DOCENCIA VIRTUAL

Finalidad:

Prestación del servicio Público de educación superior (art. 1 LOU)

Responsable:

Universitat Politècnica de València.

Derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento conforme a políticas de privacidad:

http://www.upv.es/contenidos/DPD/

Propiedad intelectual:

Uso exclusivo en el entorno de aula virtual.

Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.

La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa o civil

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

INTRODUCTION TO SOFTWARE ENGINEERING

Software Engineering

Chapter 1

DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

etsinf

Goals

- Introduce Software Engineering and explain its importance for software development
- Answer main questions related to Software Engineering
- Introduce Software Process

Contents

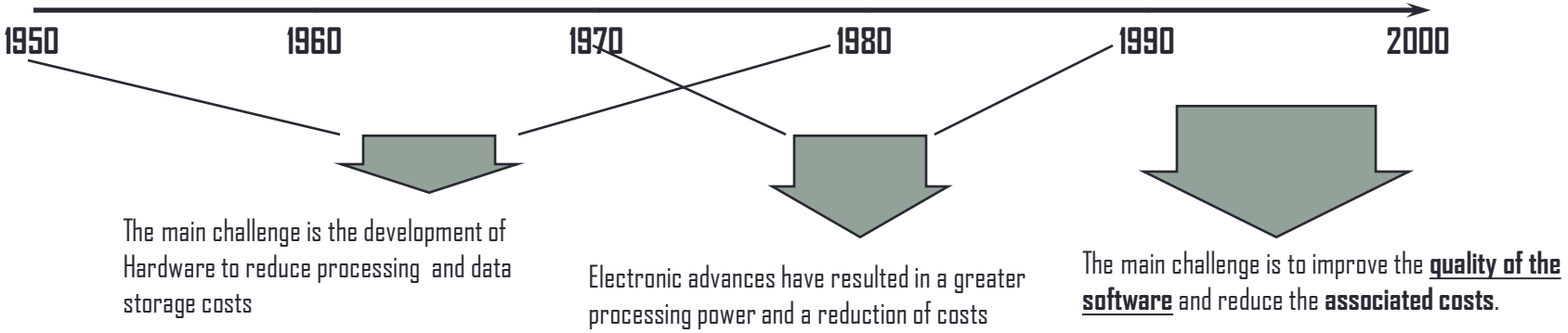
References

- Sommerville, I. Software Engineering. Addison-Wesley, 2008.
- Pressman, R., Software Engineering: A Practitioner's Approach. McGraw-Hill, 2005.
- Weitzenfeld, A., Object Oriented Software Engineering with UML, Java and Internet. Thomson, 2005

INTRODUCTION

Software makes a difference

In the latest decades software has overcome hardware as a critical factor for success

