



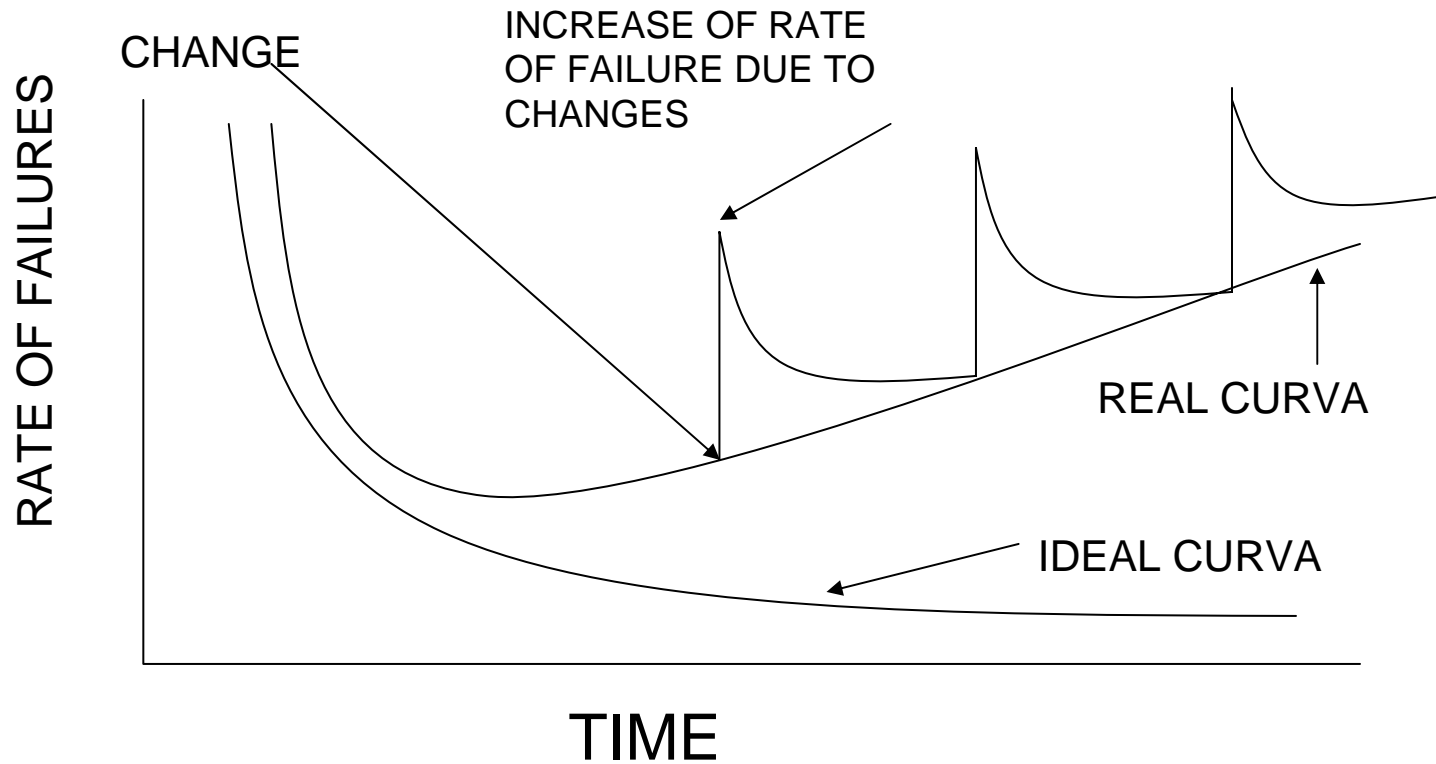
Departamento de Informática
Universidad de Valladolid
Campus de Segovia

