INTRODUCTION TO SOFTWARE ENGINEERING

Software Engineering

Chapter 1

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

1. Introduction
2. Software
    - Characteristics
    - Software Crisis
    - Quality Software
    - Industry Problems
3. Software Engineering
    - Definitions
    - Software Process
    - Management of software development projects

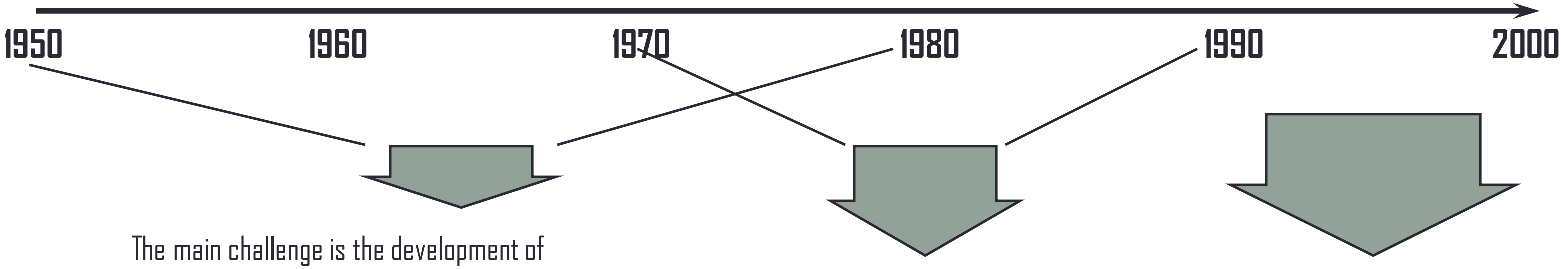
References

- 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



The main challenge is the development of Hardware to reduce processing and data storage costs

Electronic advances have resulted in a greater processing power and a reduction of costs

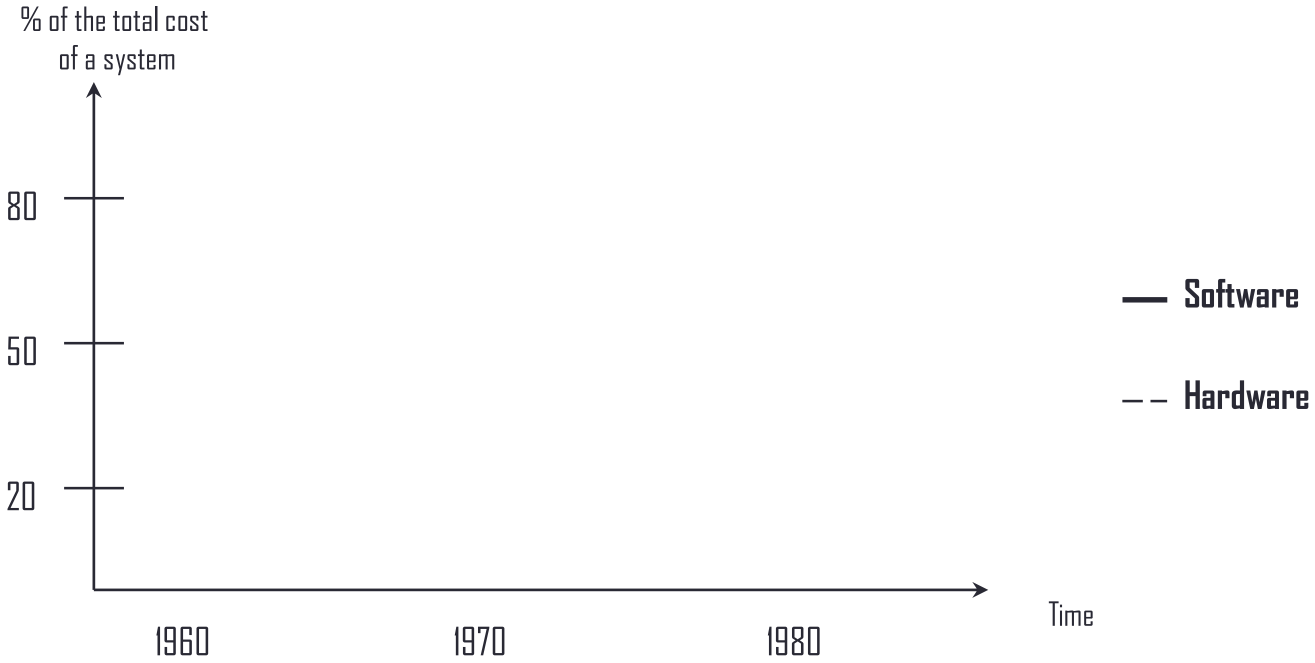
The main challenge is to improve the **quality of the software** and reduce the **associated costs**.