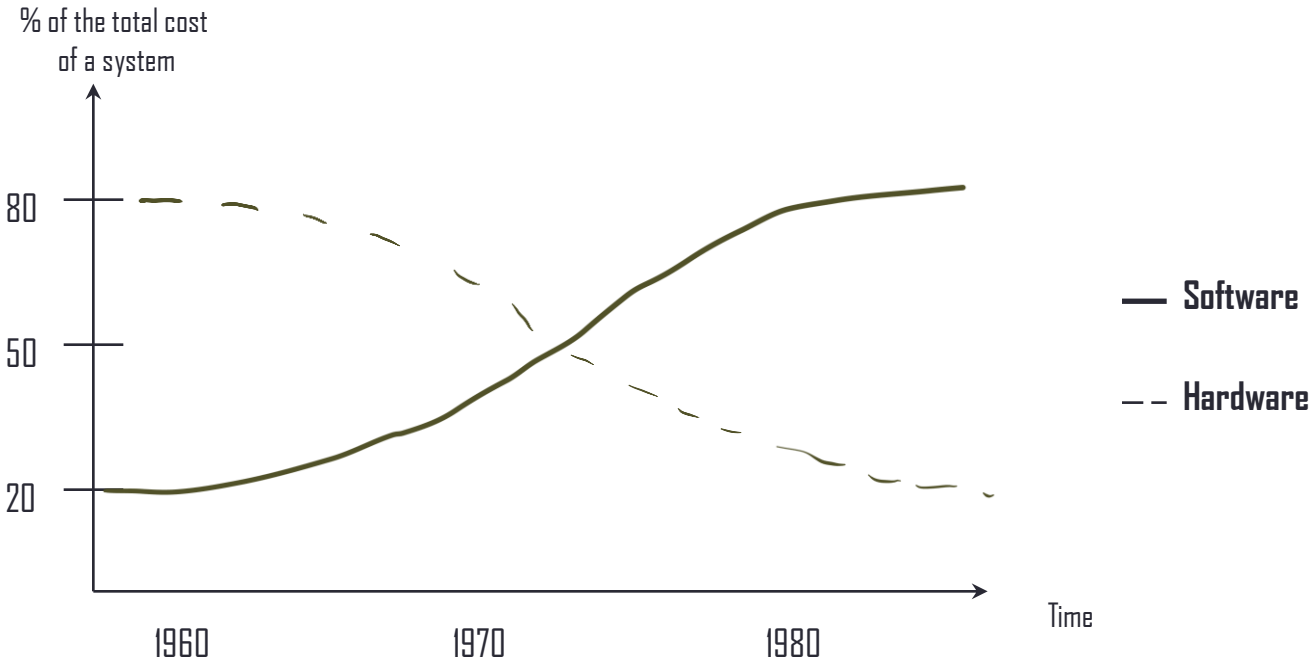


Software makes a difference

- In the last decade, as a result of the sucess of the Web as a platform and the use of mobile devices, the software industry has experienced a revolution
  - New languages
  - New HTML versions
  - New devices
  - New development methods!!

Software is more expensive…

- Evolution of the total cost of a system in terms of the percentage invested in software and in hardware



…and not just money!

- http://www5.in.tum.de/persons/huckle/bugse.html
- http://www.pcmag.com/article2/0,1759,1636333,00.asp
- http://www.microsiervos.com/archivo/ordenadores/10-peores-bugs.html
- http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all
- http://www.taringa.net/posts/info/3469982/Los-20-desastres-mas-famosos-de-la-historia-del-software.html

- http://catless.ncl.ac.uk/Risks

- ARIANE 5: https://youtu.be/PK_yguLapgA

THE SOFTWARE

- ✓ Characteristics
- ✓ Software crisis
- ✓ Quality Software
- ✓ Industry Problems

What is software?

- Instructions that provide an expected function and behavior when executed
- Data structures that allow programs to adequately manipulate information
- Documents that describe the operation and use of programs

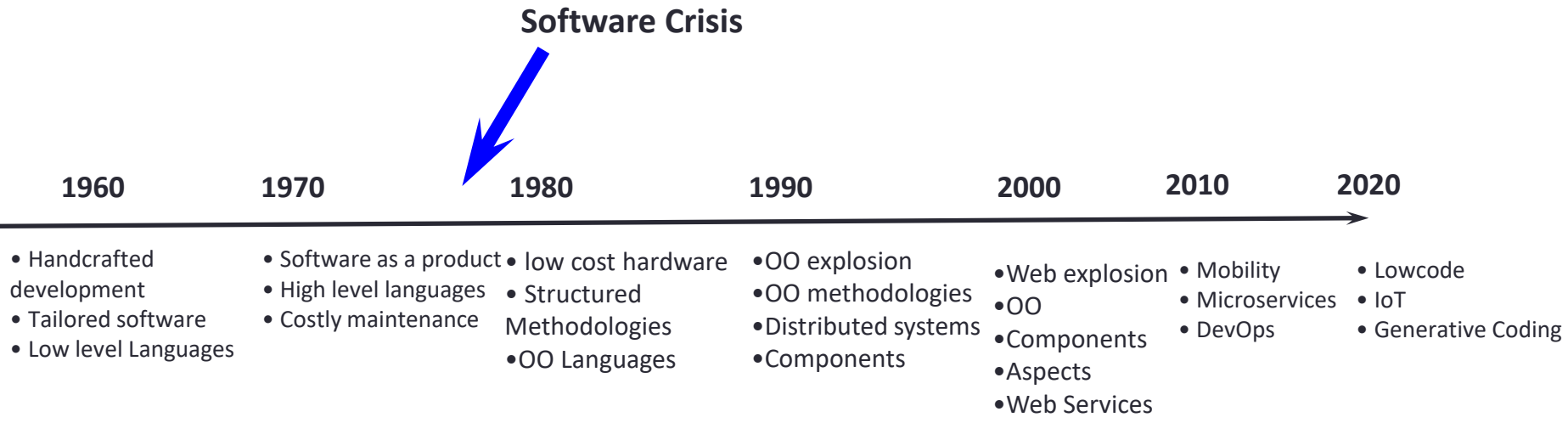
Characteristics of software

Software is a logical element:

- It is developed, not manufactured in the classical sense
- It doesn’t break down, it deteriorates as a result of changes
- Most of it is tailored for specific purposes instead of being assembled from existing components

Software Evolution

- The **context** in which software is developed is strongly related to the development of computing systems



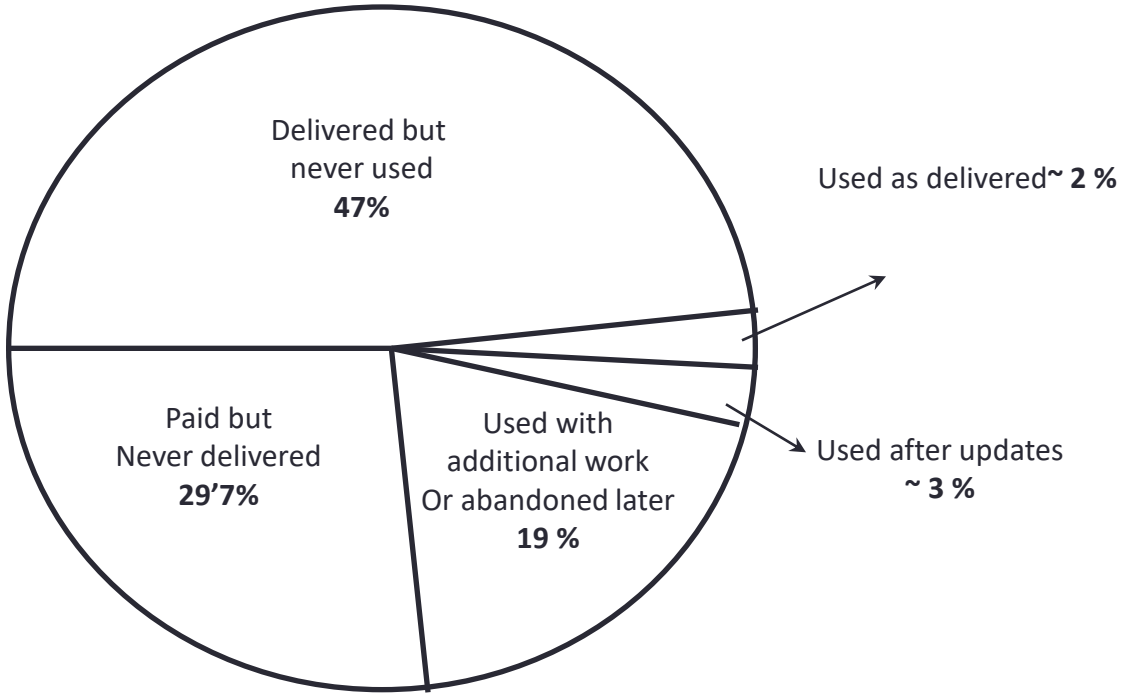
Software Crisis

- Costs are higher than planned
- Delivery dates delays
- Bad Performance
- Impossible maintenance
- High cost of Updates
- Unreliable products

Low quality Software!

