

Appunti di Progettazione del Software

Canale 1 · Prof. Paolo Liberatore

Capitolo 1: Il software gestionale

Autore: Tommaso C.

Appunti tratti dalle lezioni in aula e dal materiale del corso

Argomenti: Software gestionale · Attori · Ciclo di vita iterativo
Dimensione e durata · Documentazione e standard · Qualità e contesto

Indice

1	Il software gestionale	1
2	Gli attori della progettazione	1
3	Il ciclo di vita del software	2
4	Dimensione, durata e documentazione	3
5	Le qualità del software secondo il contesto	4
6	Perché la programmazione a oggetti	5
7	Esercizi	5
8	Soluzioni	6

1 Il software gestionale

Definizione – Software gestionale

Un **software gestionale** è un programma realizzato per un ente o un'azienda (un comune, un ministero, una banca, un'università) che tratta un certo insieme di **dati** e i **flussi di informazioni** legati alle attività dell'organizzazione.

Il discorso sulla progettazione riguarda soprattutto questo tipo di software. Restano fuori, per esempio, i videogiochi, le grandi applicazioni di rete come Google o Amazon e i programmi per sistemi embedded come quelli per Arduino: per loro valgono solo alcuni temi, e il peso delle qualità cambia (Sezione 5).

È il dominio che richiede la maggiore quantità di software, e quindi di programmatori e progettisti. Ogni ente di dimensioni almeno medie ha caratteristiche specifiche e vuole il proprio software, fatto o adattato apposta; una ditta piccola adotta più spesso programmi preconfezionati, per esempio per il personale o la contabilità.

Osservazione – Il software da zero è raro

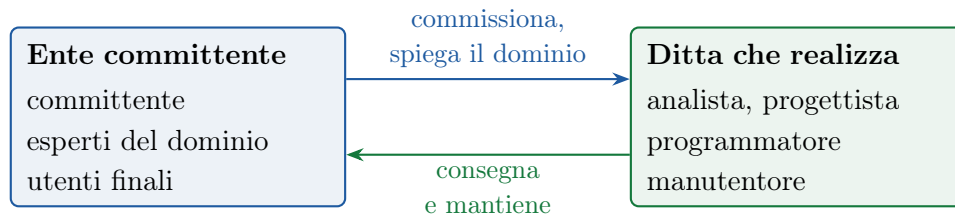
Il lavoro tipico non è scrivere un programma da zero, ma entrare in un software già sviluppato: aggiungere parti nuove, modificare quelle esistenti, farne la manutenzione.

2 Gli attori della progettazione

Definizione – Attori

- **Committente:** l'ente o l'azienda che vuole il software.
- **Esperti del dominio:** impiegati dell'ente committente (o professionisti designati) che conoscono l'attività da informatizzare, le funzionalità e i dati usati.
- **Analista, progettista, programmatore:** personale della ditta che realizza il software; analizza i requisiti e la loro fattibilità, progetta e scrive il programma.
- **Utente finale:** chi usa il programma.
- **Manutentore:** chi interviene sul software dopo la consegna per correggerlo e aggiornarlo.

L'ente si rivolge a volte a un singolo programmatore, più spesso a una ditta che produce software. Un caso completo è nell'Esercizio 1.



I due lati del contesto organizzativo.

3 Il ciclo di vita del software

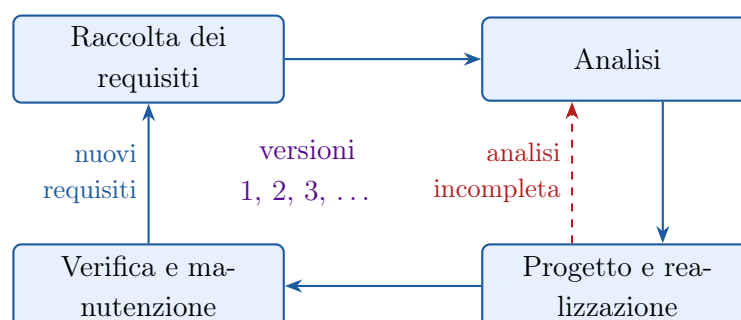
Definizione – Ciclo di vita

Il **ciclo di vita** del software è la successione delle fasi del suo sviluppo:

1. **studio di fattibilità e raccolta dei requisiti:** costi e benefici, pianificazione, ambiente di programmazione, requisiti;
2. **analisi dei requisiti:** *che cosa* l'applicazione deve fare; si descrive il dominio con lo *schema concettuale*;
3. **progetto e realizzazione:** *come* l'applicazione realizza le sue funzioni; architettura, strutture di rappresentazione, codice e documentazione;
4. **verifica:** il programma svolge correttamente, completamente ed efficientemente il suo compito?
5. **manutenzione:** controllo del programma in esercizio, correzione degli errori e adattamento alle modifiche richieste.

Il processo non è una sequenza ma è quasi **ciclico**: entrano continuamente requisiti nuovi, che vengono analizzati, realizzati e verificati, e ogni giro produce una nuova versione del programma.

