

Tecniche di programmazione ASP.NET

Corso ENAIP - 19 febbraio 2019

Contatti

- ▶ Daniel Maran
- ▶ Software Architect in Gruppo Euris S.p.A.
- ▶ daniel.maran@euris.it
- ▶ 3489390751

Fondamenti di programmazione

Corso ENAIP - 19 febbraio 2019

Da Wikipedia...

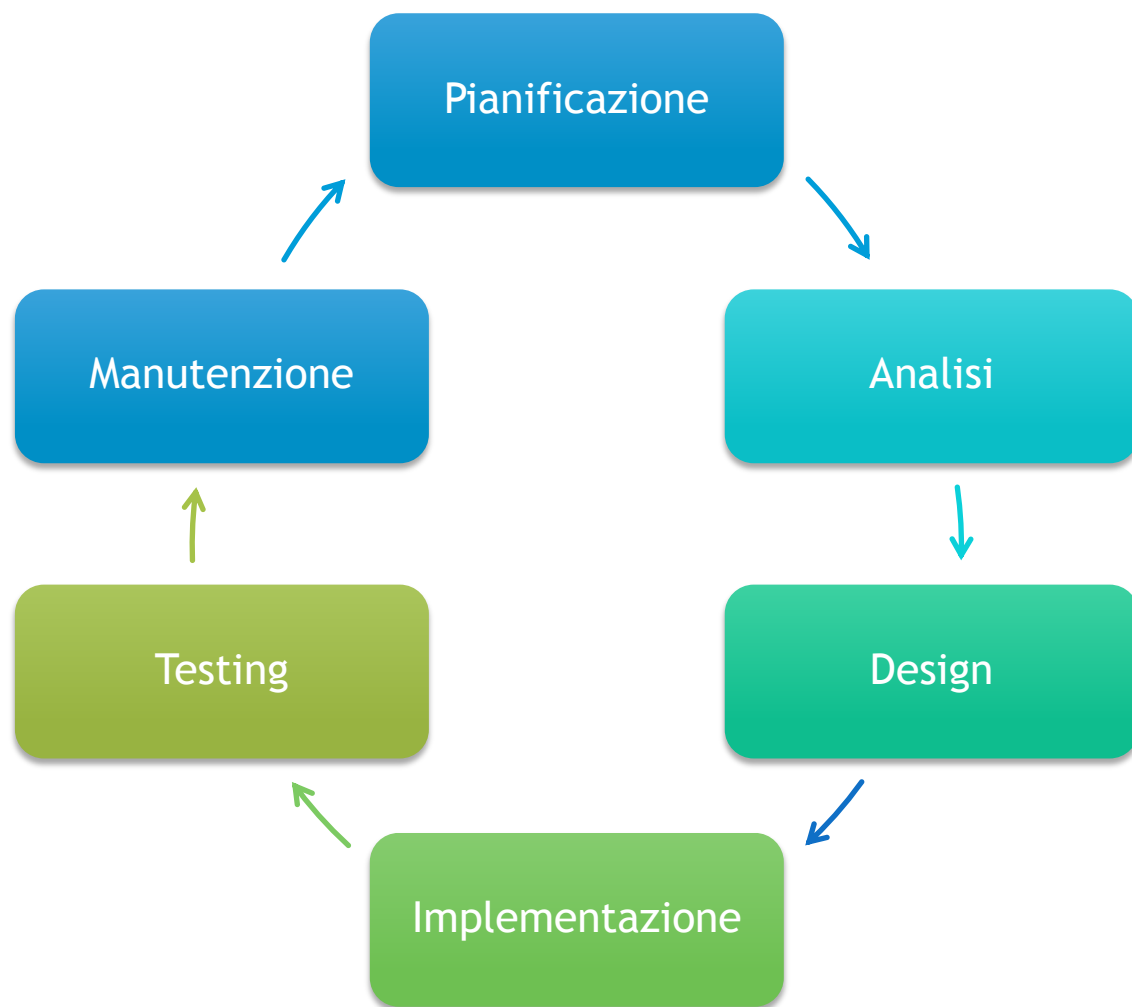
*Con fondamenti di programmazione, in informatica, si indicano le **conoscenze basilari** che un programmatore deve possedere per la programmazione in un qualunque linguaggio di programmazione.*

► Quali conoscenze?

