

# Programming in the large

Giorgio Brajnik

`giorgio.brajnik@uniud.it`

Ott 2021

# Attività varie

- designing
- programming
- refactoring
- debugging
- testing
- production
- maintenance
- documenting

# Programming

- Programming in the small
  - 1 persona, tutto sotto controllo, tiene a mente tutto
- Programming in the large
  - tante persone (100, 10000, ...)
  - disperse nello spazio e tempo
  - sistema troppo grosso per essere compreso da 1 solo ( $>10^6$  LOC)
  - varie tecnologie

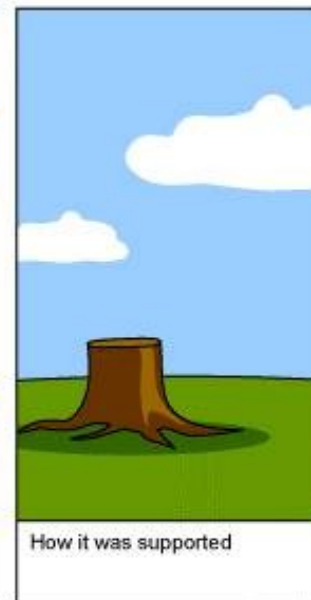
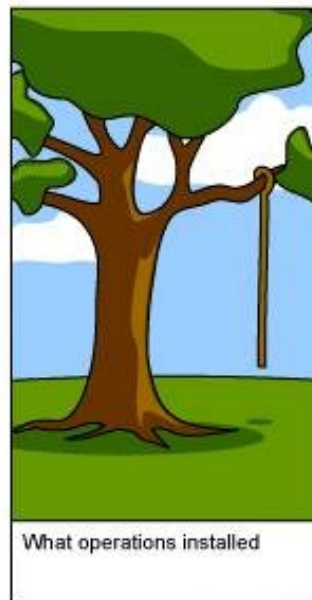
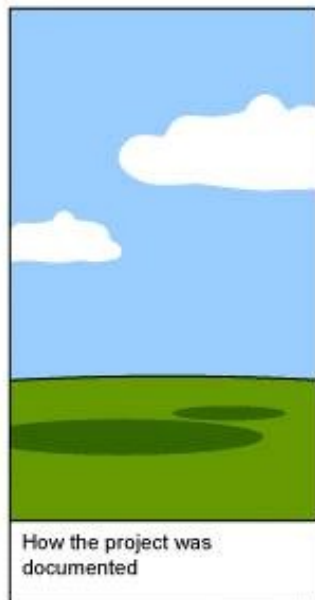
# Complessità dello sviluppo del software

- tutto è virtuale
  - alla fin fine dispositivi che lanciano segnali ad altri dispositivi
- materiale di partenza effimero
  - requisiti volatili
  - di cose che gli utenti/committenti non hanno nemmeno ancora visto
  - linguaggi di programmazione, librerie e framework complessi di per sé

# Effetto?

- fraintendimenti
- overhead x comunicazione
- incertezza relativamente alle parti fatte da terzi

