



# Progetto Ingegneria del software

*PantryManager*

Gabriele Chignoli

# Obiettivo

---

Alleggerire la gestione alimentare in ambito familiare, fornendo supporto alla gestione della dispensa e della dieta personale

- Dispensa dei prodotti
- Profili con obiettivi personalizzati
- Ricettario
- Suggerimenti per i pasti



# Difficoltà incontrate

---

- Stima risorse per soddisfare requisiti
- Approccio nuove tecnologie (Hibernate, Maven, Swing)
- Documentazione aggiornata
- Papyrus
- Test

# Paradigma di programmazione/modellazione utilizzato e tools



---

## *PROGRAMMAZIONE*

- OOP (Java)
- Eclipse (IDE)
- Maven

## *MODELLAZIONE*

- Papyrus (UML)

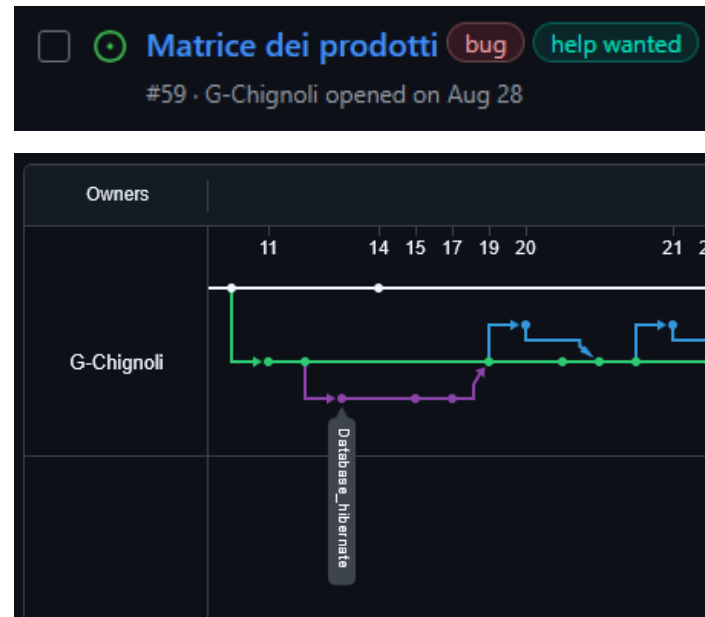
## *DOCUMENTAZIONE*

- Latex (OverLeaf)
- Markdown (GitHub)

# Software configuration management

## GitHub

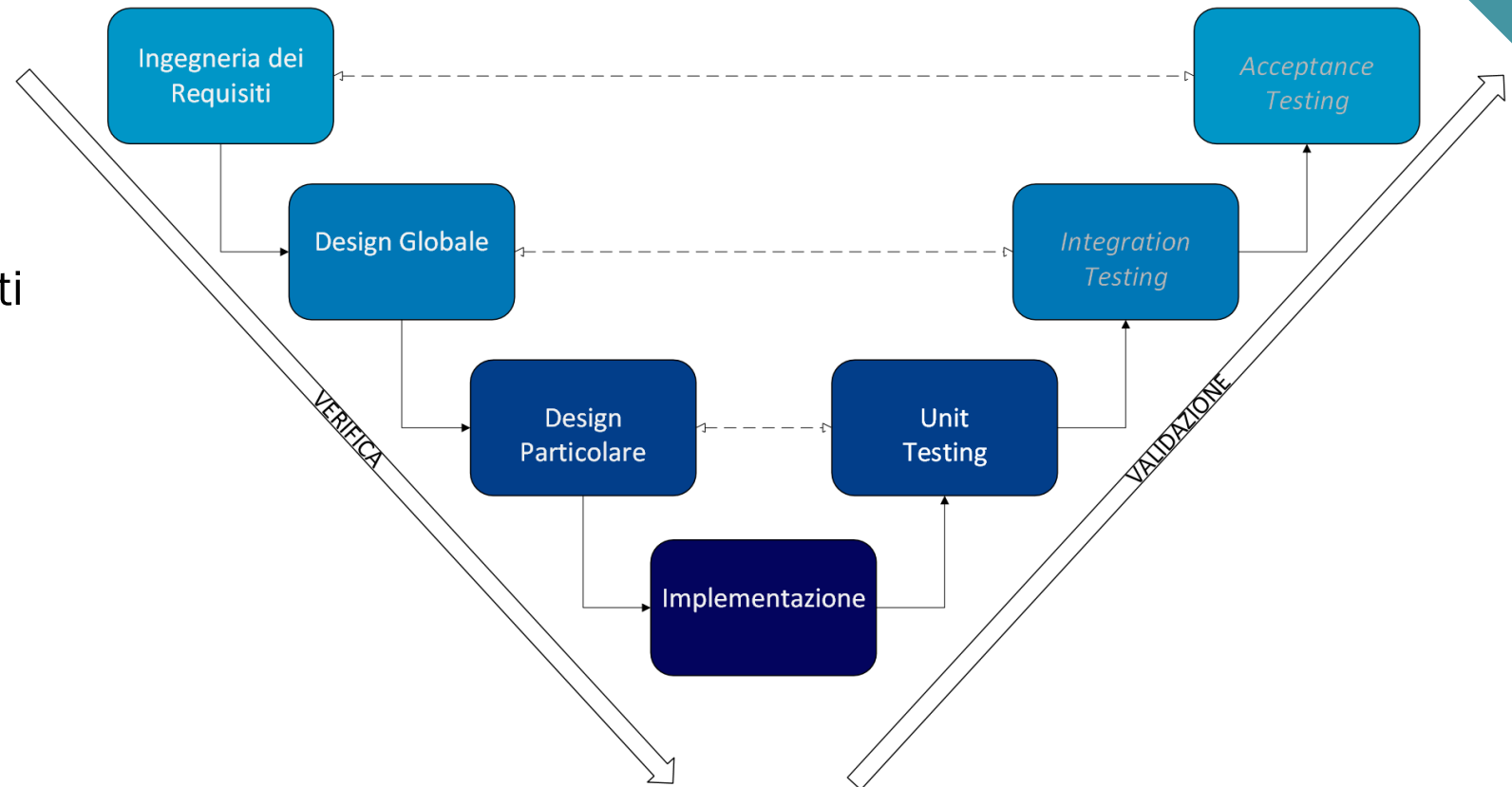
- Piccoli commits
- Branch per ogni modifica importante
- Issue per le richieste
- Uso dei tag
- Kanban board



