# 02- PROCESSO SOFTWARE

* Attività:
  + Produzione: requisiti, design, implementazione
  + Verificazione
  + gestione
* Fasi:
  + sviluppo
  + operazioni
  + mantenimento

PRODURRE SOFTWARE:

Produrre **documentazione**, gestire i **dati**, scrivere **codice**

Ci sono proprietà ben definite e predicibili:

* costo
* durata

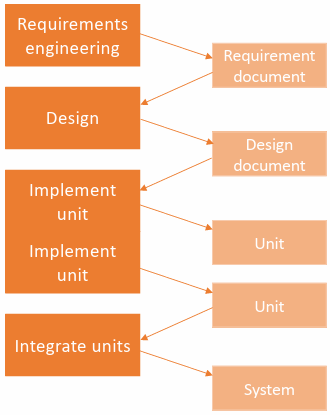
Ci sono proprietà del prodotto:

* funzionalità
* manutenibilità

#### BOTTOM UP

**parto dal basso** per ottenere il codice eseguibile

🡪 scrivo il **codice sorgente**:

* unità fisiche
  + file e cartelle
* unità logiche
  + funzioni
  + classi
  + packages
  + sottosistemi
* definite e organizzate attraverso il design

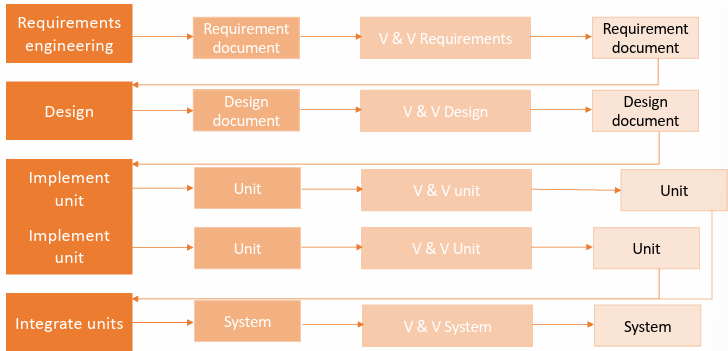
##### Attività di produzione:

* ingegnerizzare i **requisiti** 🡪 cosa deve fare il sw
* **architettura e design** 🡪 quali unità e come organizzarle
* **implementazione** 🡪 scrivere codice sorgente
* integra unità e fa **test**

**queste attività dipendono dalle precedenti** 🡪 quindi si usa il modello **waterfall**: finché la prima attività non è conclusa, non puoi passare alla successiva

*NOTA: requisiti e design sono scritti in documenti*

#### Attività di validazione e verifica

* controllare che i **requisiti siano corretti**
  + esternamente*: abbiamo compreso cosa voleva l’utente?*
  + **Internamente*: il documento è consistente?*
* controllare che il **design sia corretto**
  + esternamente*: il design è capace di supportare i requisiti?*
  + internamente*: il design è consistente?*
* controllare che il **codice sia corretto**
  + esternamente: il codice è capace di supportare i requisiti e il design?
  + Internamente: *il codice è consistente? (check sintattici)*

*Dunque, bisogna definire le varie attività e chi ha il potere di modificarle/usarle.*

#### Attività di gestione

* Gestione del **progetto**
  + Assegnare lavoro e monitorare progresso
  + Stimare e controllare budget
* Gestione della **configurazione**
  + Identificare, salvare i documenti e le unità
  + Mantenere traccia delle relazioni e della storia
* Assicurazione della **qualità**
  + Definire obbiettivi di qualità
  + Definire come il lavoro sarà fatto
  + Controllare i risultati

## FASI

Dietro lo sviluppo del codice:

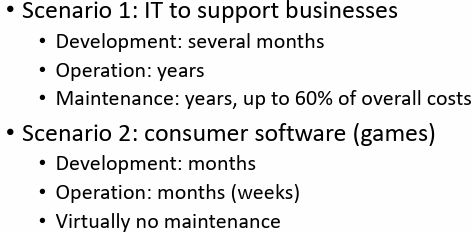
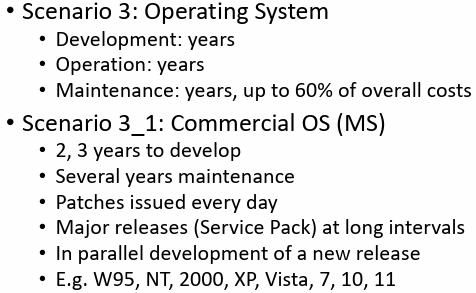
* Lo sviluppo è solo la prima parte del gioco
  + **Operare software:**
    - Sviluppo, operazioni
  + **Modifiche del software:**
    - Mantenimento
  + **End up**
    - Retirement: mando in pensione un sw e creo una nuova versione

*→ devo dare modo al cliente di poter avere i suoi dati sul sw nuovo*

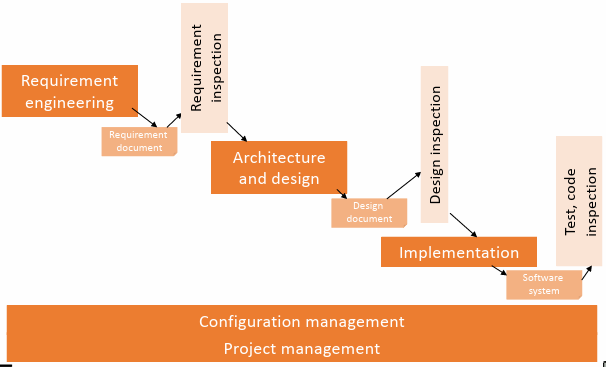
#### MAINTENANCE

* **Sequenza di sviluppi**
  + Il primo solitamente è lungo e vincola i successivi
  + Gli altri sono evolutivi, piccoli rispetto al primo ma con le stesse fasi
  + Dopo molte modifiche, bisogna ripartire da 0.

#### SCENARIO IN DEV/MAINTENANCE /OPERATION



**Dunque, le varie fasi sono:**



#### SISTEMA VS SOFTWARE

**Immagine che contiene testo, schermata, Carattere, diagramma

Descrizione generata automaticamenteSISTEMA** = software + hardware

System Integration Test: prendo hw+sw, metto insieme e testo

Dunque:

* **Attività**
* **Documenti**
* **Tecniche**
* **Linguaggi**
  + Per fare documenti (UML) +codice
* **Modelli**
  + Per guidare e supportare le attività
  + Per ottenere le certificazioni

Possono essere adottati **diversi approcci**:

* Cow Boy 🡪da non fare
  + Parto a scrivere codice in maniera disordinata
* Basato sui **documenti**
  + Uso UML → linguaggio semi formale con qui definisco proprietà e caratteristiche e man mano inizio a trasformare
* Basato su modelli formali
  + Molto formale, usato in ambiti molto specifici
* Agile
  + Basato su documenti con enfasi su codice e test

**Trend di sviluppo**

* **Software as a Service** → fornire degli applicativi che forniscano sw autogenerato o aiutano nello sviluppo
* **Cloud**
* **Devops**
* **Outsourcing**: non ho la risorsa sw all’interno dell’azienda, vado a dare ad un’altra azienda
* **Agile**

**Business model**

* **ASP** → paga per usare
  + Usi sw, quando non ti interessa più smetti di pagarlo e non lo usi
* **Freeware** → applicazione pro-gratuita per un po’ e poi se vuoi avere tutte le funzioni devi pagare
* **Shareware** → gratis ma ti chiedo di pagare
* **Adware** → sw è gratis ma con pubblicità