

# Software Testing and Validation

## A.A. 2025/2026

### Testo del Progetto

Igor Melatti

## 1 Come si consegna

Il presente documento descrive le specifiche per il progetto d'esame. La consegna deve consistere in un singolo file `STV_2025_2026_matricole.zip` (se il progetto è fatto in gruppo, scrivere tutte le matricole separate dall'underscore `_`), che contenga un'unica directory `STV_2025_2026_matricole`, la quale a sua volta deve includere:

- un file PDF `relazione.pdf` con le seguenti caratteristiche:
  - deve indicare nome, cognome e matricola di ogni studente/studentessa del gruppo;
  - deve descrivere come il progetto sia stato svolto, indicando quali tecniche sono state scelte e perché ;
- un file PDF `presentazione.pdf` da usare come slide per illustrare lo svolgimento del progetto (assumere di avere 20 minuti di tempo massimo);
- una directory `progetto` contenente tutti i file che fanno parte del progetto, con un'opportuna organizzazione in sottodirectory da spiegare in `relazione.pdf`.

Il suddetto file `STV_2025_2026_matricole.zip` andrà poi inviato per email al docente `igor.melatti@univaq.it`.

È possibile consultarsi con i compagni. Tuttavia, occorre che ciascun gruppo presenti una propria soluzione personale.

## 2 System Under Verification

Il sistema da verificare (SUV) può essere scelto tra le seguenti opzioni:

- NuSMV, scaricabile da <https://nusmv.fbk.eu/downloads.html>;
- SPIN, scaricabile da <https://spinroot.com/spin/Archive/>;
- CMurphi, scaricabile da <https://raise.uniroma1.it/software/cmurphi/> (solo per gruppi da 1 o 2 persone);
- FHP-Murphi, scaricabile da <https://raise.uniroma1.it/software/fhp-murphi/> (solo per gruppi da 1 o 2 persone).

## 3 Esercizio per Gruppi da 3 Studenti/Studentesse

Occorre soddisfare le seguenti richieste:

1. Nel SUV scelto, effettuare
  - unit testing: selezionare 6 unità (spiegare perché sono tali) e testarle con le tecniche che si ritengono opportune, tra quelle viste a lezione (*category-partition*, *catalog-based*, etc.);
  - integration testing: scegliere 3 insiemi di sottounità da testare insieme (possono includere anche unità non considerate nel punto precedente);
  - system testing: testare l'intero sistema;
  - regression testing: considerare anche la penultima versione del software e spiegare su quali parti si possono usare le tecniche di regression testing viste a lezione.
2. Selezionare una unità e verificare la coverage relativa a MC/DC e loop boundary, instrumentando opportunamente il codice.

## 4 Esercizio per Gruppi da 2 Studenti/Studentesse

Come sopra, con le seguenti facilitazioni:

- nella parte di unit testing, selezionare 4 unità;
- nella parte di integration testing, selezionare 2 insiemi di unità;
- non considerare il regression testing.

## 5 Esercizio per Singoli Studenti/Studentesse

Come sopra per i gruppi da 2, con le seguenti ulteriori facilitazioni:

- nella parte di unit testing, selezionare 2 unità;
- nella parte di integration testing, selezionare 1 insieme di unità.