

# Appunti di Architetture del Software

**A cura di:**  
Francesco Refolli  
Matricola 865955

**Anno Accademico 2024-2025**

# Chapter 1

## Sistemi Legacy

### 1.1 Situazione

Un sistema legacy e' un sistema che ha ancora una criticita' dopo moltissimi anni, che possibilmente e' stato un software vincente ma che ora e' in decadenza. Spesso usa tecnologie o approcci obsoleti che pero' sono ancora molto applicati nel mondo del software.

E' necessario prendere delle decisioni circa il suo futuro: lo cestino, lo riscrivo, lo miglio ... E' spesso molto costoso sia continuare ad operarlo che cambiarlo. Quindi e' molto importante appoggiarsi a tecniche di reverse engineering per ridurre lo sforzo di riscrittura mantenendo la piena funzionalita' del sistema.

Se la qualita' e il valore del software sono bassi allora il sistema puo' essere cestinato. Se invece l'importanza e' alta allora e' necessario re-ingegnerizzarlo, oppure rimpiazzarlo con un altro sistema esistente che assolve a tutte le funzioni del software dimissionario.

Bisogna valutare le prestazioni, l'eta del sistema, la stabilita' del distributore/manutentore e la frequenza dei fallimenti. Così come le tecnologie utilizzate che siano moderne, che il codice sia leggibile ... etc. Si possono anche raccogliere delle informazioni quantitative come il numero di cambiamenti richiesti, il numero di anomalie (sondate con tool o difetti emersi), UI utilizzate e volume dei dati del sistema.

**Definizioni** Il **Reverse Engineering** produce una versione del sistema che e' piu' facile mantenere. Mentre il **Refactoring** consente di riscrivere del codice mantenendo le sue funzioni.

### 1.2 Leggi di Lehman e Belady

-