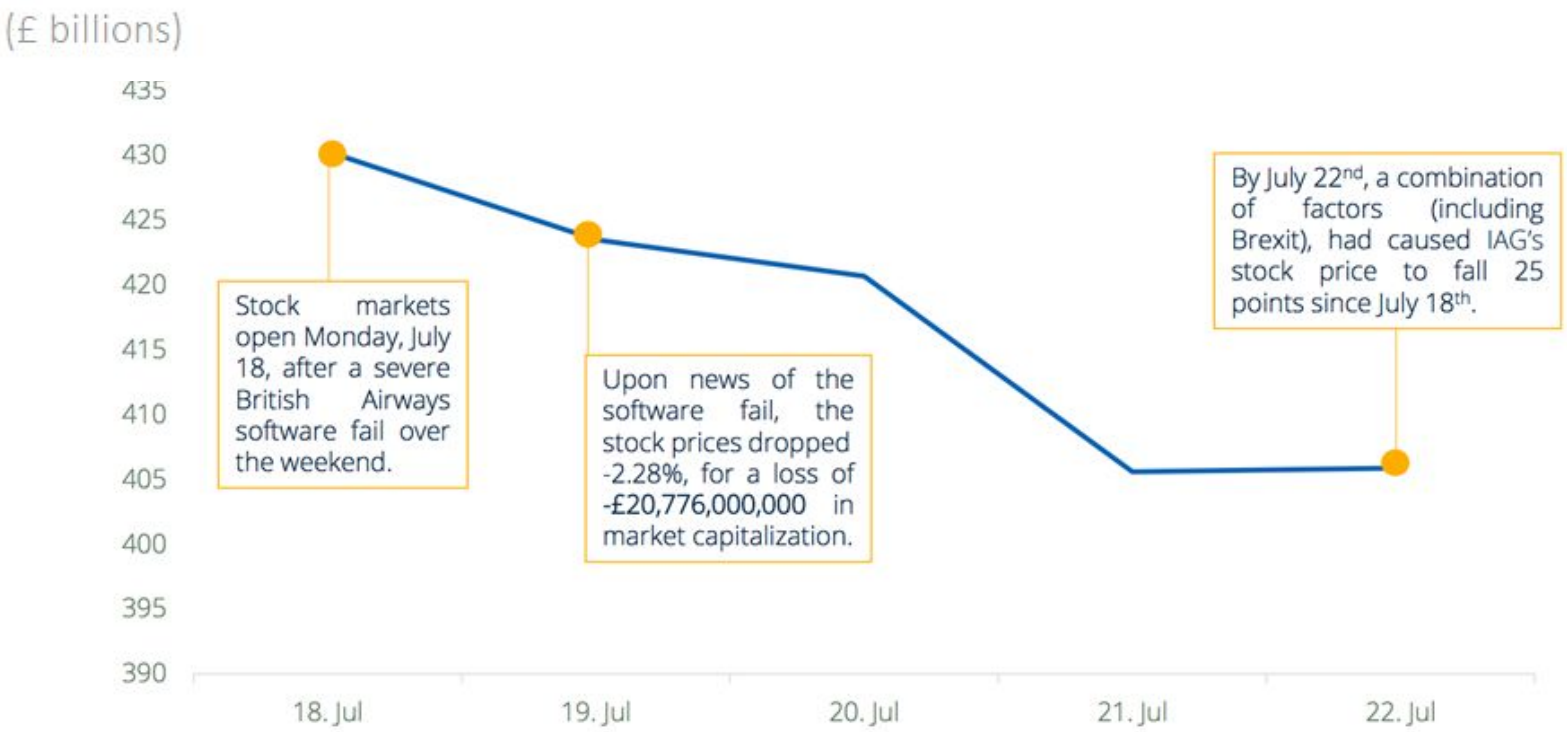
*Software is **no longer just an IT issue, it is a key asset**, opportunity of growth factor for all actors of the economy. Software drives the rise of global production systems across many sectors, software itself is emerging as the first global production system*

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

… and sometimes it is way too expensive to fail



British Airways group loses billions over software fail. Source: http://www.softwaretestingnews.co.uk/biggest-software-fails-2016