Il software

- ▶ Cos'è un software?
 - ▶ Le informazioni utilizzate da uno o più sistemi informatici, ovvero programmi e/o dati
- ▶ Come si costruisce?
 - ▶ Componendo programmi o parti di essi e dati per costruire un sistema più o meno complesso che soddisfa delle necessità
- ▶ Qual è la vita di un software?

La vita di un software



La vita di un software - 2

Immaginiamo un sistema che ogni X minuti scarica da un sito dei dati, li manipola in qualche modo e li salva così modificati in un database dedicato.

- ▶ Quanto tempo, a stima, può essere necessario per realizzarlo?
- ▶ Ora, per quanto tempo questa applicazione potrebbe durare nel tempo?
- ▶ Secondo voi, quante modifiche saranno necessarie in questo periodo di tempo?

Qual è l'unica cosa sicura nella vita di un software?

IL CAMBIAMENTO

Motivazioni al cambiamento

Cambiamenti dovuti a

- ▶ Nuove richieste da parte del cliente
- ▶ Malfunzionamenti (più o meno gravi)



Per quanto stabili, consolidati, sicuri ecc. siano i requisiti iniziali

La realizzazione «ottimale» di un sistema dovrà perciò tenere in considerazione che tutto potrà essere soggetto ad un cambiamento futuro e che, a seconda di come si realizzerà il software, questo cambiamento potrà risultare relativamente semplice o estremamente complesso.

Obiettivo



QUALITÀ

Qualità del software

- ▶ Cos'è la qualità di un software?

*«Insieme di **caratteristiche** che ne influenzano e determinano l'uso e la manutenzione»*

Qualità del software - 2

- ▶ Identificabili in due gruppi
 - ▶ **Esterne**, ovvero apprezzabili da un utilizzatore finale
 - ▶ **Interne**, misurabili e valutabili da parte di un tecnico, non visibili da parte di un utilizzatore finale ma influenzano il risultato complessivo del sistema (e le caratteristiche esterne)

Caratteristiche del software

Dunque, quali sono queste caratteristiche?

- ▶ Correttezza
- ▶ Efficienza
- ▶ Robustezza
- ▶ Affidabilità
- ▶ Usabilità
- ▶ Sicurezza
- ▶ Costo
- ▶ Estendibilità
- ▶ Riusabilità
- ▶ Strutturazione
- ▶ Leggibilità
- ▶ Manutenibilità
- ▶ Modificabilità
- ▶ Portabilità
- ▶ Compatibilità

Caratteristiche del software - 2

Dire che un software deve avere queste caratteristiche è sufficiente per sapere come realizzarlo?

- | | | | |
|-----------------|--------------------------|------------------|---------------------------|
| ▶ Correttezza | di cosa? | ▶ Riusabilità | rispetto quali necessità? |
| ▶ Efficienza | di cosa? | ▶ Strutturazione | di cosa? |
| ▶ Robustezza | di cosa? | ▶ Leggibilità | di cosa? |
| ▶ Affidabilità | di cosa? | ▶ Manutenibilità | cosa significa? |
| ▶ Usabilità | per chi? Come la valuto? | ▶ Modificabilità | in che termini? |
| ▶ Sicurezza | a che livello? | ▶ Portabilità | dove? |
| ▶ Costo | di cosa? | ▶ Compatibilità | di cosa? |
| ▶ Estendibilità | rispetto a cosa? | | |

In generale senza una definizione, queste caratteristiche non hanno significato e dunque non sono applicabili.

