03/10/2022: Introduzione: metodo e concetti chiave di sviluppo del software

Cose utili:

* Tullio usa solo il Moodle e mette il materiale sul Moodle o sul calendario stesso (tende ad usare siti esterni, comunque accessibili dal Moodle stesso)
* Legenda del calendario:
  + T 🡪 Teoria
  + P 🡪 Pratica
  + PD 🡪 Progetto Didattico
  + E 🡪 Esercitazione
* Glossario (termini specifici del dominio di interesse) delle parole chiave:
  + Protocollo: in informatica, un insieme di regole o procedure per la trasmissione di dati tra dispositivi elettronici, come i computer. Affinché i computer possano scambiarsi informazioni, deve esistere un accordo preesistente su come le informazioni saranno strutturate e su come ciascuna parte le invierà e le riceverà.
  + Progetto (software) 🡪 Visto (esaminato) da chi lo crea e da parte di chi ci ha a che fare
    - Sviluppo
    - Professionale
    - Valutazione qualitativa
    - Collaborativo
    - Monitoraggio

Esso deve essere un mezzo e non un “obiettivo” (goal/aim)

Quest’ultimo, infatti, va compreso; infatti, è visibile a pochi da subito ma l’obiettivo deve essere chiaro a tutti. Non si intende programmazione, quanto piuttosto la *realizzazione di obiettivi*.

* + Sviluppo: Percorso di realizzazione tra obiettivi *detti* ed obiettivi *raggiunti* per il tramite del software. Essi rispondono principalmente a dei *bisogni (needs)*, essendo essi stessi il *mezzo*.

Si intende tutto ciò attraverso il percorso di progettazione (*design*), di cui il progetto è un prodotto. L’insieme delle regole che lo realizzano è il *paradigma*, per mezzo degli informatici che realizzano software.

* + - Es. Zoom 🡪 Architettura client (inoltrano richieste) – server (soddisfano le richieste presenti) non bastevole; infatti, tutte le richieste sono soddisfatte da un server centrale (senza che l’utente sappia nulla di tutto ciò). Esso è uno strumento

web-based (quindi, una risposta ad un bisogno) 🡪 risultando *portatile*.

Essa è una *piattaforma* (un aggregato che, per esistono, raduna tanti main(), idealmente compiendo tante cose insieme).

Si deriva dalla concezione del Web 2.0 (utenti che creano contenuti).

* + - Per realizzare in maniera coesa (interconnessa, espressa come unica entità) le azioni di tanti programmi distinti.
    - Il ruolo dell’informatico è distinguere i bisogni tecnici e i bisogni utente creando un prodotto di *qualità* (senza seguire una chiara idea di realizzazione (protocollo), scomodo da usare, poco accessibile, ecc.), risultando dunque professionale. Si cerca di raggiungere lo stato dell’arte (inteso come la qualità in un senso personale) attraverso un costante *autoapprendimento*. Colui che ha gli strumenti e crea è il *practitioner*, continuando ad imparare in autonomia.

04/10/2022: Introduzione (slide omonima): obiettivi, metodo, concetti chiave

(Eventuali approfondimenti:

<http://worrydream.com/refs/Brooks-NoSilverBullet.pdf>

<https://www.acm.org/binaries/content/assets/membership/images2/fac-stu-poster-code.pdf>)

Apprendiamo metodi e pratiche di lavoro alla base della professione informatica.

* Gestire il tempo 🡪 Risorsa scarsa con numerose scadenze
  + Disponibilità, scadenze, conflitti, priorità
* Collaborare 🡪 I compiti da svolgere richiedono molto tempo per un singolo individuo
  + Fissare obiettivi, dividersi compiti, verificare progressi, riportare difficoltà
* Assumersi responsabilità 🡪 Incarico per garantire una partecipazione equa, di qualità e rispetto a chi ha assegnato un compito
  + Fare quanto pattuito, agire al meglio delle proprie capacità, auto-valutarsi prima di valutare
  + Usare strumenti allo stato dell’arte (sia consolidato in un ambito che una cosa attuale)
* Auto-apprendere 🡪 Cercare di “stare al passo” con le tecnologie e con le competenze richieste (avvicinandosi allo stato dell’arte)
  + “Imparare a imparare”, essenziale competenza trasversale

L’obiettivo è cercare di:

* trasmettere dei concetti su cui si elabora da chi le usa e chi le insegna attivamente
* per avvicinarci a modo di lavorare (way of working) professionale, «operante allo stato dell’arte», per conoscenze tecnologiche e metodologiche

La conoscenza passa dalla comprensione profonda, sperimentata, dei significati

* Non ricordare, ma riconoscere (so chi sei …)

Vogliamo fissare tali conoscenze in un glossario

* Raccolta di termini/concetti centrali al dominio SWE
* Registrati in modo da facilitarne la localizzazione
* Corredati dalla nostra personale specifica del loro significato e ogni altra informazione utile a riconoscerli