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

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

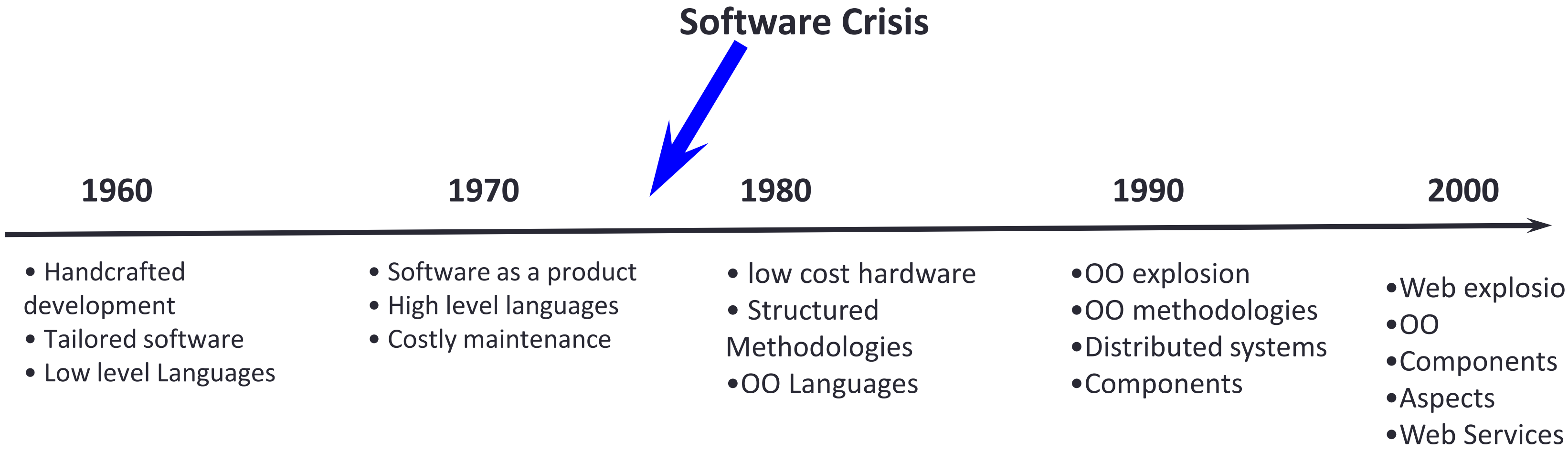
Characteristics of software

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

1960

- Handcrafted development
- Tailored software
- Low level Languages

1970

- Software as a product
- High level languages
- Costly maintenance

1980

- low cost hardware
- Structured Methodologies
- OO Languages

1990

- OO explosion
- OO methodologies
- Distributed systems
- Components

2000

- Web explosion
- OO
- Components
- Aspects
- Web Services

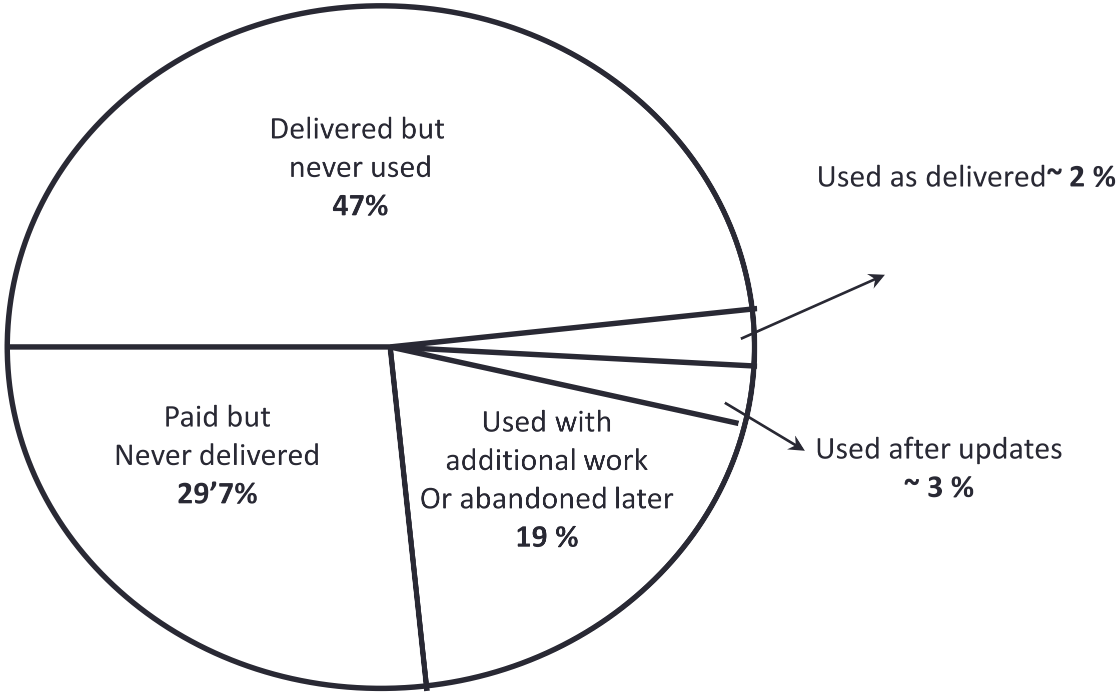
Software Crisis

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

Low quality Software!