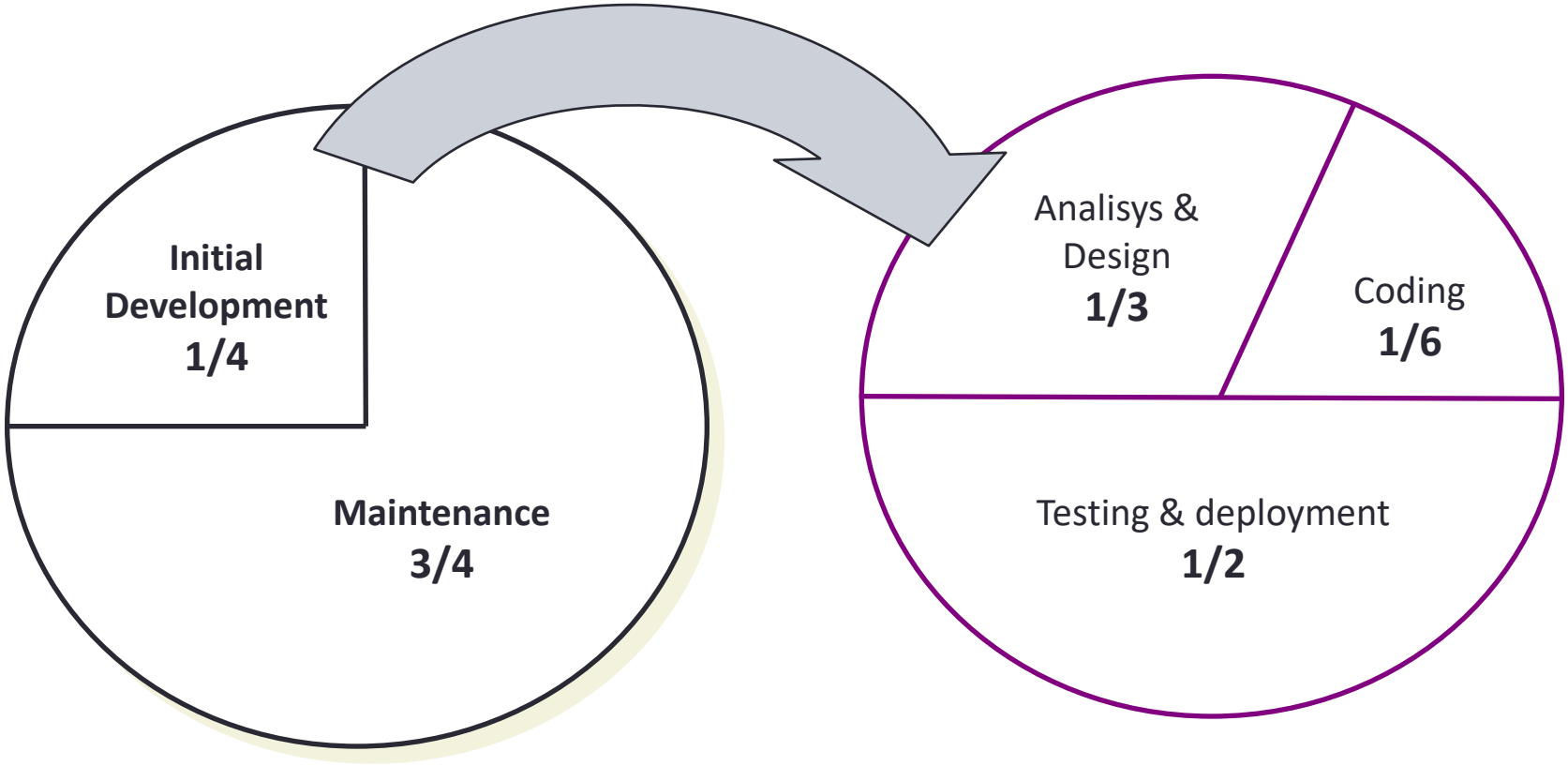
Software Crisis

- **Investment** in Software development
  - Year 1979 (Total: $ 6.8 million)



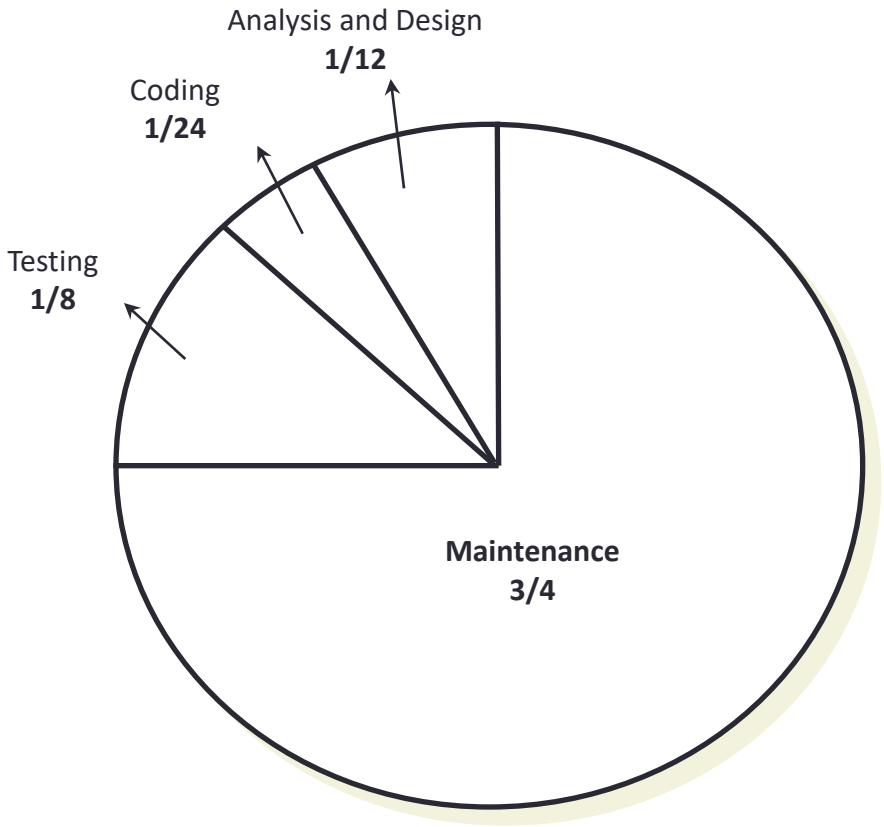
Software Crisis

- **Investment** in software development (by development phase):