TOPIC 6: INTRODUCTION TO SOFTWARE ENGINEERING

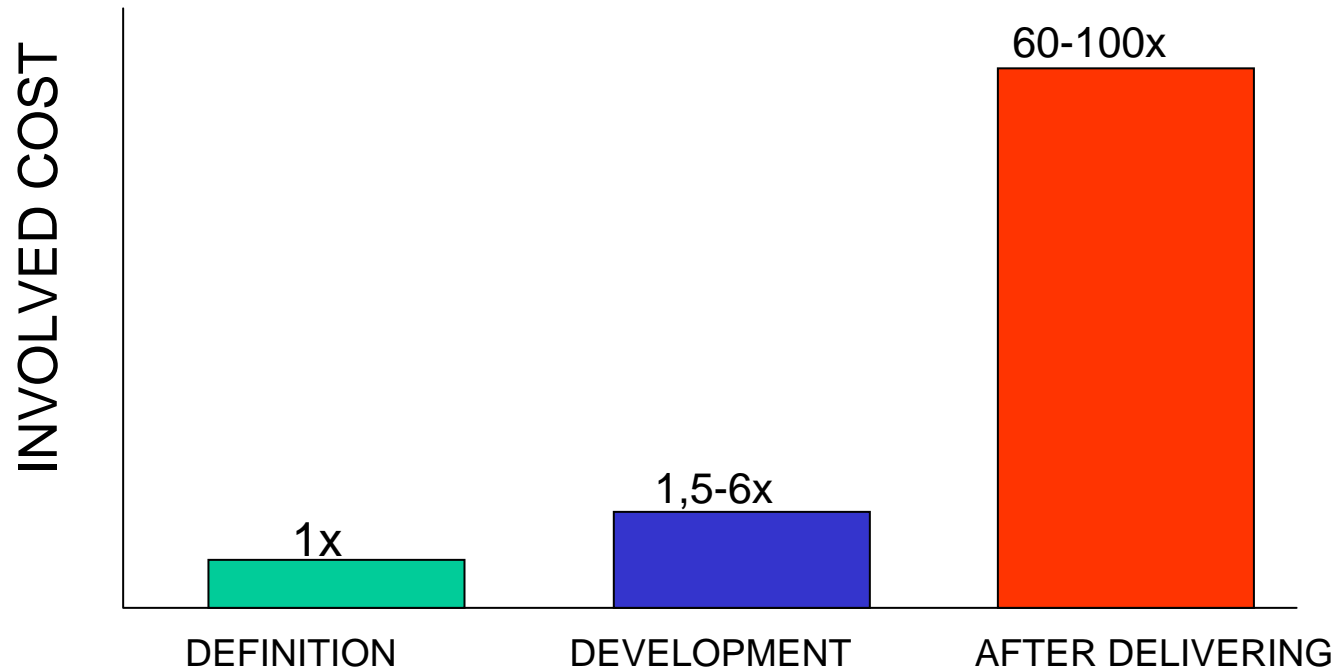
THE SOFTWARE FEATURES

- Software is developed not manufacture.
- Software does not break when it is used but spoiled due to changes.
- Nowadays, software is developed without reuse.

REAL AND IDEAL SOFTWARE FAILURE CURVE



CHANGE IMPACT ACCORDING TO THE PHASE



**“ The sooner you write code the
later you will finish it.”**

SOFTWARE ENGINEERING. DEFINITION

- It is the most suitable framework to guarantee the **quality of software**.
- Consists in a process of development and some models and tools.

THE PROCESS

- The software development is an iterative learning process
- In order to describe the different steps that define a process, S.I. provides:
 - **Models:** describe how to develop software (a process for the creation of software)
 - **Tools:** give us an automatic or semiautomatic approach to the process and methods.

GENERAL DEVELOPMENT PHASES

- **Definition phase** (What do I have to do?).
(problem definition=user-analyst communication).
- **Development phase** (How do I have to do it?).
- **Maintenance phase** (Changes).
 - Correctness
 - Adaptation
 - Improveness
 - Prevention