# Software life cycle

## MODELLO A V

- Metodi puramente tradizionali troppo soffocanti
- Validazione non formalizzata esplicitamente
- Manutenzione (refactoring)
- Modello a cascata?
- Modello a spirale?



# Requisiti

## ELICITAZIONE

- Carattere etnografico
- Scenari (Casi d'Uso)

## QUALITÀ

- Correttezza
- Usabilità
- Riutilizzabilità

## SPECIFICA DEI REQUISITI

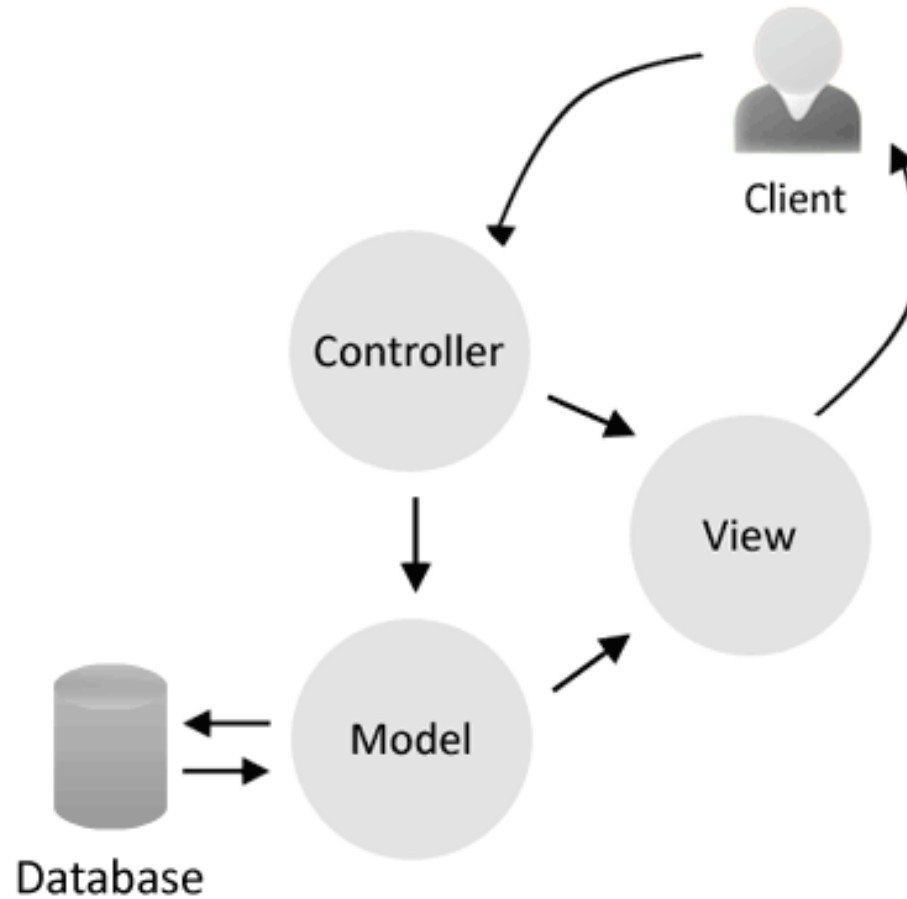
- Tracciabilità
- Priorità (MoSCoW)
- Implementazione