Software Crisis

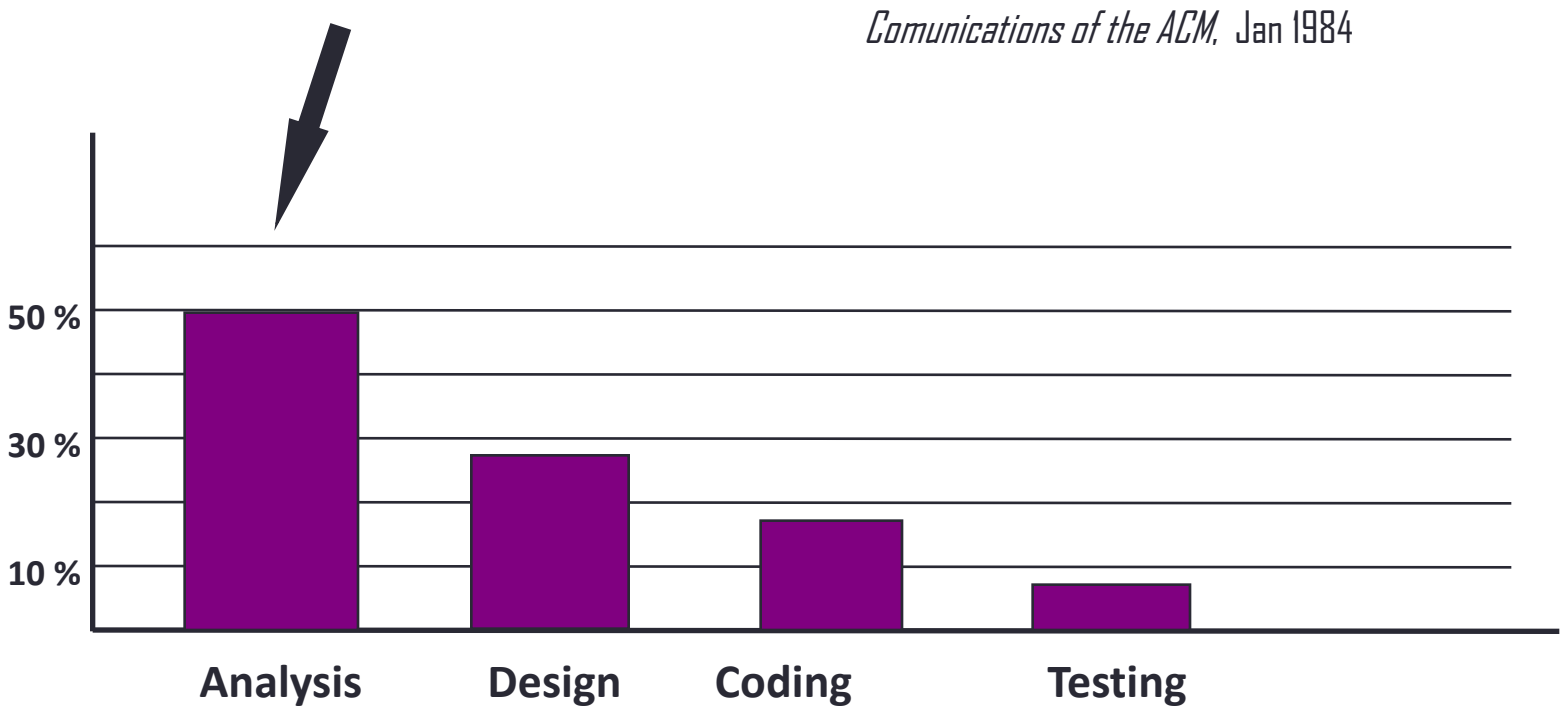
- Summary of **investment**:



- **Analysis & Design** ⟹ **8** %
- **Coding** ⟹ **4** %
- **Testing and Maintenance** ⟹ **88** %