Caratteristiche del software - 3

- ▶ **Correttezza:** capacità del software di eseguire correttamente i suoi compiti (secondo le relative specifiche)
- ▶ **Efficienza:** capacità del software di utilizzare in modo economico e razionale le risorse di calcolo (nello spazio e nel tempo)
- ▶ **Robustezza:** capacità del software di funzionare anche in condizioni anomale
- ▶ **Affidabilità:** capacità del software di presentare rari guasti
- ▶ **Usabilità:** il software presenta facilità di apprendimento e di utilizzo per l'utente
- ▶ **Sicurezza:** capacità del software di non consentire utilizzi non autorizzati
- ▶ **Costo:** il software presenta un adeguato costo (tempi di sviluppo, risorse) rispetto ai benefici
- ▶ **Estendibilità:** il software presenta facilità ad essere adattato a cambiamenti (nelle specifiche)

Caratteristiche del software - 4

- ▶ **Riusabilità:** possibilità di utilizzare il software (in tutto o in parte) per diverse applicazioni
- ▶ **Strutturazione:** grado di organizzazione interna in parti con funzioni specifiche ed interagenti
- ▶ **Leggibilità:** capacità del programma di presentare esplicitamente le scelte fatte dal progettista
- ▶ **Manutenibilità:** il software presenta facilità di gestione dei problemi durante la fase di utilizzo