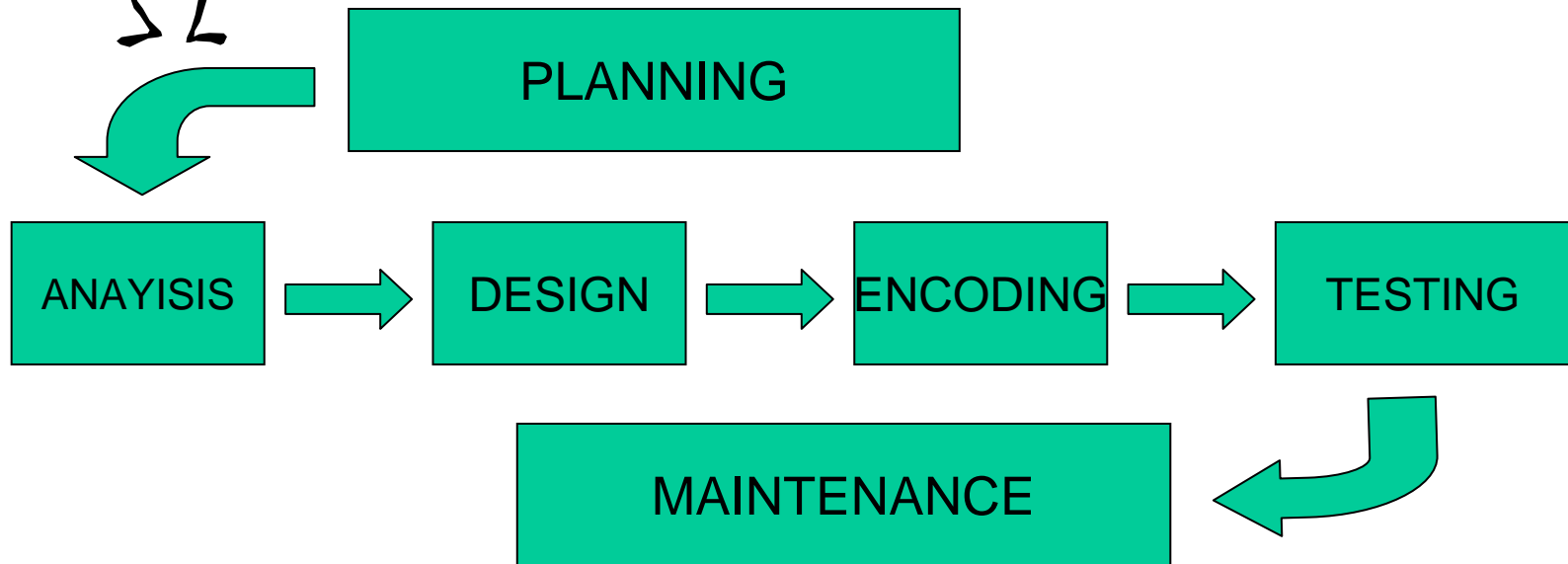
MODELS

- Waterfall model.
- Prototypes.
- Evolutive models. The spiral model.

WATERFALL MODEL

- The **waterfall model** is a sequential software development model.
- In this model development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing, (validation), integration, and maintenance.

WATER FALL MODEL



THE WATERFALL MODEL . PLANNING

- User-analyst Communication.
- First approach to software requirements.
- Assessment of human and technical resources.

THE WATERFALL MODEL . ANALYSIS

- Analysis phase consists in:
 - Describing the application software functionality.
 - Integrating the modules that makes up the application.
 - Determining how will be the testing phase.
- As a result of this phase a complete requirement specification is achieved.

THE WATERFALL MODEL . DESIGN

- Design phase consists in:
 - Describing the software application architecture (relationships among the different parts that makes up the application).
 - These parts are delivered among the programmers.
- Each part before mentioned is more detailed described using an algorithmic language.

