

LINGUAGGI AD ALTO LIVELLO

- COBOL
- FORTRAN
- C
- GO

LINGUAGGI
IMPERATIVI

- LISP
- ML
- Haskell
- Scheme

LINGUAGGI
FUNZIONALI

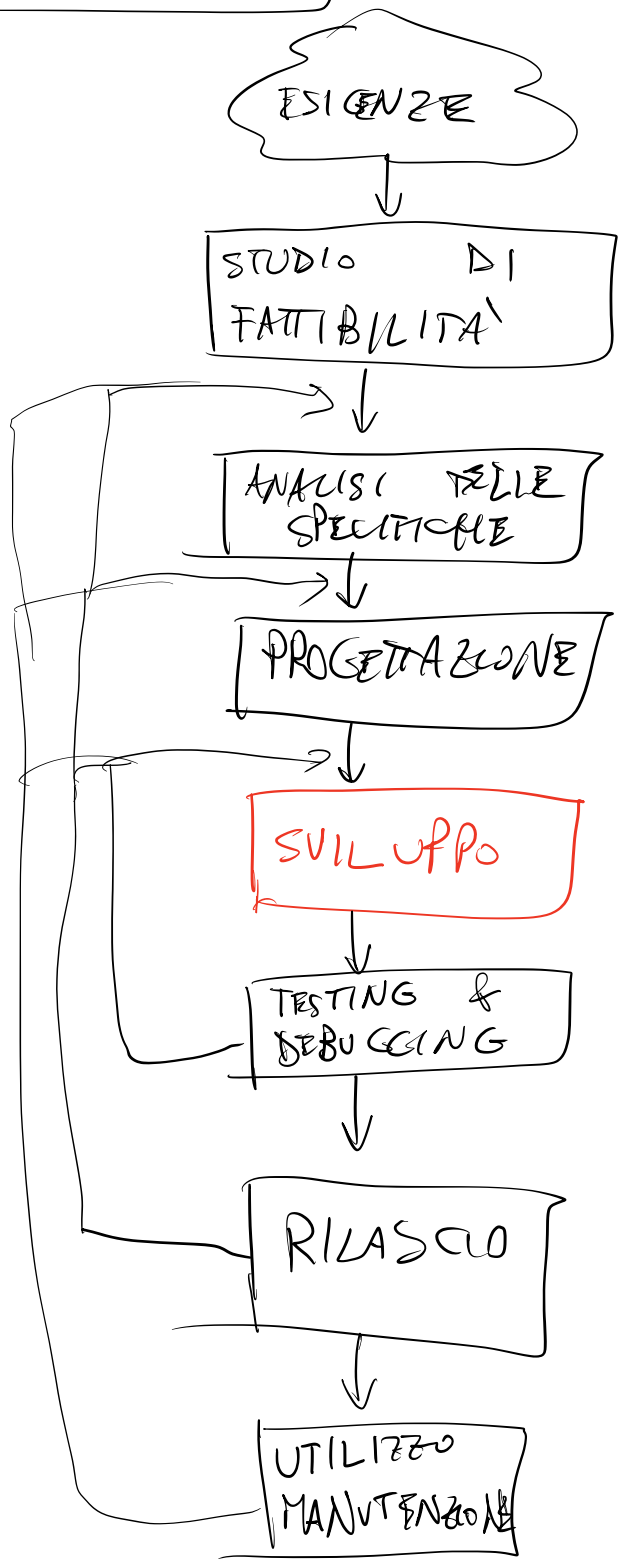
- PROLOG
- SQL

LINGUAGGI
DICHIARATIVI
(LOGICI)

- JAVA
- C++
- C#
- ...

LINGUAGGI
OBJECT-ORIENTED

CICLO DI VITA DEL SW



Clienti

Commerciale + Analisti

Analisti

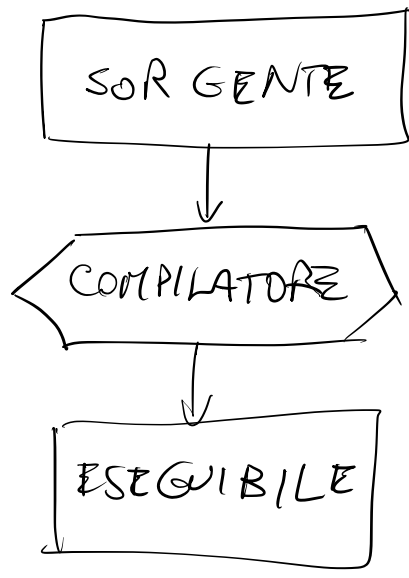
Analisti + Programmatori

Programmatori

Tester

TRADUZIONE

- COMPILAZIONE

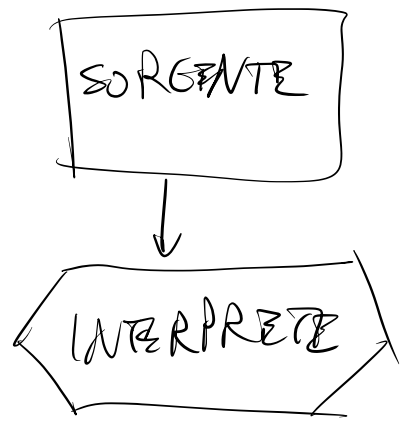


Editor hello.go

go build hello.go

./hello
(o hello.exe)

- INTERPRETAZIONE



Editor hello.py

python hello.py

STRUMENTI

→ Editor
(Atom, emacs, ...)

→ Compilatore

Go

- 2007 Google
 - Robert Griesemer (Java Hotspot VM)
 - Rob Pike (Unix, Plan 9, Inferno)
 - Ken Thompson (Unix, C, UTF-8)
- 2009 lanciato

package main

hello.go

import "fmt"

func

main() {

var n, i int

fmt.Print("Quante volte, padrone? ")

fmt.Scan(&n)

for i=0; i<n; i++ {

fmt.Println("Sono il tuo schiavo")

}

}

\$> go build hello.go

\$> ./hello

Quant' te volte, padrone? 5

Sono, il tuo schizzo

\$> go run hello.go

\$> go doc fmt.Print

\$> go help

\$> go fmt hello.go