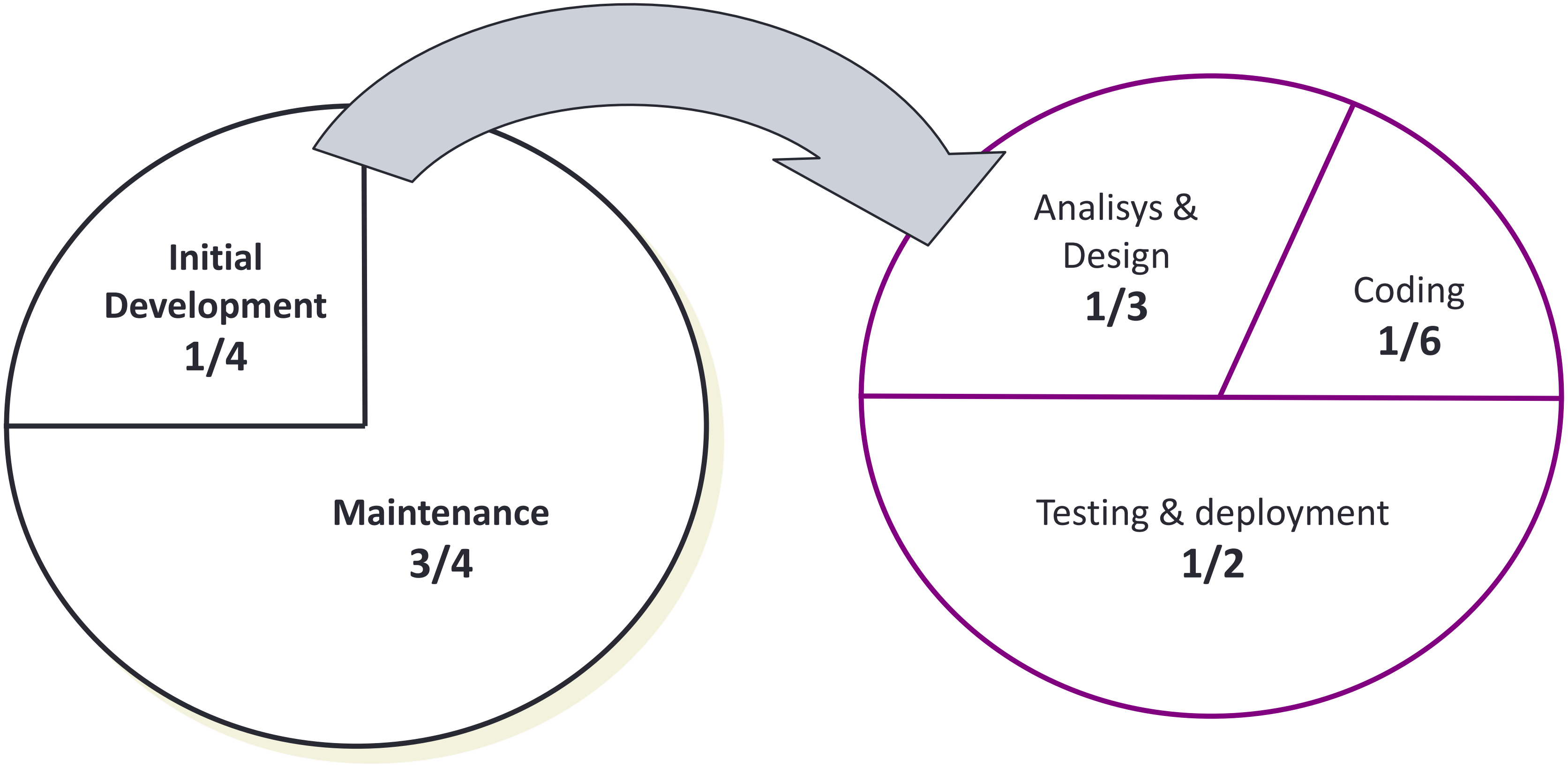
Software Crisis

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



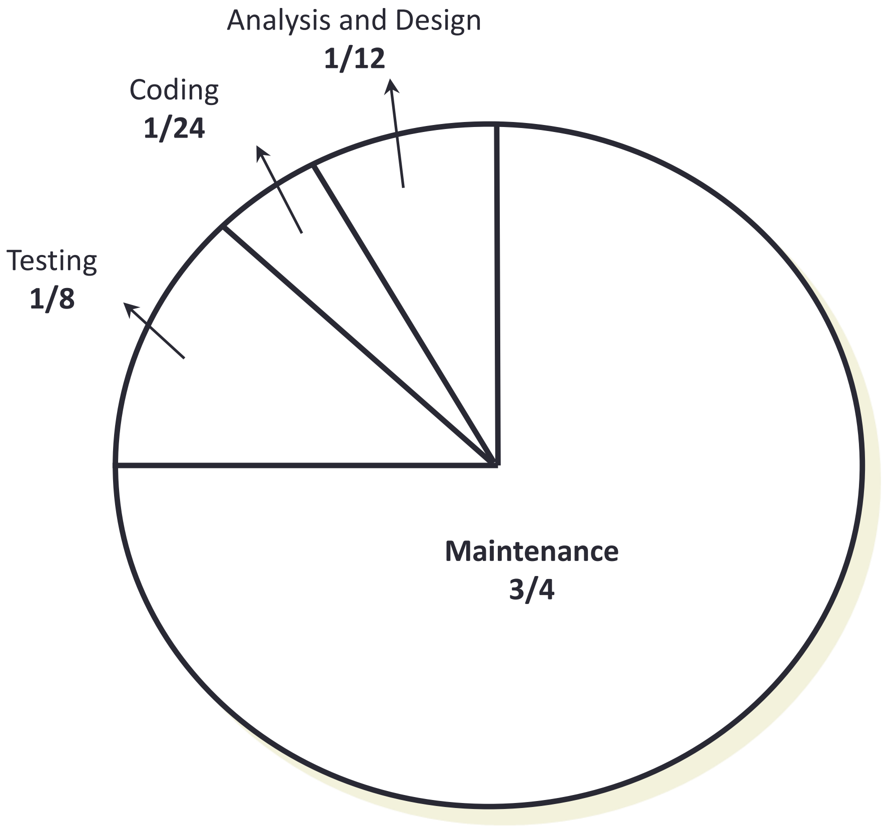
Software Crisis

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



Software Crisis

- Summary of **investment**:

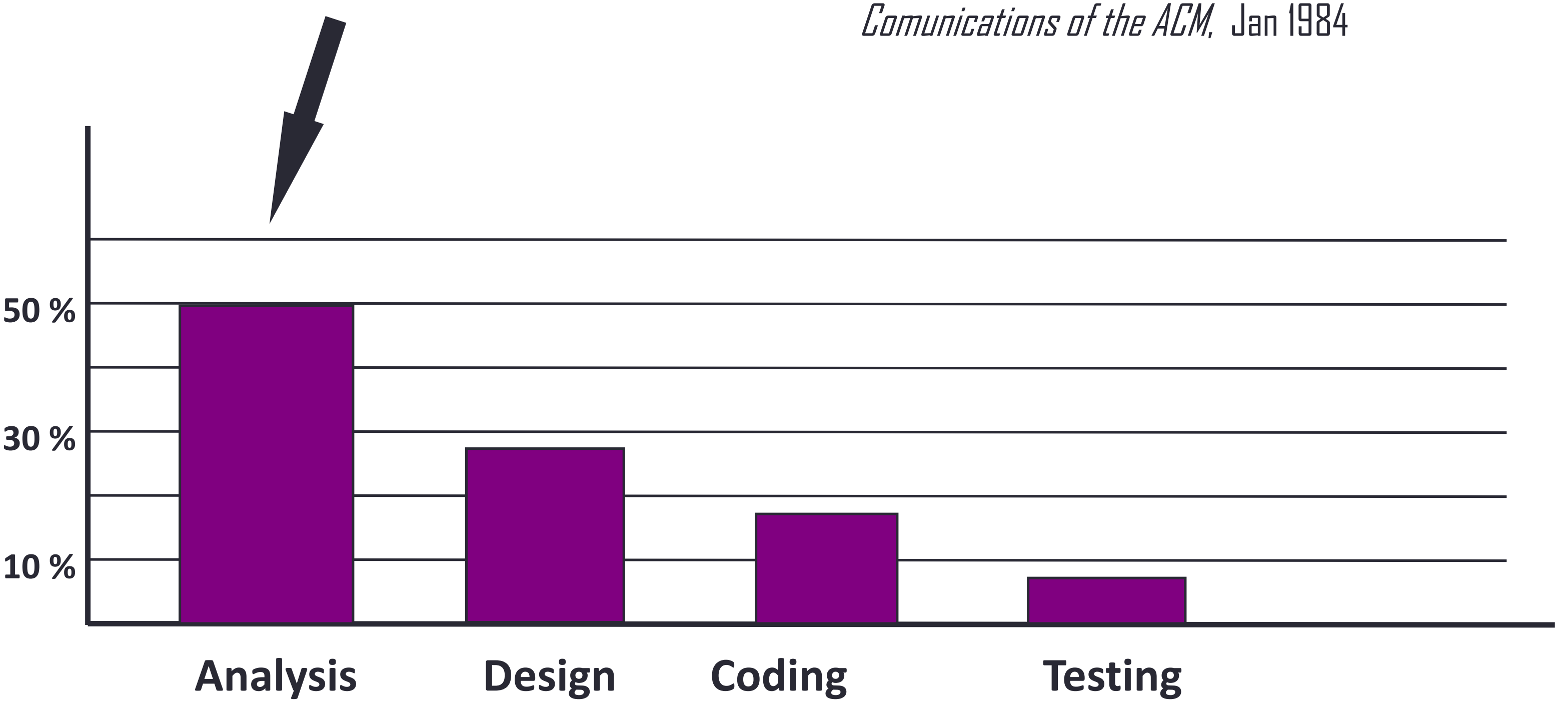


- **Analysis & Design** ⟹ **8** %
- **Coding** ⟹ **4** %
- **Testing and Maintenance** ⟹ **88** %

Software crisis

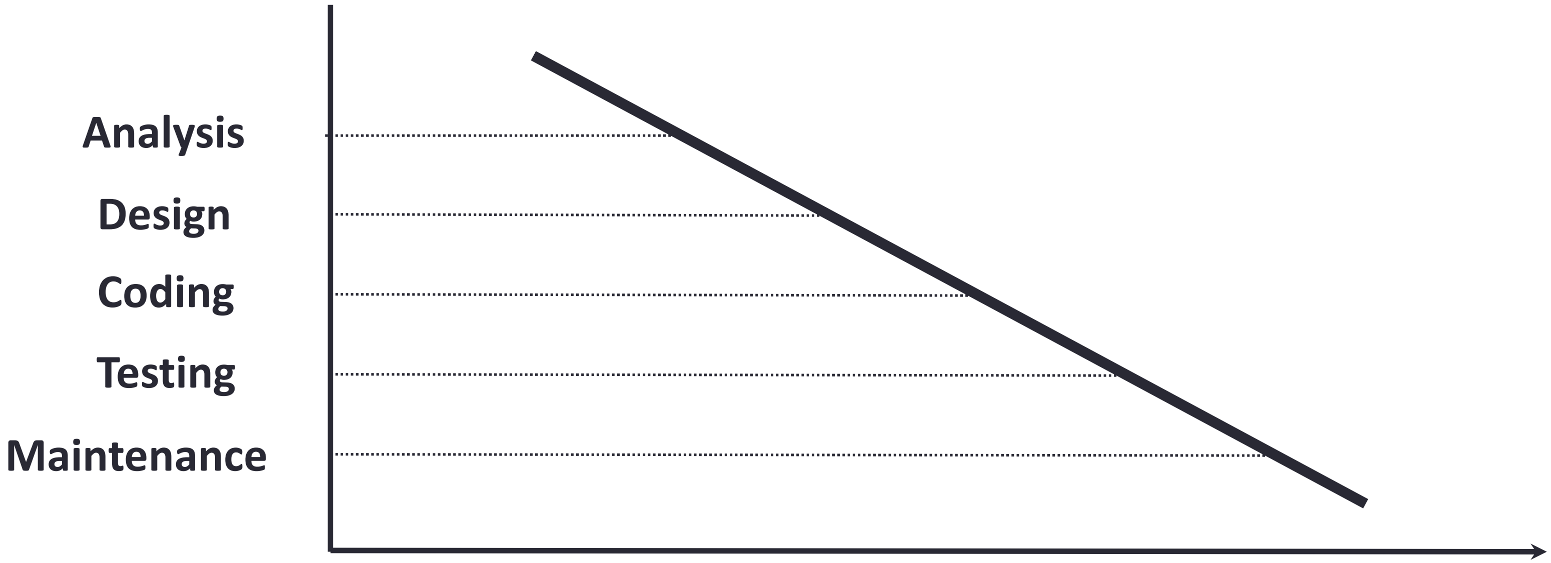
- **Errors** during software development (by development phase) :

Comunications of the ACM, Jan 1984



Software Crisis

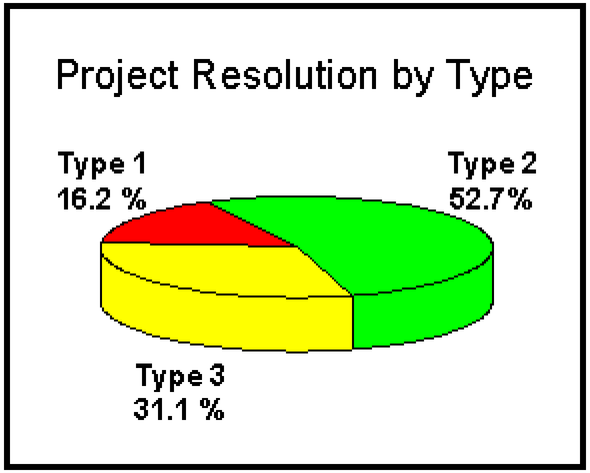
- **Cost** of fixing errors:



Software crisis

Investment in software development.