THE WATERFALL MODEL . IMPLEMENTATION

- Encoding the algorithms by means of an specific programming language

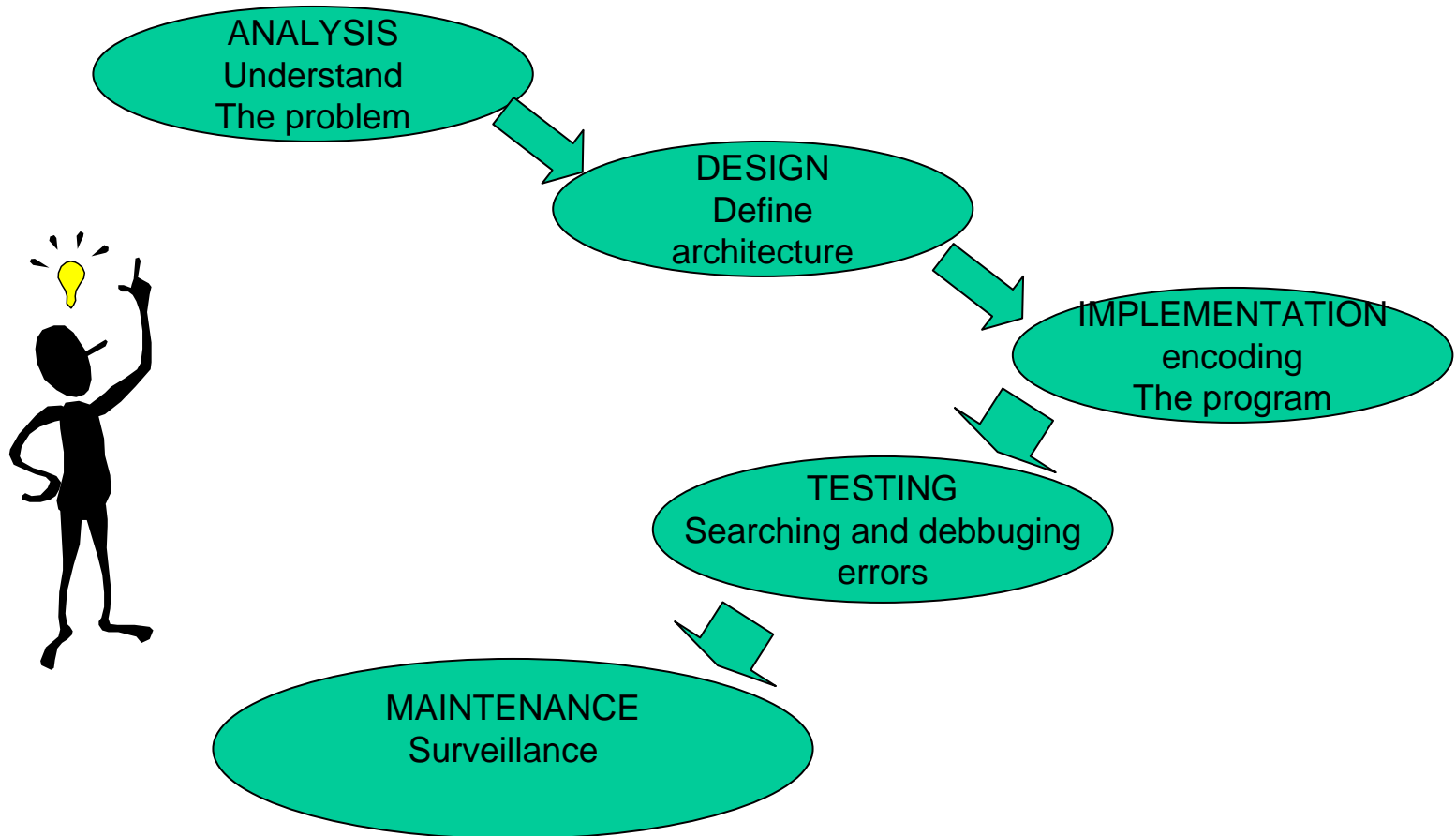
THE WATERFALL MODEL . TESTING

- The testing phase consists in searching errors in order to debug them.
- In this regards:
 - each module and their connection among them are tested
 - Finally the application, as a whole, is tested.

THE WATERFALL MODEL . MAINTENANCE

- The final user and maintenance documentation is achieved.
- New detected errors are debugged.
- The application is surveillanced in order to consider future possible modifications