- ▶ **Modificabilità:** il software presenta facilità di modifica a seguito della scoperta di un errore o a causa di una variazione delle necessità applicative
- ▶ **Portabilità:** il software presenta facilità nel trasferimento in ambienti hardware e software di base diversi
- ▶ **Compatibilità:** facilità di combinare fra loro differenti prodotti software (necessità di standard per protocolli, interfacce, strutture dati standard)

Caratteristiche del software - 5

- ▶ Le caratteristiche sono tutte compatibili tra loro?

Compromessi della qualità

Non è possibile massimizzare sempre tutti le caratteristiche nella realizzazione di un software

Ad esempio

- ▶ Efficienza ↔ Portabilità
- ▶ Sicurezza ↔ Usabilità

Una caratteristica è sicuramente quella che più influenza la completezza delle altre...

Il costo

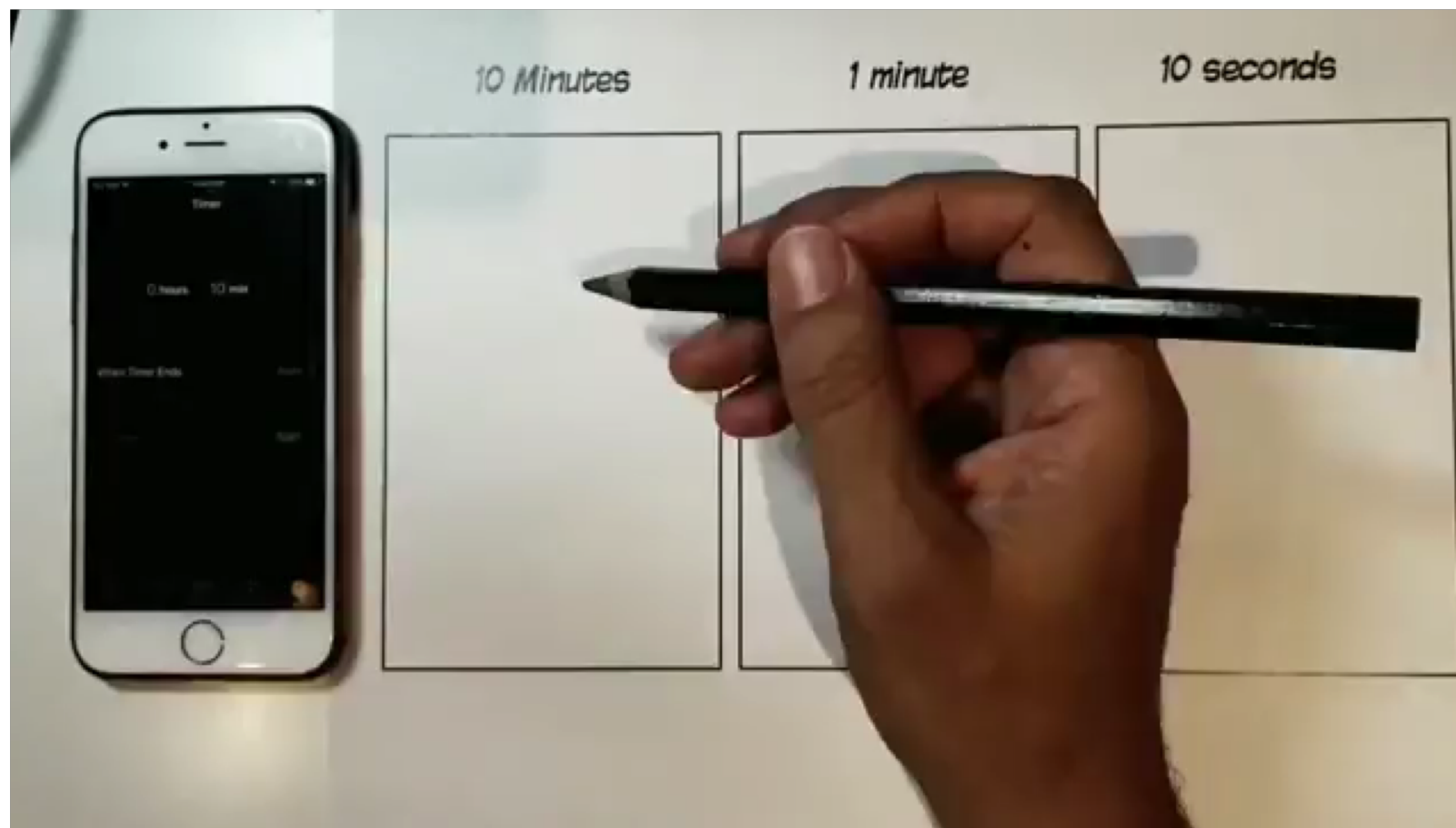
- ▶ Il costo di un software è ciò che maggiormente influisce con la sua qualità di realizzazione. Più alto sarà il budget a disposizione, più (teoricamente) ci sarà la possibilità di incrementare l'attenzione alle caratteristiche positive dello stesso
- ▶ Si suddividono in
 - ▶ Costi diretti
ovvero i costi da sostenere per la realizzazione del software
 - ▶ Costi indiretti
ovvero i costi da sostenere in caso di malfunzionamenti
- ▶ Aumentare la qualità (e di conseguenza i costi diretti) ha come effetto quello di diminuire i malfunzionamenti (e i costi indiretti)