- Year 1994 (Total: $ 250 Kmillions/year -- 175.000 projects)



- ***Type 1 (Success).*** The project is Ended on time and within butget with all initially planned features and functionalities.

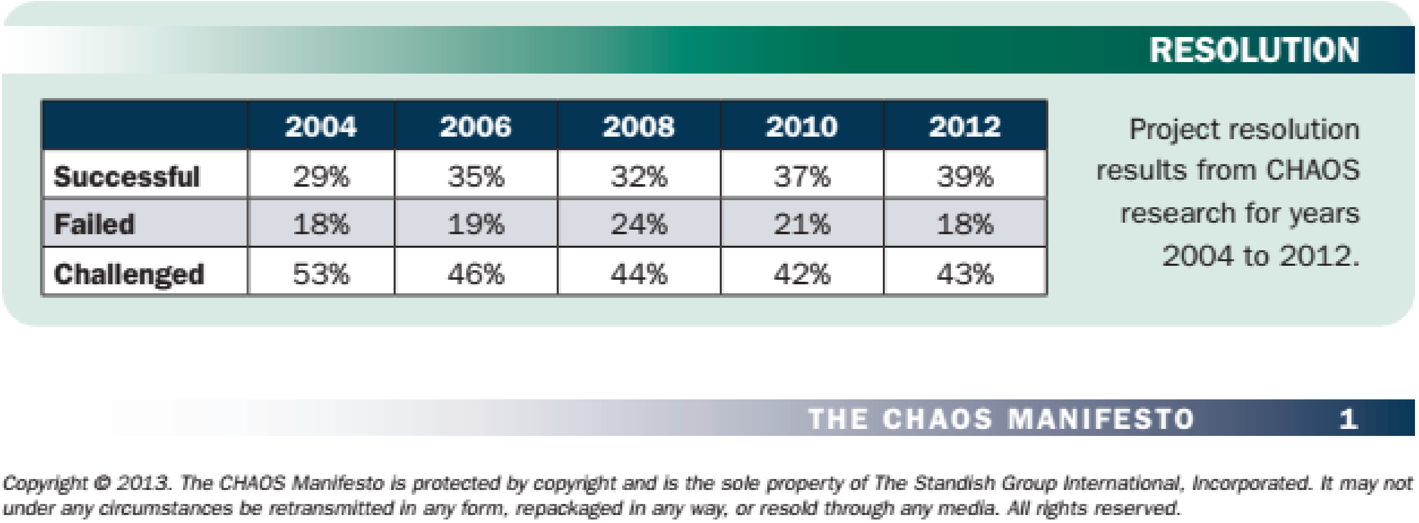
- ***Type 2 (Updates).*** The project is Is ended late and at a higher cost It has less features and functionalities than were specified.

- ***Type 3 (Cancelled).*** The project is cancelled during its development.

http://www.standishgroup.com/sample_research/chaos_1994_1.php

Software crisis

CHAOS Report...



RESOLUTION

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

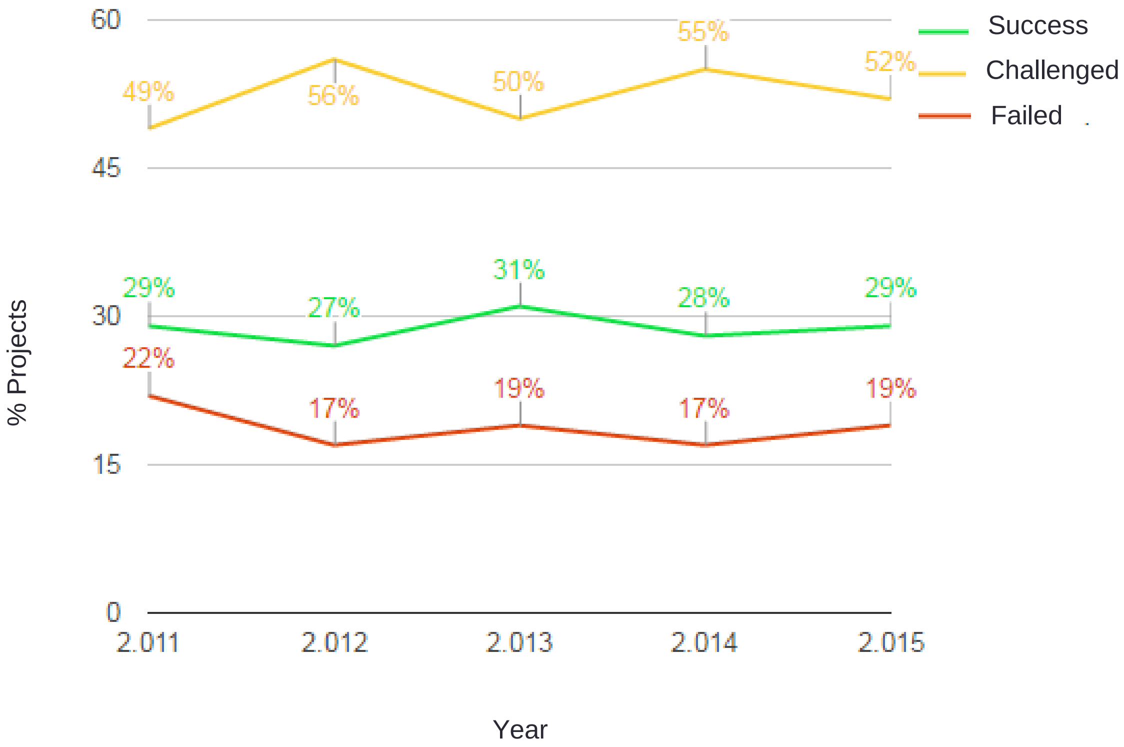
Project resolution results from CHAOS research for years 2004 to 2012.