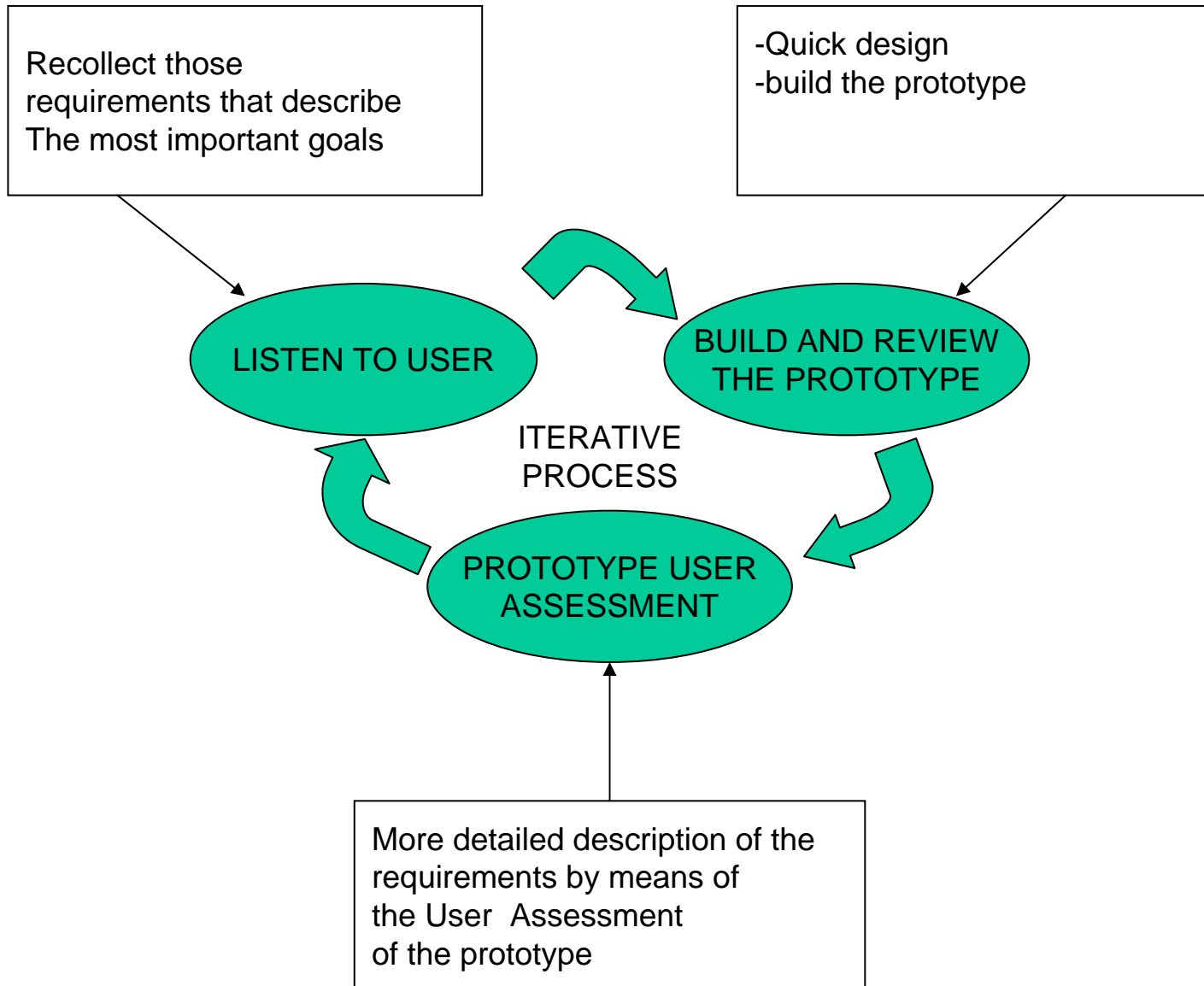
SUMMARIZING



INCONVENIENTS OF THIS MODEL

- Software Projects unfrequently present a sequential evolution.
- Furthermore:
 - usually all the requirements are not explicit described at the beginning as it is required for this model.
 - And the user has to be patient since the complete application only will be available at the final stages.

PROTOTYPES



INCONVENIENTS OF THIS MODEL

- User may think that the built prototype is a finished software application.
- Sometimes the chosen development tools may not be suitable.