ID	Descrizione	Priorità	Impl.
FUNZ.1	Il sistema dovrebbe utilizzare una <i>Dispensa</i> per tenere traccia dei prodotti in possesso dell'utente.	M	✓

# Architettura

## *MODEL VIEW CONTROLLER (MVC)*

- Modularità (interfaccia grafica, calcolo, interazione con il DB)
- Manutenzione
- Flessibilità





# Design pattern

## *Singleton*

Avere una sola istanza dell'interfaccia

```
public static MainWindow getInstance() {  
    if(instance == null) {  
        instance = new MainWindow();  
    }  
  
    return instance;  
}
```

## *Observer?*

Subject -> Componenti Swing (JButton...)

Observer -> Listener

## *Analisi statica*

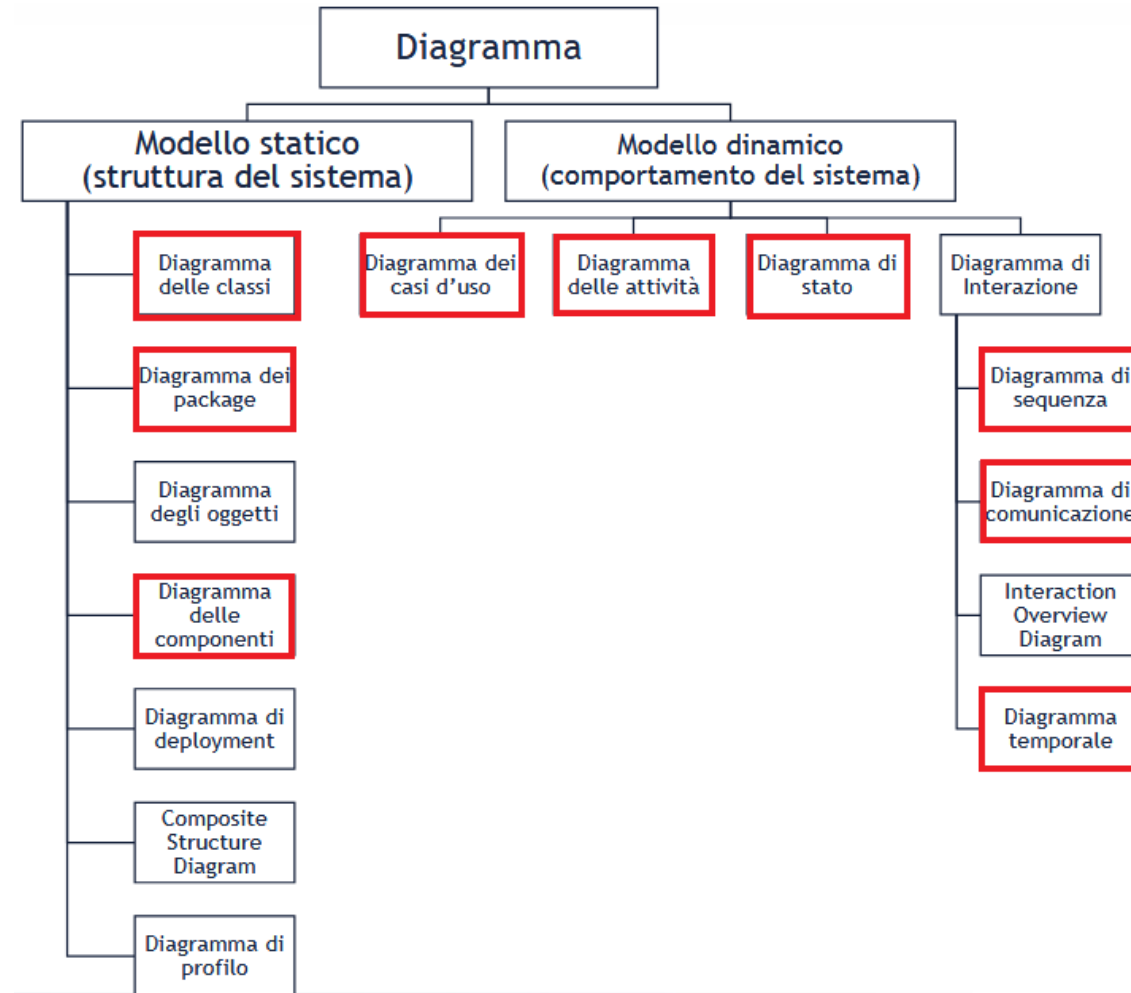
Con Stan4j e CodeMR

Metrica	Valori
Cyclomatic Complexity (CC)	1.59
Fat	0
Afferent Coupling (Ca)	1
Efferent Coupling (Ce)	6
Instability (I)	0.86
Abstractness (A)	0