Compromessi della qualità - 2

C'è un altro aspetto che influenza la qualità del software

► Qual è?

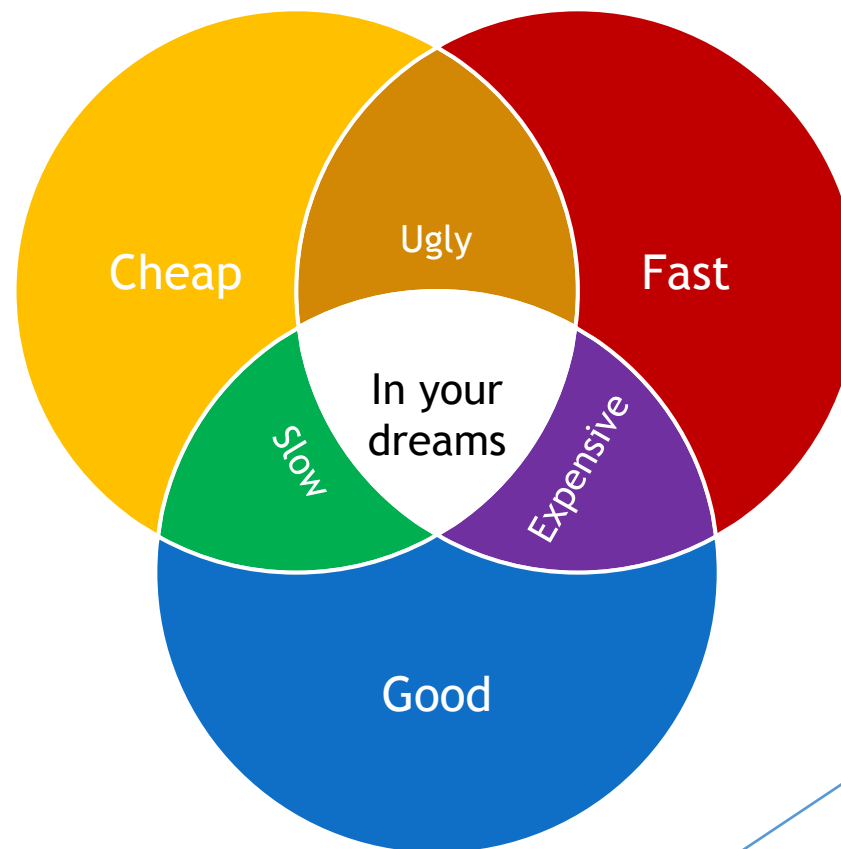
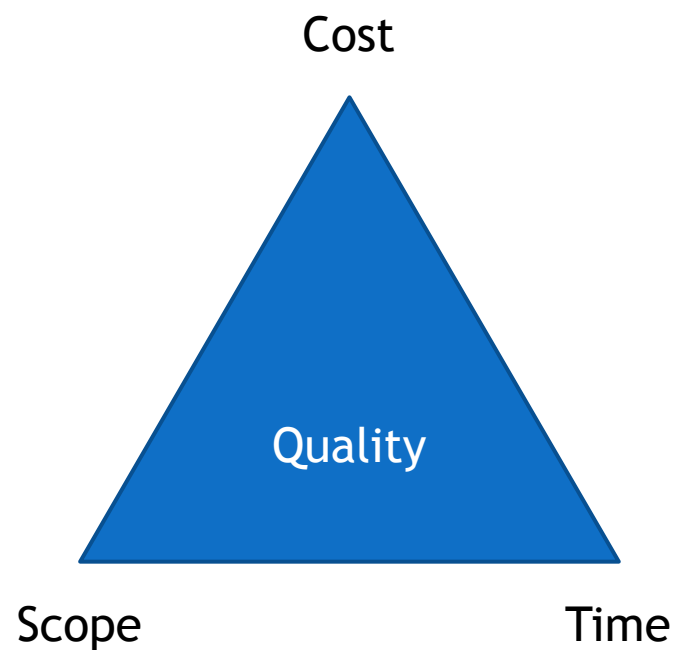
Il tempo