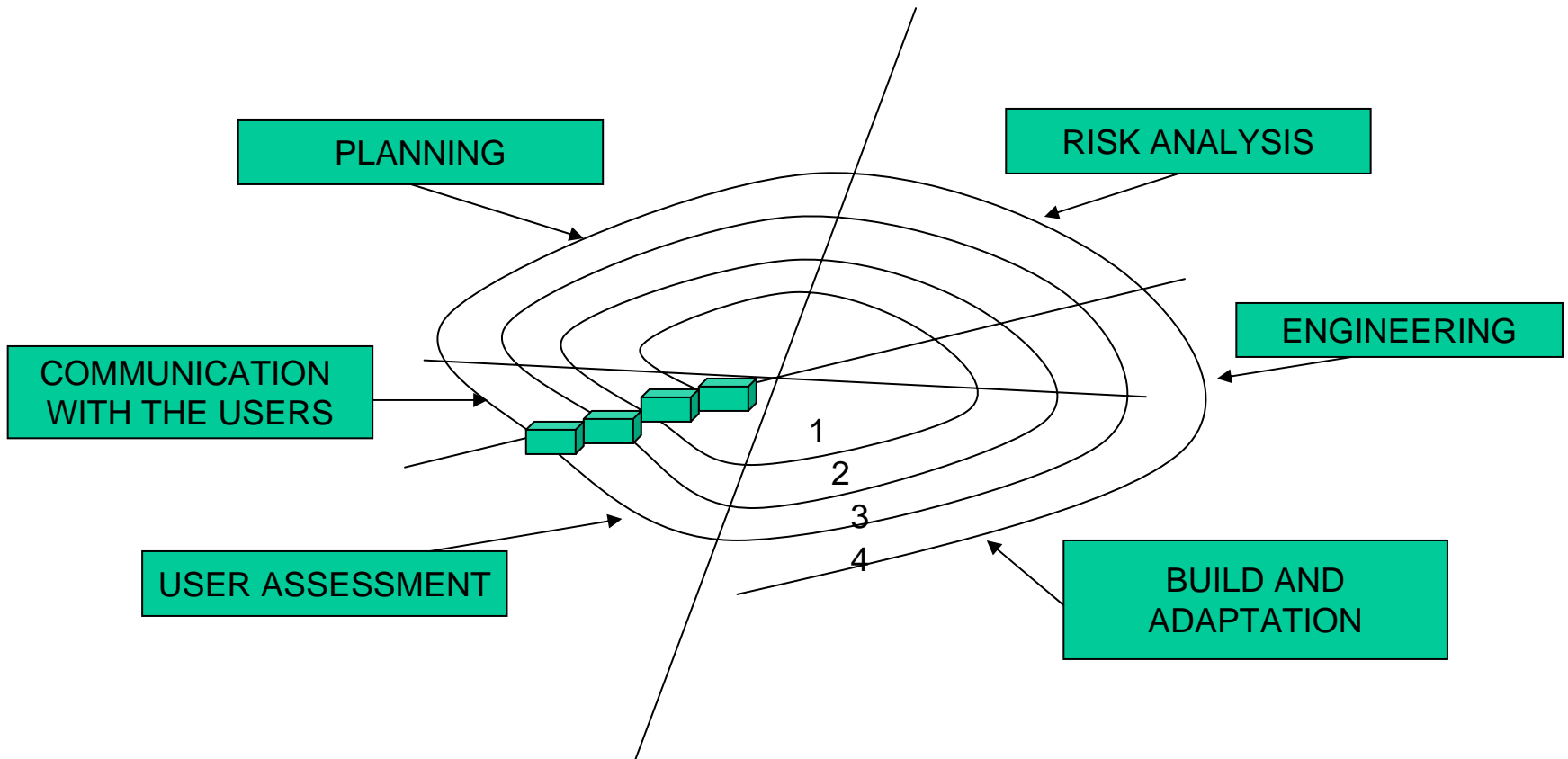
- **“THE SUCCESS KEY OF THIS MODEL CONSIST IN DEFINING AT THE BEGINNING THE GAME RULES, THAT MEANS, STABLISHING A GOOD USER-PROGRAMMER COMMUNICATION”.**

EVOLUTIVE MODELS. THE SPIRAL MODEL

- It was proposed by Boehm. [BOE 88]*
- The spiral model is:
 - Iterative like prototypes.
 - Presents some controlled and systematic aspects like waterfall model.
- First iterations: prototype
- Last iterations: more complete version

*Boehm, B. "A Spiral Model for Software Development and Enhancement", Computer, vol. 21, n° 5 Mayo 1988.

THE SPIRAL MODEL



1. CONCEPT DEVELOPMENT
2. DEVELOPMENT OF NEW PRODUCTS
3. IMPROVING THE PRODUCTS
4. PRODUCT MAINTENANCE

SOFTWARE QUALITY. RELIABILITY AND CORRECTNESS

- **SOFTWARE QUALITY:** that means a correct and reliable implementation.

CORRECTNESS

- **An algorithm is partially correct** if given an input, it is possible to guarantee that the algorithm will reach an output, satisfying both set of data the algorithm specification, once the algorithm finishes but without making sure this last premise.
- **An algorithm is completely correct** if it is partially correct and beside it is also possible to guarantee that this one finishes.

SOFTWARE TESTING

- Software Testing consist in:
 - Checking the proper application operation regarding only a few check data sets.
 - The testing process does not avoid the presence of more errors.

SOFTWARE RELIABILITY

- Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment.
- Software Reliability differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection.
- The high complexity of software is the major contributing factor of Software Reliability problems.