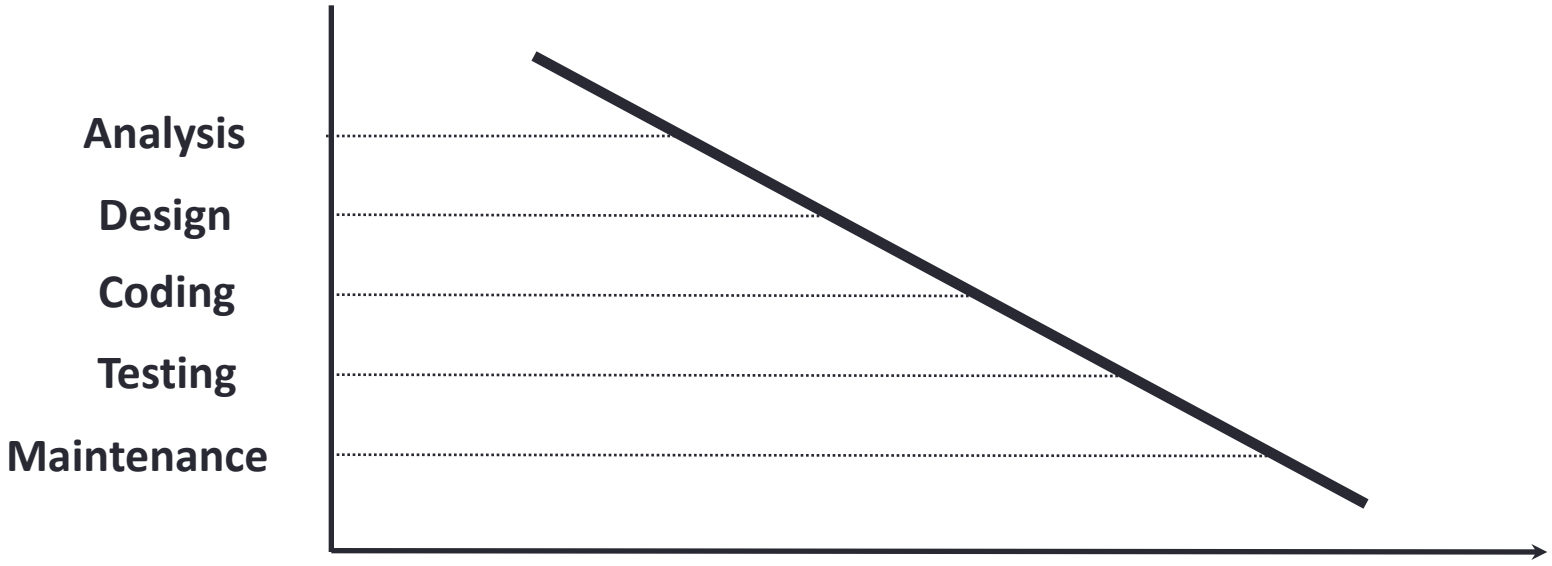
Software crisis

- **Errors** during software development (by development phase) :



Software Crisis

- **Cost** of fixing errors:



Software crisis

CHAOS Report 2020

Year	Success	With Difficul	Failed
1994	16%	53%	31%
2020	31%	50%	19%

Quality software

FUNCTIONAL requirements define what a system must do, describing its specific features and tasks, such as logging in or sending a message.

NON-FUNCTIONAL requirements, conversely, define how the system performs, focusing on its quality attributes like performance, security, reliability, and usability, such as a system loading in less than two seconds.

- The end goal is to produce high quality software

What is high quality software?

Agreement with:

- Functional and non functional requirements
- The documented development standards
- The expected features exhibited by any software developed profesionally

Necesary for the application to work

related to the

Quality factors

* IMPORTANT

- The classification of the quality factors of software takes into account three important aspects of a software product

  1. Its operational features
  2. Its capability to support updates
  3. Its adaptability to new environments

- These must be measured direct or indirectly during the whole development process

Quality factors

Q: Can you describe 3/4 quality factors?
Q: Which do you know?

Correctness Does it do what I want?

Reliability Is it reliable all the time?

Efficiency Will it run in the HW platform efficiently?

Integrity Is it safe?

Usability Is it designed to be used?

→ Operational features

Maintainability May it be corrected?

Flexibility May it be easily changed?

Testability May it be verified?

→ Supporting Updates

Reusability Is the software reusable?

Portability May it be used in another HW or OS?

Interoperability May it interact with another system?