Vincoli di un progetto

- ▶ La creazione di un software che ha come obiettivo la qualità deve tenere conto che
 - ▶ Il progetto deve essere completato entro un determinato costo (budget)
 - ▶ Il progetto deve essere completato entro un determinato tempo (duration)
 - ▶ Il progetto deve soddisfare determinati requisiti, né più né meno (scope)
 - ▶ Il progetto deve soddisfare le aspettative del cliente (requirements)

Vincoli di un progetto - 2



Il costo degli errori

- ▶ L'esito finale della realizzazione di un software è la soddisfazione dei bisogni del cliente
- ▶ Se l'utente non è soddisfatto significa che sono stati commessi degli errori durante la realizzazione del sistema
- ▶ Sulla base del ciclo di realizzazione, quali sono le fasi in cui un errore pesa di più?
- ▶ In generale, quanto più un errore avviene nelle prime fasi del progetto, tanto più tardi sarà intercettato e tanto più costosa sarà la sua correzione

Pianificazione

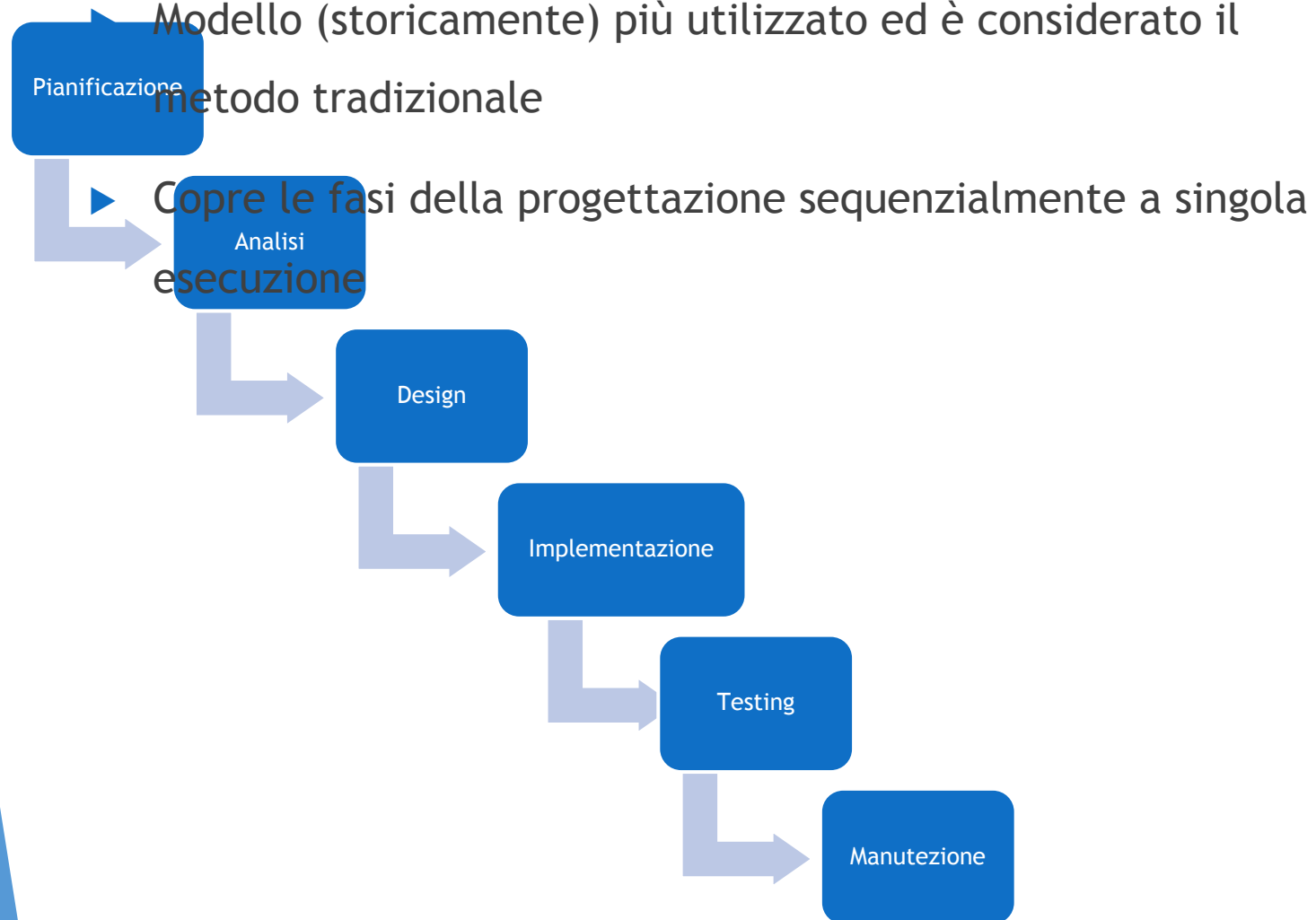
Modelli utilizzati e strumenti a supporto

Modelli

- ▶ Evoluzione nel tempo della stessa gestione del progetto, dovuta principalmente
 - ▶ Al cambio delle esigenze dei clienti che vogliono avere un riscontro più frequente dell'avanzamento del progetto
 - ▶ Al cambio di focus da *whole product* a *minimum viable product*