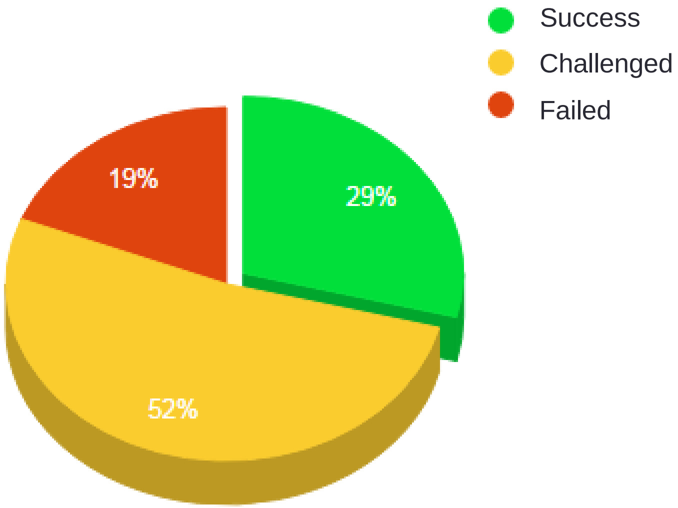
THE CHAOS MANIFESTO 1

http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf

Software crisis

CHAOS Report 2015



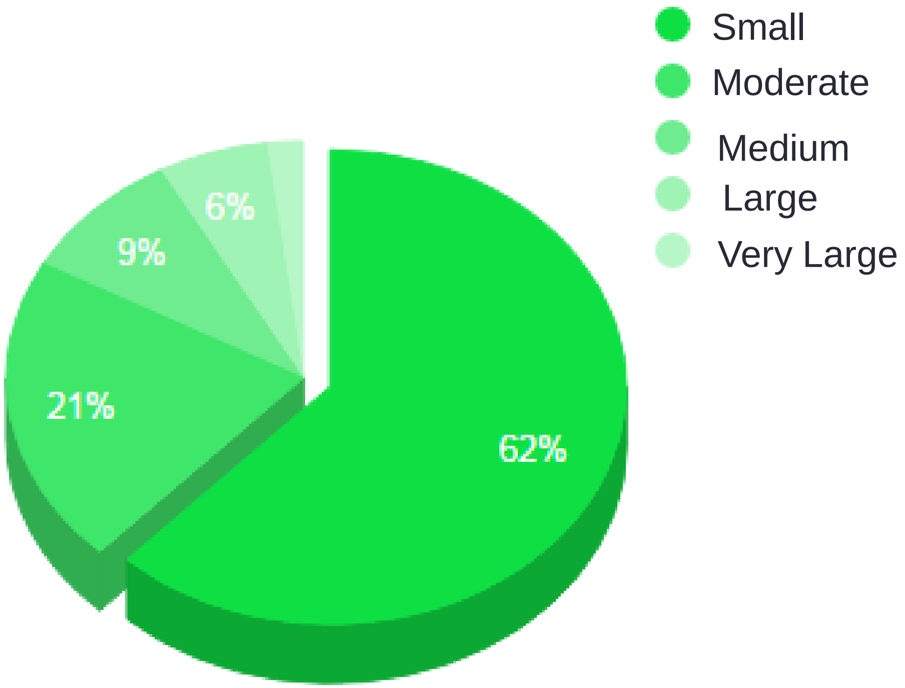


Software crisis

CHAOS Report 2015

Is the size of a software project relevant?

% over successful projects
2011-2015



Quality software

- 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

Quality factors

- The classification of the quality factors of software takes into account three importact 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

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*
- ✓*Comumunication 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 Developmet Process