→ Adaptability to New environments

Software Industry Problems

- Products are low quality
- High maintenance and development costs
- Delivery Delays

Reasons:

- ✓*Little investment and effort in the analysis and specification phases*
- ✓*Use of informal and inadequate models*
- ✓*Non physical nature of programming*
- ✓*Poor theorerical foundations*
- ✓*Products already in the market make it difficult innovation*
- ✓*High levels of hand crafting*
- ✓*Groupwork*
- ✓*Communication with users*
- ✓*Project management by non computing engineers*

Solutions

- Education:
  - Formal methods (executable formal languages: logic + algebra)
  - New development methods and new lifecycles
- Difusion of technological advancements
  - New programming paradigms
  - Architectures, protocols, computation models
- Tools investment
  - Modern development environments
  - Documentation generation engines

SOFTWARE ENGINEERING

- ✓ Definitions
- ✓ The Software Development Process

Won't ask the def. ⟶ No need to memorize, BETTER to understand

Definitions

- B. Boehm:
  - *"SE is the practical* application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate, and maintain them
- R. Pressman:
  - *"SE is a discipline that integrates methods, tools and procedures for the development of Software"*.
- A. Davis:
  - *"SE is the application of scientific principles for: (1) the transformation of a problem into a SW solution and (2) its maintenance during all its life".*
- I. Sommerville:
  - *"SE is an engineering approach covering all aspects of software production"*

Summarizing…

- SE is more than just coding
- The SE process starts well before writing lines of code and it continues after the first version of the product has been completed
- Key tasks are planning and rigurous control of software projects