Many projects fail or are significantly challenged

MODERN RESOLUTION FOR ALL PROJECTS

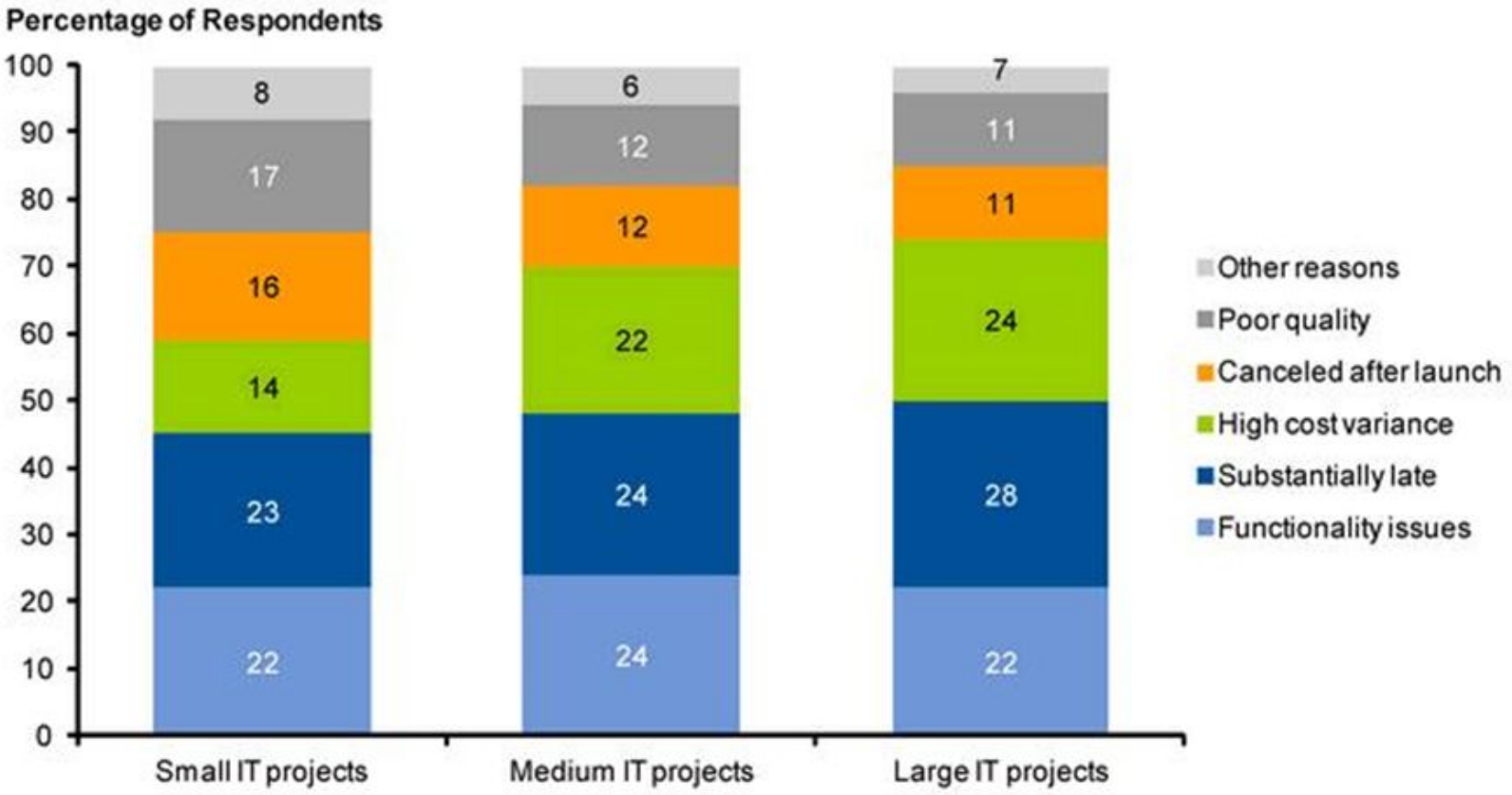
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

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 budged, 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

Source:
Standish Group, “The Chaos Manifesto”, 2015.

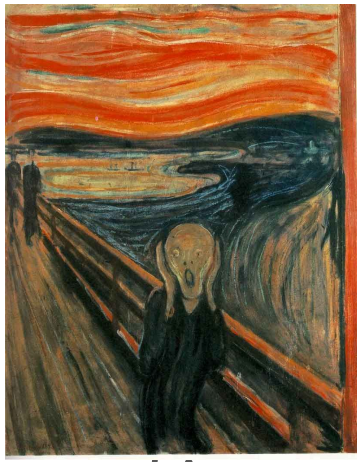
Why do software projects fail?



Source: Gartner survey on software failures, June 2012

The *"Software Crisis"*?