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”*

A little bit of history

- http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/Web/History/

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

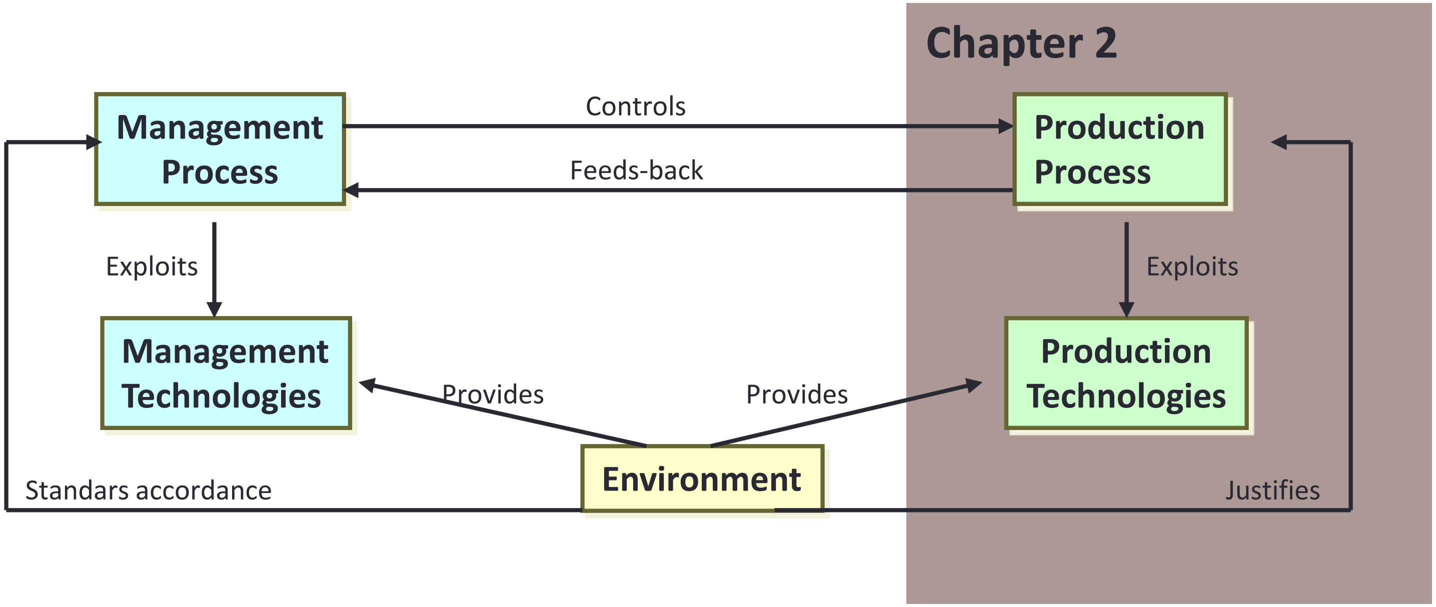
The importance of the process

- 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

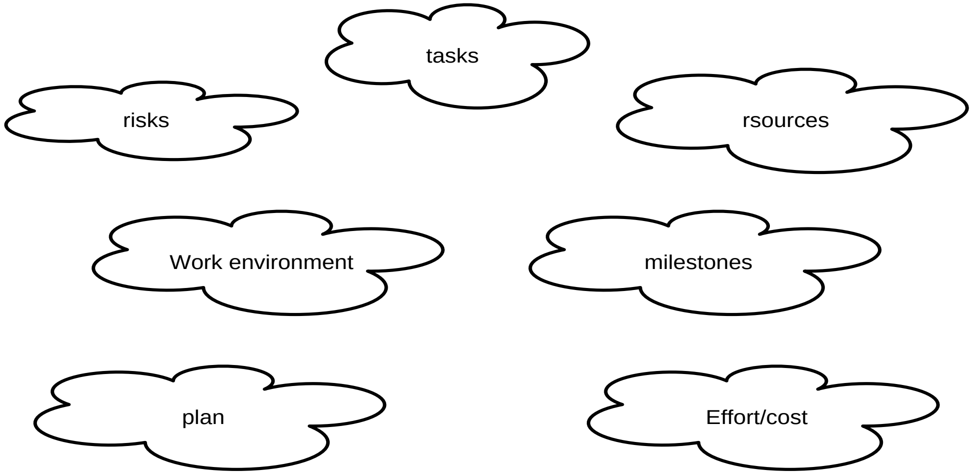
The Software process

- It is a framework for the development of software



Software projects management

- The management of a software project is the first level of a software development process and it covers all the development process



tasks

risks

rsources

Work environment

milestones

plan

Effort/cost

Software engineering vs Engineering

- Similarities
  - Activities to be done are not specific of software projects
  - Many common management techniques
  - Many similar problems (time, resources, changing specifications…)

- Differences
  - The product (software) is not tangible and flexible
  - The software process is not standard. Several alternatives exist
  - Many times software projects are “unique”

Software projects management

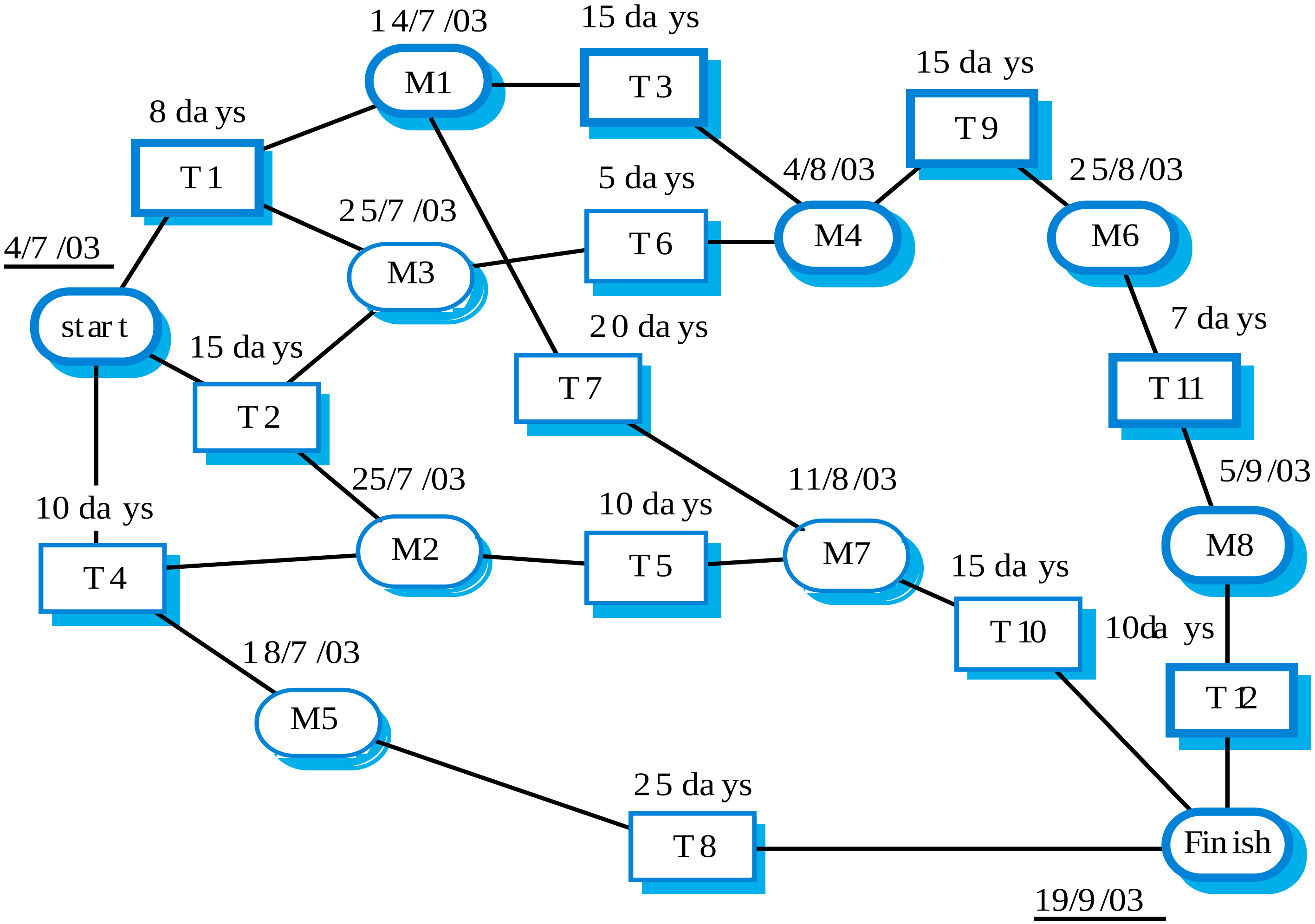
- Activities involved:

  - Writing proposal
  - Project planning
  - Cost estimation
  - Selection and evaluation of human resources
  - Project control
  - Writing and presenting reports

Project Plan - Structure

1. **Introduction** Goals and restrictions (budget, time…)
2. **Project organization** Team Organization (personnel-roles)
3. **Risks analysis** Risks, probabilities and estrategies
4. **Software and Hardware requirements** Acquisitions, prices, delivery dates …
5. **Division of labour** Activities, milestones and products to deliver
6. **Planning** Dependencies between activities
   Estimated time, assigning resources
7. **Supervision and reporting**

Tasks networks (PERT diagrams)



14/7/03

M1

15 days

T3

15 days

T9

8 days

T1

4/8/03

25/8/03

5 days

T6

M4

M6

25/7/03

M3

4/7/03

start

15 days

T2

20 days

T7

7 days

T11

10 days

25/7/03

11/8/03

5/9/03

T4

M2

10 days

T5

M7

M8

15 days

T10

10days

18/7/03

T12

M5

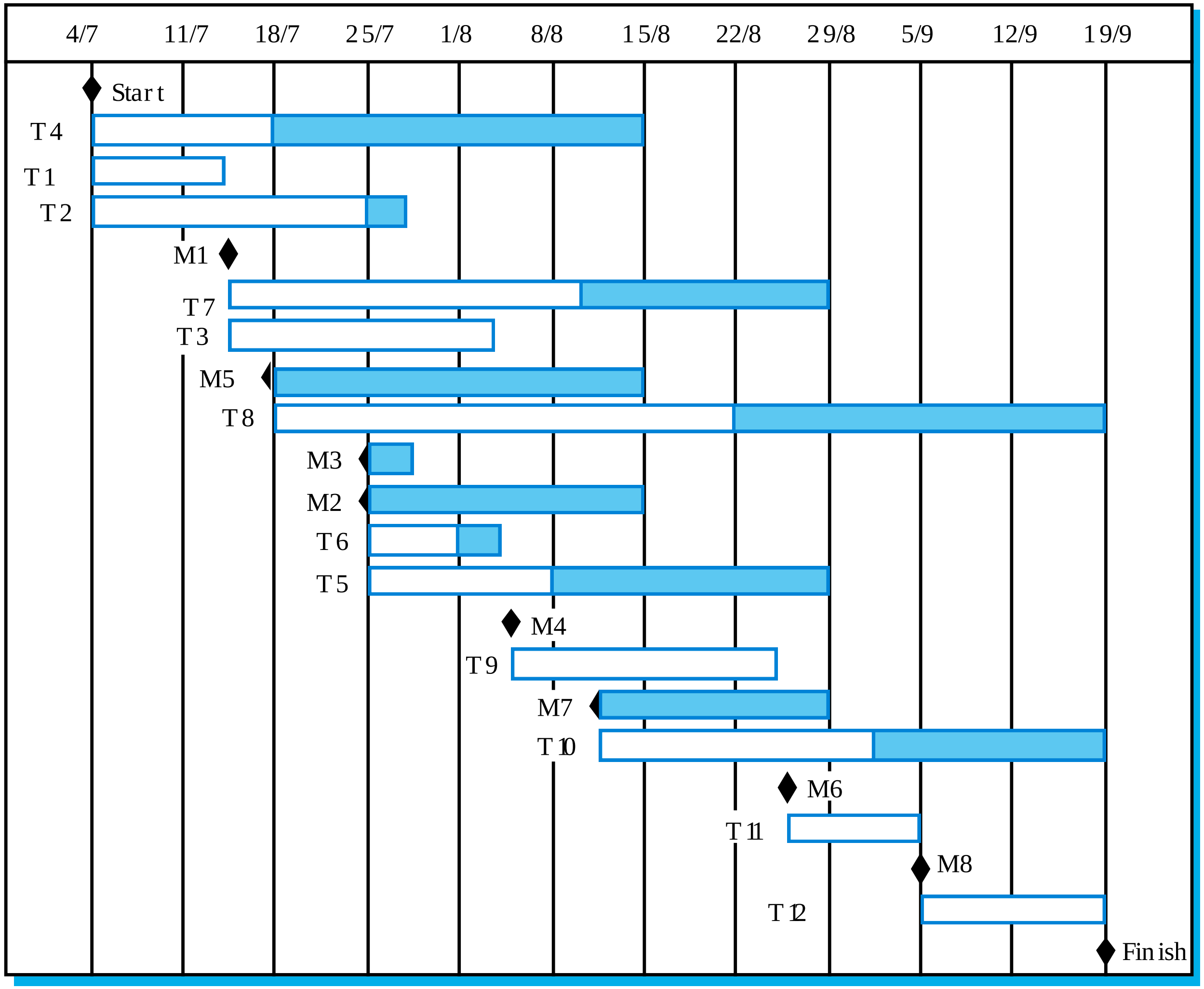
25 days

T8

Finish

19/9/03

Gantt diagram


4/7
11/7
18/7
25/7
1/8
8/8
15/8
22/8
29/8
5/9
12/9
19/9
Start
T4
T1
T2
M1
T7
T3
M5
T8
M3
M2
T6
T5
M4
T9
M7
T10
M6
T11
M8
T12
Finish