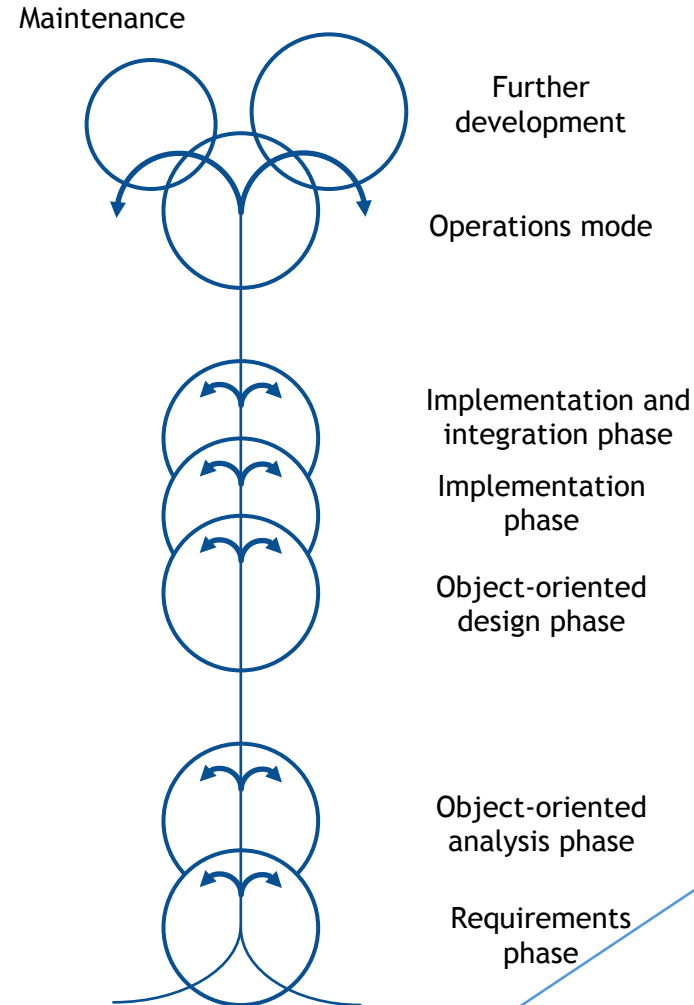
Modello a cascata

Modello (storicamente) più utilizzato ed è considerato il metodo tradizionale



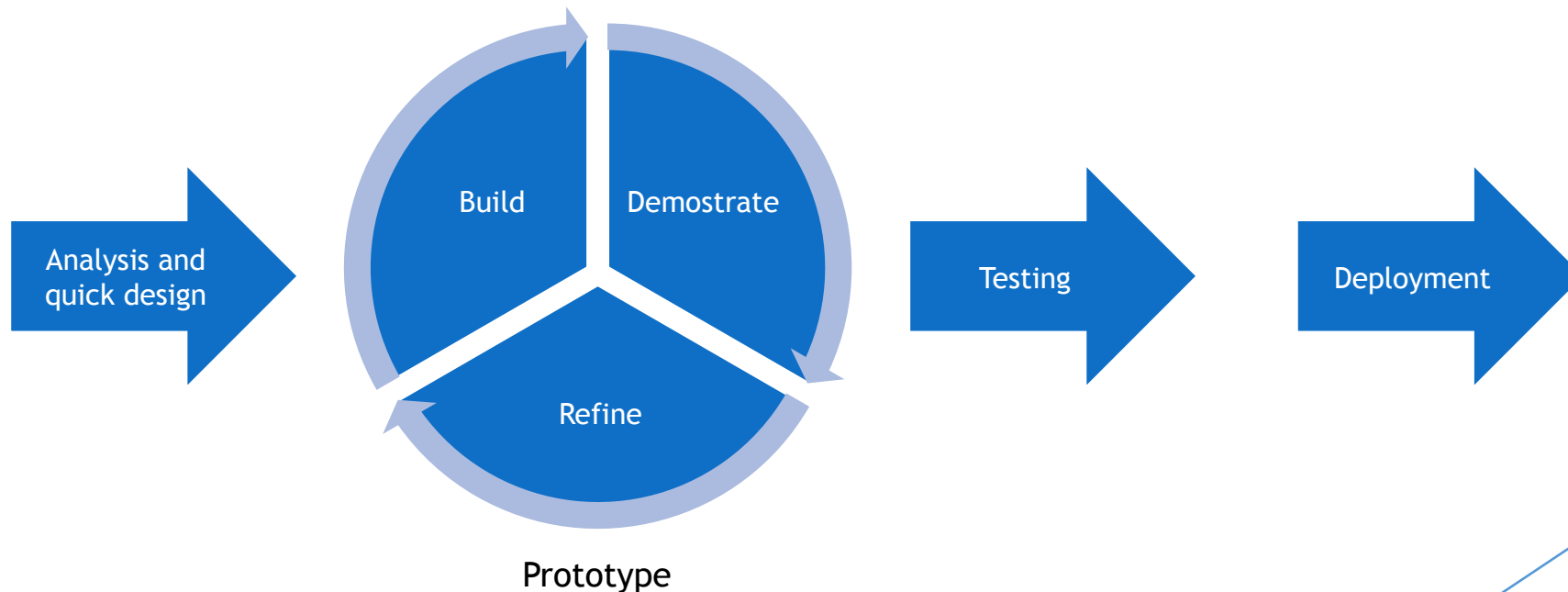
Modello a fontana

- ▶ Evoluzione del modello a cascata che consente una riesecuzione dei vari step, ottimizzando di tempi
- ▶ Adatto a software *mission critical* dove gli errori non sono tollerabili (controllo aereo)
- ▶ Adatto allo sviluppo di applicazioni con linguaggi object-oriented
- ▶ Ottimizza la parallelizzazione delle attività