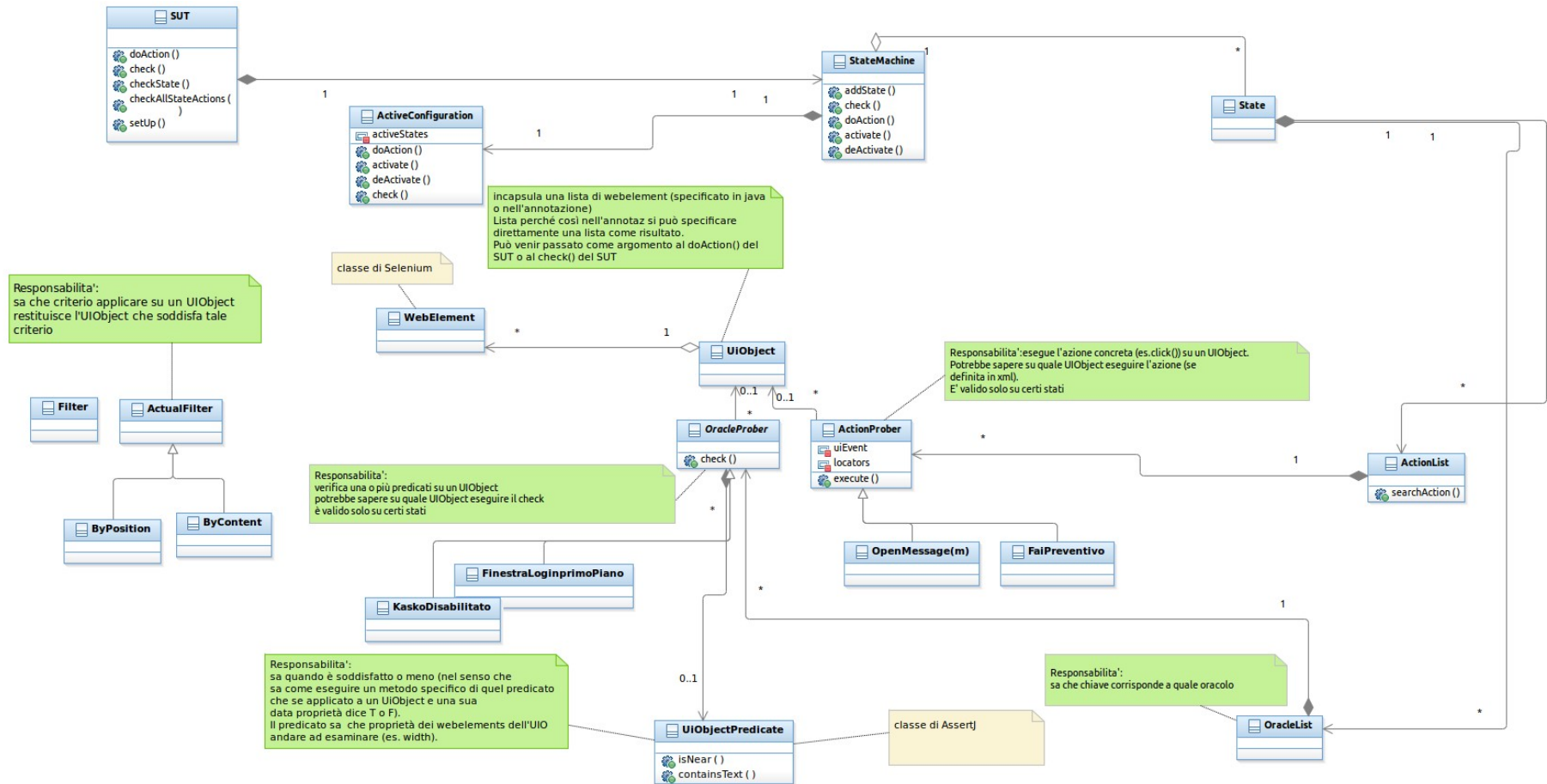
# Ingegneria dei requisiti



# Documenting

- Come si fa?
- Funziona?

# Es. di documentazione





# Documenting

- Problematiche?
  - out of sync rispetto al codice
  - file diversi, gestiti in maniera diversa
  - diversi formati, diversi linguaggi

The code is the  
documentation

**e UML?**

# Documenting

- UML va molto bene
  - per capire, per progettare, per comunicare
  - **singole parti** del design
- è difficile usarlo come unica rappresentazione di tutta la documentazione

# Change management

## i requisiti cambiano!

- perché non li avevamo capiti
- perché il cliente/committente cambia idea
- perché cambia il mercato
- perché è arrivato un concorrente nuovo
- perché qualche sistema esterno è cambiato
- perché è cambiato qualche sottosistema interno/libreria

# Change management

- è inevitabile che occorra modificare il sw
  - e quindi per ogni modifica
    - l'overhead di comunicazione cresce
    - la probabilità di fraintendimenti anche

Refactoring!

# Cosa dovete fare

- padroneggiare un tool di sviluppo, IDE: intellij, eclipse, visual studio code, ...
- padroneggiare un tool di gestione delle versioni: git, mercurial, ...
- padroneggiare Java
- scaricare un progetto open source e lavorarci sopra: capirlo, decidere cosa modificare, e farlo