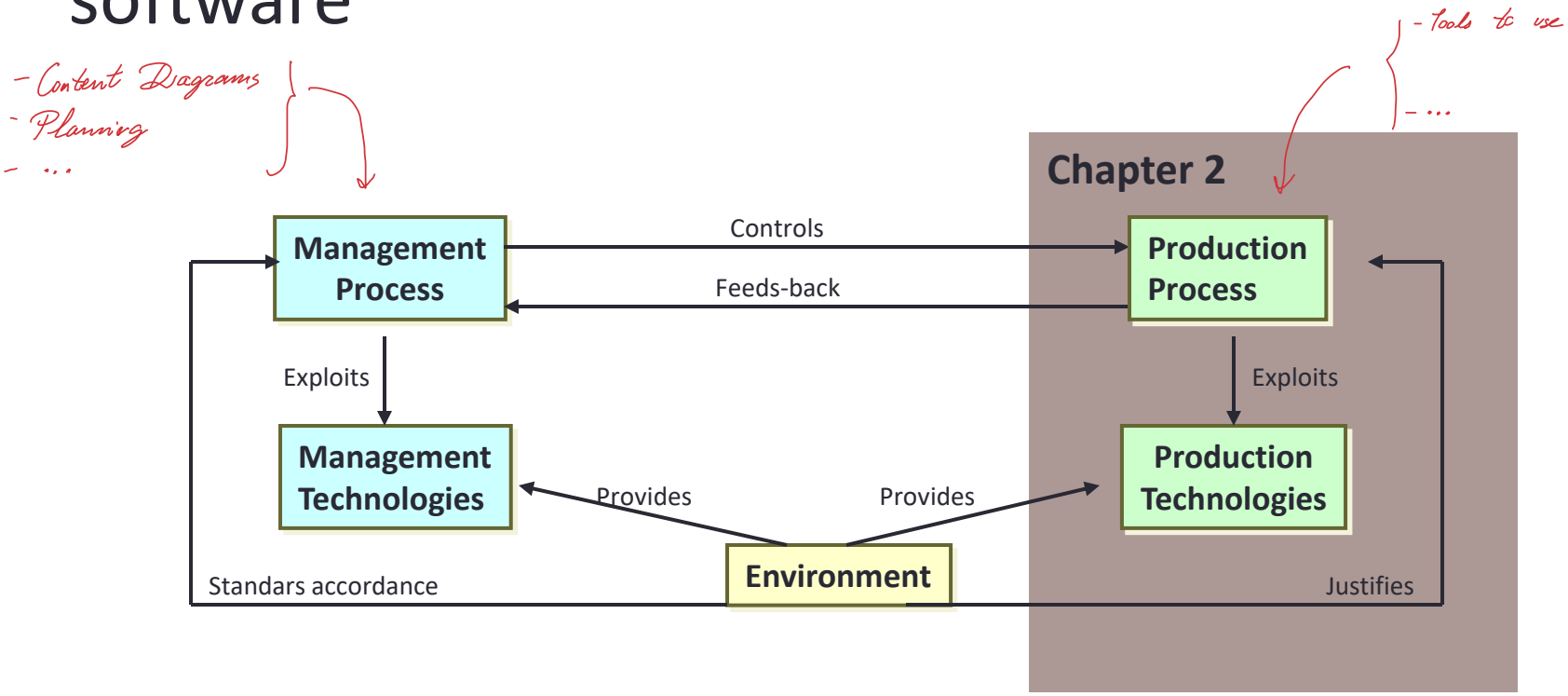
The importance of the process

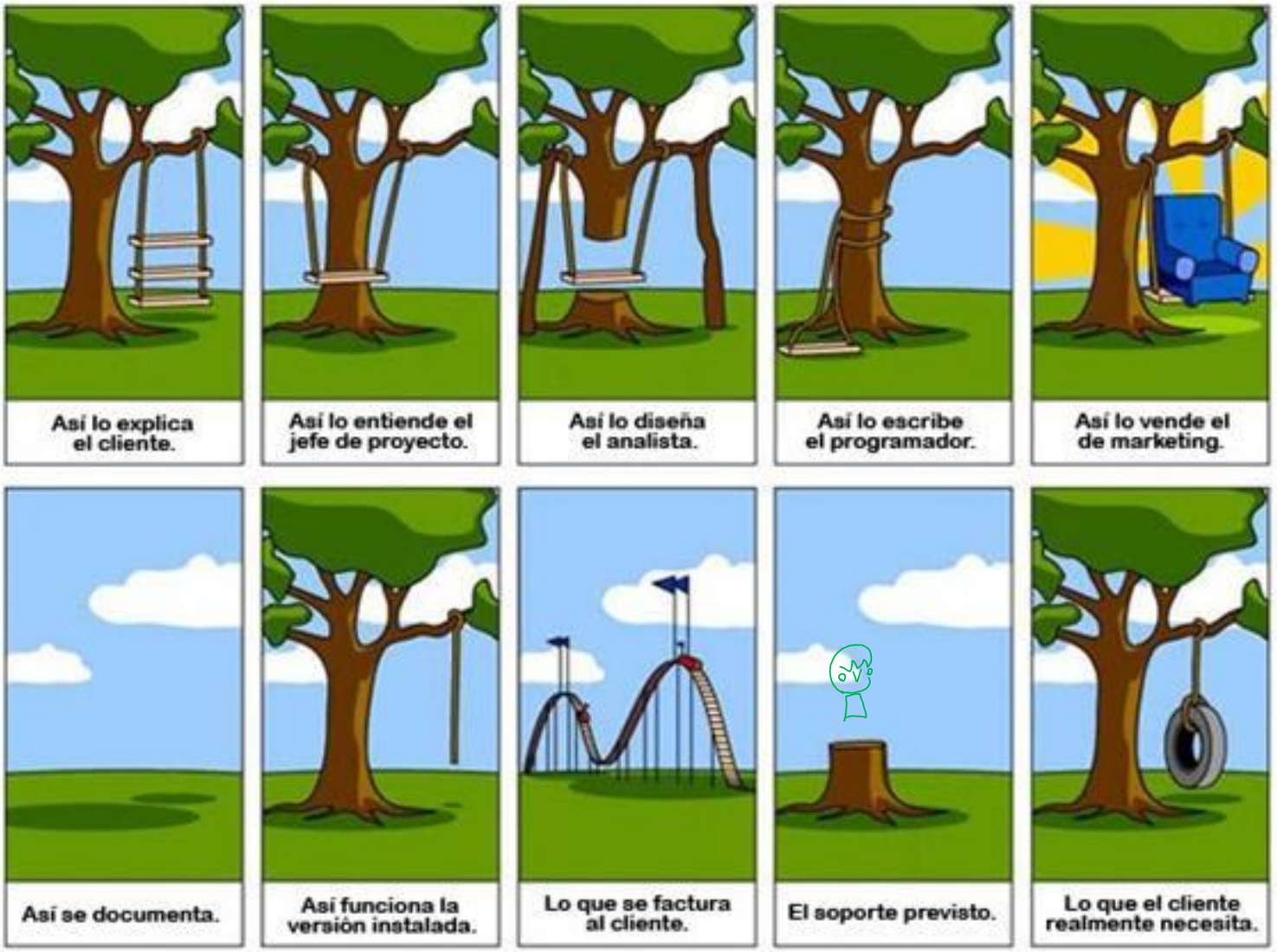
- The new challenges to be faced when developing software require fast and effective answers to changing requirements.

- The specification of a development process and the use of tools for its execution and monitoring are mandatory

The Software process

- It is a framework for the development of software





http://histinf.blogs.upv.es/files/2011/01/vi%C3%B1eta.jpg