

# MODELLO A CASCATA

ESIGENZE

Cliente

STUDIO DI  
FATTIBILITÀ

Cliente + Analisti

ANALISI DI  
SPECIFICHE

"

PROGETTAZIONE

Analisti + Progr.

SVILUPPO

Progr.

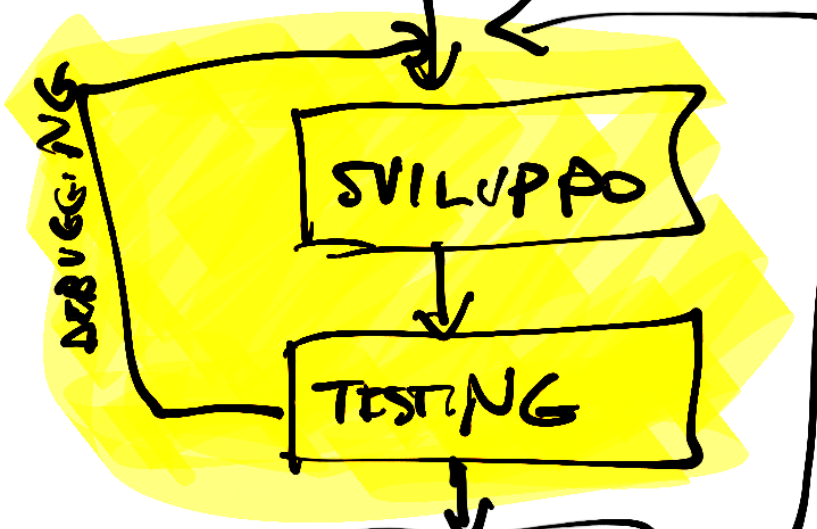
TESTING

Tester

RLASCIO

Cliente

DEBUGGING



# LINGUAGGI

- LM

- LINGUAGGI AD ALTO LIVELLO

  - IMPERATIVI

    - (Fortran, COBOL, ...

    - C, ..., Go, ...)

  - ORIENTATI AGLI OGGETTI

    - (Java, C++, C#, ...)

  - FUNZIONALI

    - (Lisp, ML, Haskell, ...)

  - LOGICI

    - (Prolog, ..., SQL, ...)

TRADUZIONE

- COMPILAZIONE

SORGENTE



COMPILATORE



ESEGUIBILE

hello.go

go build hello.go

./hello

- INTERPRETAZIONE

SORGENTE



INTERPRETE

hello.py

python hello.py

**Go** (Golang)

- Robert Griesemer  
2007 { Rob Pike  
Ken Thompson

- LAUNCHPAD DEC 2009

- "C for the 21st century"  
(Ken Thompson)

<http://golang.org/>

→ TUTORIAL

→ DOWNLOAD !!

→ DOCUMENTATION

→ GO PLAYGROUND

<http://play.golang.org/>

- INSTALLARE Chrome
- INSTALLARE Atom
- INSTALLARE go
- METTERE proapt dei comandi nella barra applicazioni

hello.go

```
package main
import "fmt"
func main() {
    var m, i int
    fmt.Println("Quantas vezes? ")
    fmt.Scan(&m)
    for i=0; i<m; i++ {
        fmt.Println("Ciao")
    }
}
```

```
$> go build hello.go ] COMPILAR & EXECUTE
$> ./hello
```

```
$> go run hello.go ] RUN
```

```
$> go help
$> go doc fmt.Println
$> go fmt hello.go
```