Vogliamo raffinarne costantemente la comprensione

* Legando la teoria (quanto ascoltato) con la pratica (quanto riscontrato)

Tramite un progetto didattico collaborativo (a scopo di learning, rispetto ad un futuro professionale)

* Promosso da un proponente esterno
* Con esigenze e obiettivi funzionali innovativi
* Complesso, impegnativo, visionario
* Tecnologicamente avanzato

Occorre fare attenzione alle norme e regole, per *non inabissarsi e sparire dai radar* (modalità sottomarino)

In generale (definizione di Harold Kerzner, autore di “Project Management” ed esperto mondiale in materia)

Progetto: Insieme di attività che

* + - Devono raggiungere determinati obiettivi a partire da determinate specifiche (rispondendo ai *needs-bisogni*)
    - Hanno una data d’inizio e una data di fine fissate
    - Dispongono di risorse limitate (persone, tempo, denaro, strumenti)
    - Consumano tali risorse nel loro svolgersi

L’uscita di un progetto è un prodotto composito

* + SW sorgente/eseguibile, librerie, documenti, manuali

La vera definizione in ambito universitario è *assignment*, quindi un compito specifico assegnato e gestito in maniera flessibile, sia di chi impone la scadenza, sia di chi lo esegue.

I costituenti di un progetto sono:

* Pianificazione 🡪 Studiare la natura del problema e capire come impostare le attività sulla base della domanda
  + Gestire risorse (persone, tempo, denaro, strumenti) in modo responsabile, in funzione degli obiettivi
* Analisi dei requisiti 🡪 I needs sono tradotti in prodotti informatici (non sapendo come fare, si chiede e dialoga con una serie di interlocutori)
  + Definire cosa bisogna fare
* Progettazione (design) 🡪 Trasformare l’idea di comprensione (profonda) del problema in modo attuabile e con una soluzione sensata ed accettabile. Qui non si ha una sola soluzione, ma ce ne possono essere tante
  + Definire come farlo
* Realizzazione (implementation) 🡪 Creare un utensile (prodotto è usabile; quindi, una volta creato compie la sua funzione), quindi creare qualcosa che serve effettivamente
  + Farlo, perseguendo qualità
  + Accertando l’assenza di errori od omissioni
  + Accertando che i risultati soddisfino le attese

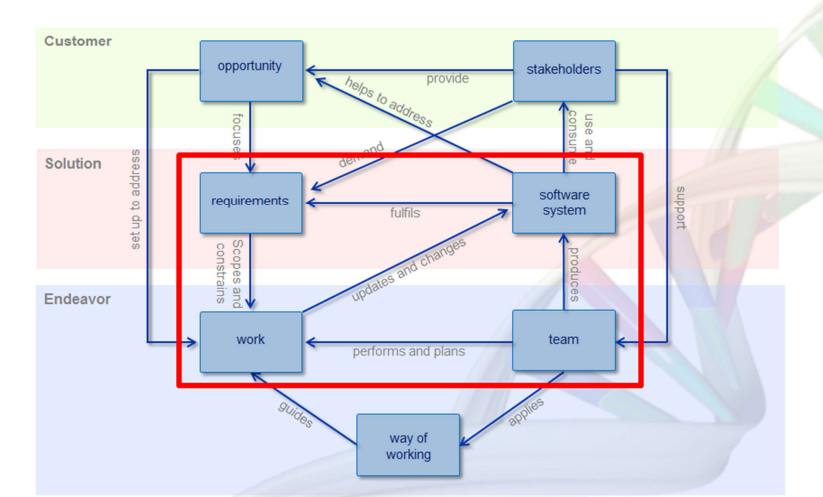


Figura 1 - Proveniente da https://semat.org/

Stato dell’arte, adattato al proprio livello sulla base delle regole presenti

Collassare design ed implementazione, esempio scrivendo codice, è tipico di chi progetta male.

Prima si pensa bene a cosa fare e l’implementazione risulta essere conseguenza (non premessa) del design.

In tal modo, mi assicuro di soddisfare le aspettative di chi lo richiede, chiedendo ed interagendo spesso.

Cosa non è un progetto:

* Si è accecati dalla fondamentale l'inutilità di fondo dei loro prodotti, dal senso di successo che si prova nel farli funzionare (il fatto che compili ci toglie dall’esaminare ulteriormente un prodotto)
* In altre parole, i loro difetti fondamentali di progettazione sono completamente nascosti dai loro difetti di design difetti superficiali di progettazione (errori di sintassi, segmentation fault, ecc.)

Esso dovrebbe essere l’espressione di realizzare sulla base di regole un prodotto preciso e professionale, valorizzando l’esperienza e creandole al meglio delle regole esistenti.