

Verifica – parte I

Rif. Ghezzi et al
6.1 -6.2

Convalida e verifica

- Accezioni più comuni:
 - convalida: il prodotto è quello giusto (cioè rispetta i suoi veri requisiti)?
 - verifica: il prodotto (o ogni artefatto intermedio) è costruito nel modo giusto (cioè rispetta la sua specifica)
- Useremo solo il termine “verifica”
- In inglese, Validation & Verification (V&V)

Necessità della verifica

- Ogni fase del processo di produzione del software è soggetta a errore.
- Quindi è necessario verificare
 - ogni artefatto (dal documento dei requisiti al codice sorgente)
 - ogni fase del processo, compresa la verifica stessa.

Scopo della verifica

- Lo scopo dell'attività di verifica è convincere della bontà del prodotto (finale o intermedio) chi è interessato ad esso
 - Ad es. il software raggiunge i suoi obiettivi, ma anche è portabile, è efficiente, etc
- Le risorse e le tecniche (formali, sistematiche, informali) da impiegare dipendono dal tipo di prodotto e dal suo valore.

Risultati non binari

- Difficilmente il risultato della verifica è un sì o un no.
- La certezza assoluta del risultato di una verifica è praticamente impossibile da raggiungere, e non sarebbe comunque sufficiente
 - Ad es. garanzia di assenza di errori?
- Nella prassi, un certo grado di non-correttezza è accettabile e accettato.

Verifica oggettiva e soggettiva

- Non tutte le qualità sono facilmente misurabili.
- Verifica oggettiva:
 - osservazione della risposta a input e confronto con la risposta corretta
 - dimostrazione matematica di una proprietà
- Verifica soggettiva:
 - usabilità
 - riusabilità

Verifica di qualità implicite

- Le proprietà desiderabili di un prodotto software dovrebbero essere espresse nel Documento di Specifica dei Requisiti
- La bontà di un prodotto consiste anche nel rispetto di requisiti non specificati, perché impliciti.
 - Robustezza
 - Prestazioni
 - Manutenibilità

Approcci alla verifica

- Sperimentale
- Analitico

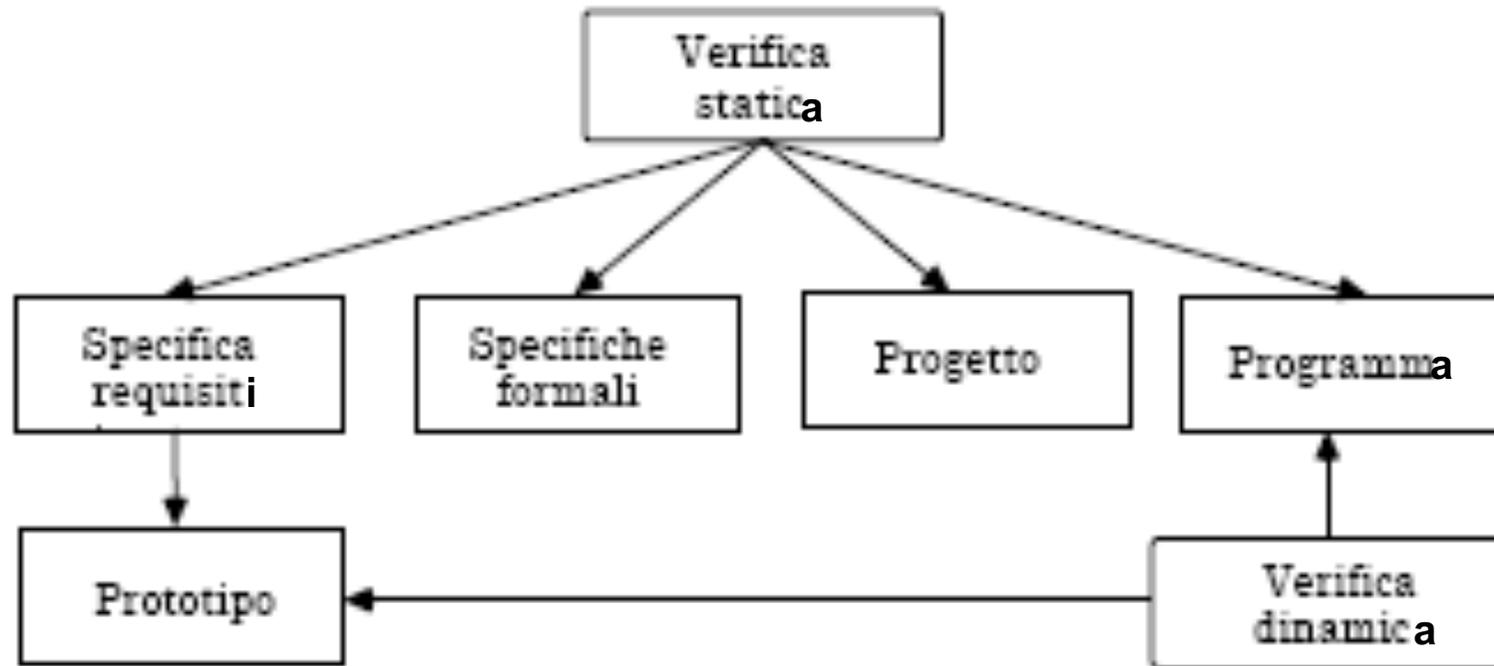
Approcci alla verifica

- **Sperimentale**
 - Osservazione del comportamento del prodotto e confronto con i suoi requisiti
 - Necessità di approccio sistematico
- **Analitico**
 - Osservazione del prodotto e ragionamento sulle sue proprietà

Approcci alla verifica

- **Dinamico (test):**
 - Osservazione del comportamento del prodotto e confronto con i suoi requisiti
 - Necessità di approccio sistematico
- **Statico (analisi):**
 - Osservazione del prodotto e ragionamento sulle sue proprietà
- **Approcci complementari**

Verifica statica e dinamica: applicabilità



Terminologia (IEEE)

- Il funzionamento non corretto di un programma è detto **malfunzionamento** (*failure*)
- E' legato al comportamento di un programma, ma non alla sua struttura statica (il codice)
- Un malfunzionamento è causato dalla presenza di **anomalie** (*fault*) nel programma (**bug** nell'uso comune)
- Esempio: programma che somma X e Y, anomalia se uso l'operatore * al posto dell'operatore +
- Il termine **errore** (*error*) indica, nello standard proposto dall'*IEEE*, la **causa** di un anomalia