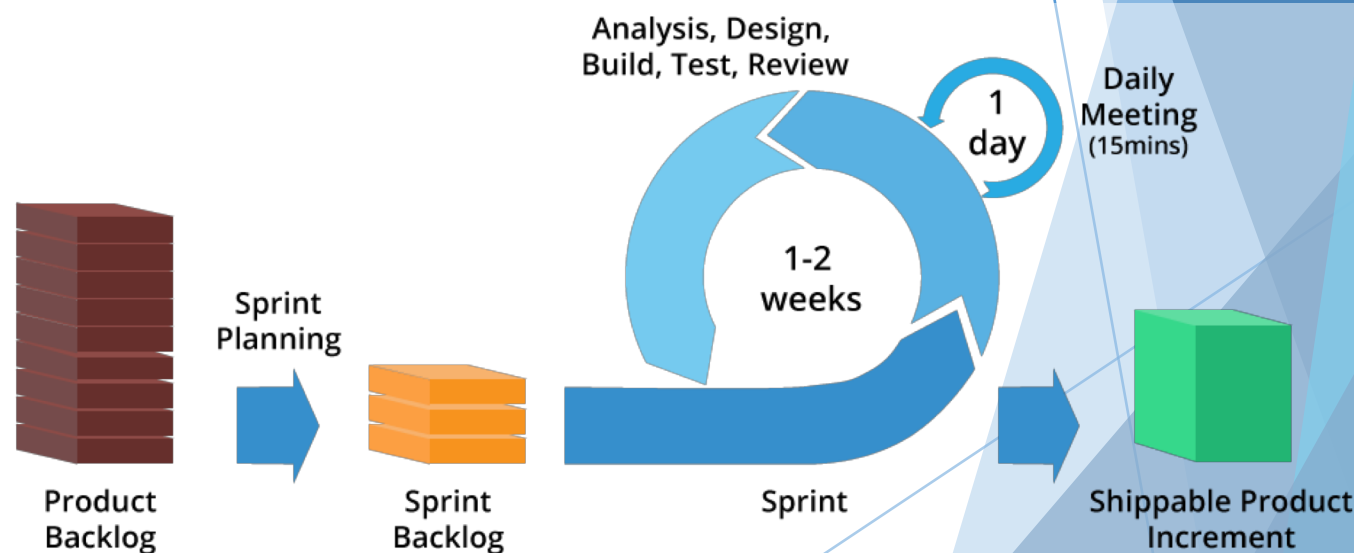
Modello prototipale

- ▶ Denominato ufficialmente Rapid Application Development (RAD)
- ▶ Basato ripetizione iterativa di design e implementazione per la realizzazione di prototipi via via più evoluti e vicini al prodotto finale



Metodologia agile

- ▶ Basato concettualmente su multiple iterazioni del modello a cascata che determinano porzioni di software **completo**
- ▶ Si fonda sulla stretta collaborazione fra fornitore e cliente ad ogni ciclo di sviluppo
- ▶ Iterazioni brevi, spesso di una o due settimane (sprint)
- ▶ Quasi sempre sfrutta strumenti di supporto che automatizzano gli step di build e test, nonché di raccolta delle informazioni relative a task e *issue* evidenziate



Fonte: www.bytelion.com/scrumb-for-business/agile_development_process/

DevOps

- ▶ Evoluzione della metodologia agile per coprire tutte le fasi dello sviluppo fino alla pubblicazione del software
- ▶ Ha come obiettivo la messa in produzione incrementale del sistema via via che le singole parti costituenti vengono completate e testate
- ▶ Necessita di una collaborazione stretta fra le risorse che si occupano dello sviluppo (dev) e chi si occupa dell'infrastruttura (ops - operations)

Waterfall



Agile



DevOps