Distance (D)	-0.14

# Modellazione

## UML

- Diagramma dei casi d'uso
  - Cosa permette di fare
- Diagramma delle classi
  - Struttura effettiva dei package
- Diagramma di sequenza
  - Operazioni effettive dei componenti



# Implementazione

## *Eclipse*

- Java
- SonarQube
  - Qualità nel workflow
- Maven
  - Struttura progetto
  - Gestione dipendenze
  - Package (fat-jar)
- H2
- Hibernate (ORM)
- JPA (Jakarta Persistence)
- Log4j
- Swing

IMPLEMENTATO	NON IMPLEMENTATO
Interfaccia Grafica	Gestione Profili
Gestione Prodotti	Gestione Dieta
...	...

# Testing

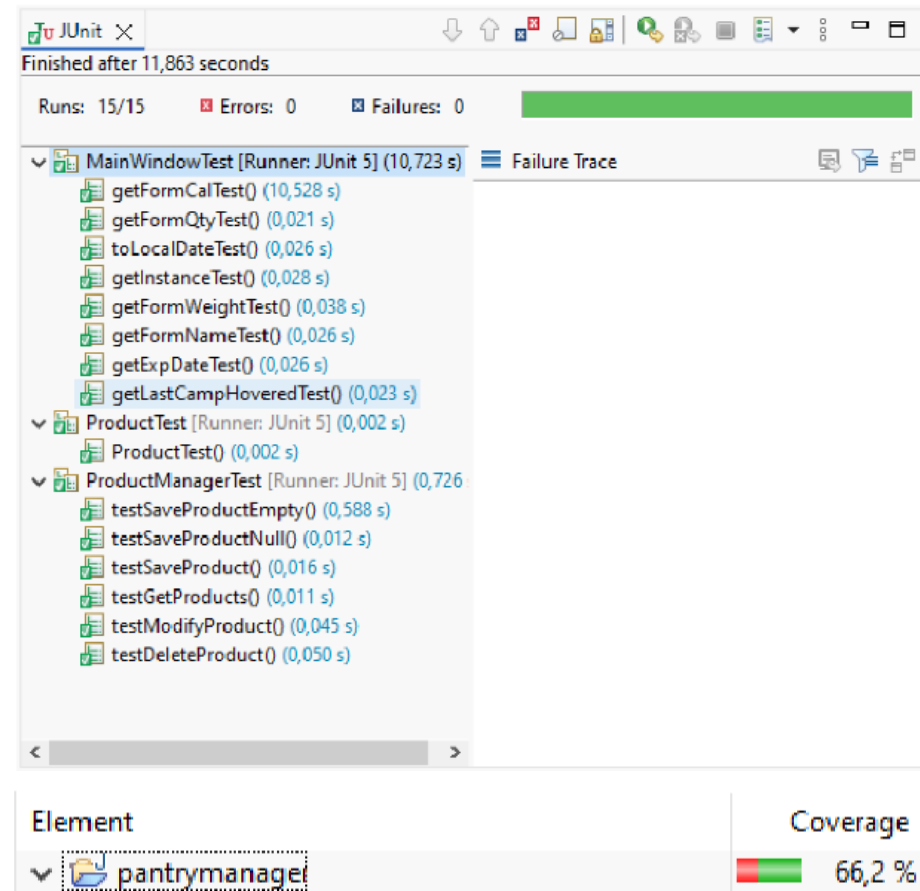
## *JUnit*

### Test automatico

- Test di unità
- Test di integrazione
- Test di copertura

### Test dell'interfaccia grafica

- Come?
- Test manuale



# DEMO



---

<https://github.com/G-Chignoli/PantryManager/releases/tag/alpha>