

# **aMULEto**

**progetto di ing. del software**

# Obiettivo

- Da progetto java a UML
- Visione d'insieme dello sviluppo in ogni istante
- Flessibile (terminale/GUI)

# Problematiche

- Papyrus non sempre funziona
  - Mancanza di shortcut per operazioni di base:
    - No cmd+c, cmd+v per attributi e metodi
    - No name editing da tastiera
    - No integrazione con plantUml
    - No 'Generate Java code from UML' (sul mio computer)
- Gestione dei file con eclipse (crash e riavvii)
  - Più efficiente VSCODE + Java ext. pack

# Standard usati e tools

- Standard di programmazione Java (Oracle way)
- Tools:
  - VSCode + Maven
- Dipendenze:
  - JUnit
  - JavaParser
  - PlantUML
  - GSON
  - Batik

# SW Configuration Management

- GitHub
- Git
- Maven

# SW Life Cycle (1)

- Metodologia Agile
- Requisiti:
  - Funzionali
    - GUI
    - Terminale
  - Non Funzionali
    - Prestazioni
    - Portabilità
    - Manutenibilità

# SW Life Cycle (2)

- Architettura del sistema:
  - Pattern MVC:
    - Model: rappresentazione di classi UML, attributi, metodi e relazioni
    - View: GUI (canvas, dialog, menu)
    - Controller: gestione eventi, aggiornamento diagrammi

# SW Life Cycle (3)

- Tecnologia:
  - Java e Swing per la GUI
  - Maven: per la gestione del progetto e delle dipendenze
  - PlantUML: per la generazione di diagrammi UML
  - GSON: serializzazione di diagrammi in JSON
  - Batik: rendering SVG



# SW Life Cycle (4)

- Implementazione:

- Package principali:

- com.fabio.org.amuleto: classe principale App
    - com.fabio.org.amuleto.view: componenti GUI
    - com.fabio.org.amuleto.model: definizione modelli UML
    - com.fabio.org.amuleto.utils: funzioni utilitarie
    - com.fabio.org.amuleto.serialization: salvataggio dei diagrammi
    - com.fabio.org.amuleto.controller: gestione logica model e view

# SW Life Cycle (5)

- Approccio allo sviluppo:
  - Implementazione iterativa con integrazione aggiuntiva
  - Utilizzo di JUnit per validare le funzionalità di ciascun modulo
  - Creazione di un fat jar tramite Maven-shade-plugin

# SW Architecture

- Layered Architecture basata su MVC
- Il progetto include ulteriori componenti come:
  - Converter: per l'analisi del codice java e la generazione di UML
  - Utils: supporto al parsing

# Requirements (1)

- Requisito principale di informazione per l'elicitazione:
  - Gli utenti principali sono sviluppatori e architetti sw.
- Tecniche di elicitazione utilizzate:
  - Interviste
  - Analisi delle attività

# Requirements (2)

- Classificazione “MoSCoW”:
  - Must Have:
    - Generazione automatica dei diagrammi UML a partire dal codice Java
    - Interfaccia grafica interattiva per la modifica e il posizionamento degli elementi UML.

# Requirements (3)

- Classificazione “MoSCoW”:
  - Should Have:
    - Funzionalità avanzate di log e tracking per il salvataggio/caricamento dei diagrammi (in formato JSON).
    - Esportazione dei diagrammi in formati vettoriali (es. SVG).

# Requirements (4)

- Classificazione “MoSCoW”:
  - Could Have:
    - Integrazione con strumenti di analisi della qualità del codice (ad es. SonarLint, PMD) per correlare la struttura UML con la qualità del codice sorgente.

# Requirements (5)

- Classificazione “MoSCoW”:
  - Won’t Have:
    - Funzionalità non strettamente legate alla generazione e gestione dei diagrammi UML, come sistemi di gestione completa del progetto o moduli di collaborazione in tempo reale.



# UML Modeling (1)

- **Use Case Diagram:** describe la funzionalità prevista dal sistema in fase di sviluppo
- **Class Diagram:** mette in relazione gli oggetti di un sistema
- **State Machine Diagram:** utilizzato per modellare i possibili stati di un sistema e le sue transizioni

# UML Modeling (2)

- **Sequence Diagram:** utilizzato per descrivere sequenze di specifiche dell'evento
- **Communication Diagram:** mostra i messaggi che gli oggetti si inviano
- **Activity Diagram:** specifica il comportamento definito dall'utente
- **Components Diagram:** mostra le relazioni tra i diversi componenti di un sistema
- **Package Diagram:** mostra l'organizzazione e la disposizione degli elementi in pacchetti

**I diagrammi sono  
caricati qui:**

**[https://github.com/fabiobassini/aMULeto/  
tree/main/docs/diagrams](https://github.com/fabiobassini/aMULeto/tree/main/docs/diagrams)**