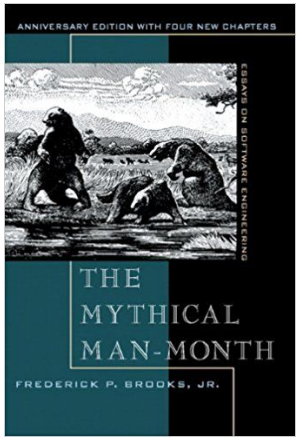
Term coined by Bauer in the first NATO 1968 Software Engineering Conference



"The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now that we have gigantic computers, programming has become an equally gigantic problem"

Edsger Dijkstra, *"The humble programmer"*, Communications of the ACM, 1972

Accidental *vs.* essential complexity



Accidental complexity

(Not essential to the problem to be solved)

```
public static final int UNITE = 1;
public static final int UNITED = 2;

// meowwwww meow
public int meow(int KITTENS_OF_THE_WORLD) {
    // MEOW
    if (KITTENS_OF_THE_WORLD < UNITED) {
        return KITTENS_OF_THE_WORLD;
    } else {
        // meeoowwwwwwwww
        // meooooowwwwwwwwwwwwwwwwwwww
        return meow(KITTENS_OF_THE_WORLD - UNITE)
            + meow(KITTENS_OF_THE_WORLD - UNITED);
    }
}
```

Essential complexity

(Inherent and unavoidable)

```
public int fibonacci(int x) {
    if (x == 1) {
        return 1;
    } else if (x == 2) {
        return 1;
    } else {
        return fibonacci(x - 1) + fibonacci(x - 2);
    }
}
```

There is no silver bullet to solve the crisis

"the very nature of software makes it unlikely that there will be any [silver bullets]" Brooks, 1987

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

Brooks, *"No Silver Bullet: Essence and Accidents of Software Engineering"*, IEEE Computer, 20(4), April 1987

Why do projects fail (revisited)

- Size of the projects (when compared to previous ones)
- Development cost explodes due to lack of productivity
- Lack of efficiency of the communication channels on big teams
- Key personnel has left
- Fail to understand requirements
- Projects delivered with lack of quality
- Introduction of badly understood recent technology
- Others...

These challenges require a Software Engineer!

Metaphor - Developing Software (by Grady Booch)

Professional Software Development

Software Engineering is intended to support professional software development rather than individual programming

Software is much more than just code

- Code
- Specification
- Configuration
- Manuals (system and/or user)
- Libraries
- Support websites
- Maintenance
- Not just for yourself (amateur) but for other people to use and engineers to change or maintain

Software Engineers

- Concerned with developing software-intensive trustworthy products that cope with increasing diversity, short time-to-market needs:
  - **Generic** - standalone developed by an organization and sold in the open market
  - **Customized** - commissioned and developed for a particular customer.

Software Engineering

Engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance in a cost-effective way.

Software Engineering Methods

Structured approach for developing **high quality** software that is **efficient** in a **cost effective** manner.

Computer Science *vs.* Software Engineering

- Computer Science focuses on theory and fundamentals
- Software Engineering is concerned with practicalities of developing and delivering useful software

They are **BOTH** extremely important!

Software engineers - Moral and Ethics

Your job involves wider responsibilities that go beyond the scope of technical skills.

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

Software engineers - Moral and Ethics

Deontology - regulates the relations and behavior of the professionals with:

- The Society
- The Employer
- The Clients
- The Colleagues

Software engineers IEEE/ACM Code of Ethics

"Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:"

Assignment: *Read*

https://www.ieee.org/about/corporate/governance/p7-8.html

https://www.ordemengenheiros.pt/fotos/editor2/regulamentos/codigo_ed.pdf

Public: *Software engineers shall act consistently with the public interest.*

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.*

Product: *Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.*

Judgement: *Software engineers shall maintain integrity and independence in their professional judgment.*

Management: *Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.*

Profession: *Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.*

Colleagues: *Software engineers shall be fair to and supportive of their colleagues."*

In Brasil - "Os Dez Mandamentos para Ética na Informática"

Instituto para Ética da Computação

"1- Você não deverá usar o computador para produzir danos em outra pessoa;

2- Você não deve interferir no trabalho de computação de outra pessoa;

3- Você não deve interferir nos arquivos de outra pessoa;

4- Você não deve usar o computador para roubar;

5- Você não deve usar o computador para dar falso testemunho;

6- Você não deverá usar software pirateado;

7- Você não deverá usar recursos de computadores de outras pessoas;

8- Você não deverá se apropriar do trabalho intelectual de outra pessoa;

9- Você deverá refletir sobre as conseqüências sociais do que escreve;

10- Você deverá usar o computador de maneira que mostre consideração e respeito ao interlocutor."

What are we going to learn /introduce ?

Software Modelling

Main Standard Engineering Processes

Notions of Project Management

Several other topics are left out for advanced courses

Stay tuned for scenes from our next episode

- are there Blueprints for Software like in other Engineering areas?
- UML what ?!?
- Use Case Diagrams ?

Who are the Three Amigos?