Dentro il ciclo ci sono continui **passi all'indietro**: durante il progetto si può scoprire che l'analisi è incompleta, o che alcuni requisiti raccolti erano ambigui; allora si torna all'analisi e si riparte da lì.



Il ciclo di vita come processo ciclico, con i ritorni alle fasi precedenti (tratteggio).

Esempio – Il sistema universitario degli esami

Nelle prime versioni del sistema con cui gli studenti si prenotano agli esami c'erano solo le funzionalità di base: senza il controllo dei piani di studio, uno studente poteva prenotarsi a qualsiasi esame, anche di un altro corso di laurea. I piani di studio e gli altri controlli sono stati aggiunti in seguito. Anche la verifica è continua: alcuni errori sono emersi alla prima prova d'uso, altri a distanza di anni.

4 Dimensione, durata e documentazione

Il software gestionale si distingue da un esercizio di programmazione per due caratteristiche, entrambe visibili nel sistema degli esami.

- **Dimensione.** Il sistema deve coprire studenti, docenti, segreterie didattiche e altro: una persona da sola ci metterebbe anni. Servono uno o più gruppi di programmatori, eppure è un programma di media grandezza.
- **Durata nel tempo.** È in uso da almeno quindici anni e viene aggiornato di continuo. Un programma che serve per poco, per esempio a convertire i dati nel passaggio da una piattaforma a un'altra, non ha quasi manutenzione: il discorso sulla progettazione riguarda i primi.

Dimensione e durata hanno due conseguenze.

1. **Più programmatori, anche in team diversi**, scrivono parti diverse dello stesso programma, che devono interagire: la **comunicazione** tra loro è fondamentale.
2. **I programmatori cambiano.** Non restano a lungo nella stessa ditta, o vengono spostati su altri progetti. Se una parte va corretta cinque anni dopo, lo farà quasi certamente un'altra persona; e anche chi l'ha scritta faticherebbe a ricordare che cosa aveva fatto.

Osservazione – Semplice, documentato, standardizzato

Per questo il software deve essere:

- **semplice e ben documentato**, così che si trovi subito la parte che svolge una certa funzione;
- **standardizzato**: si adottano il meno possibile soluzioni originali e non si reinventa quello che esiste già e funziona.

La regola non serve a legare le mani al programmatore. Chi deve correggere o modificare una parte del software, o collegarci un'altra parte, deve sapere che cosa aspettarsi, senza interpretare ogni volta quello che era stato fatto prima. In un codice non documentato e

“a spaghetti”, in cui non si capisce dove porti il flusso, si passano ore solo a capire come funziona.

5 Le qualità del software secondo il contesto

Definizione – Qualità esterne e interne

Le **qualità esterne** sono visibili agli utenti del sistema, anche non specialisti, senza ispezionare il codice. Le **qualità interne** riguardano gli sviluppatori: le valutano gli specialisti e richiedono di conoscere la struttura del programma. La distinzione non è sempre netta.

Le qualità usate in questo capitolo, nei due gruppi:

- esterne:
 - **correttezza**: il software fa quello per cui è stato progettato;
 - **affidabilità**: ci si può fidare delle sue funzionalità; se i vincoli sono rispettati, i risultati sono disponibili;
 - **robustezza**: il comportamento resta accettabile anche in situazioni non previste dalla specifica dei requisiti;
 - **usabilità**: la facilità d'uso da parte dell'utente;
- interne:
 - **efficienza**: il “peso” del software sulle risorse del sistema, cioè tempo di esecuzione e memoria;
 - **modularità**: il grado di organizzazione del software in parti ben specificate che interagiscono.

Quali qualità contino dipende dal tipo di software e dal contesto in cui si usa.

Nel gestionale contano correttezza e affidabilità. Un'applicazione bancaria non corretta potrebbe trasformare il saldo di un cliente da 50 000 euro in 50 milioni. L'efficienza conta meno: se un'operazione richiede due secondi, l'impiegato aspetta. Contano anche la modularità e la leggibilità del codice.

Altrove conta l'efficienza. In un videogioco la riproduzione audio e video deve restare fluida. Chi entra in un sito di e-commerce e deve aspettare due secondi passa a un altro sito.

L'usabilità dipende da chi usa il software. Un impiegato che deve per forza usare un certo gestionale finisce per farsi uno schema delle operazioni. Quando invece l'utente

può scegliere, la semplicità dell'interfaccia può decidere il successo: Google si è imposto sui primi motori di ricerca con una pagina che mostrava il solo campo della parola chiave. Nella centralina di un'automobile, che non interagisce con chi guida, l'usabilità non è nemmeno applicabile.

6 Perché la programmazione a oggetti

La programmazione a oggetti favorisce le qualità interne del software, ed è per questo molto usata nelle applicazioni gestionali. I suoi vantaggi diventano evidenti quando i programmi sono **molto grandi**: lì contano l'ereditarietà e la **modularizzazione** fatta attraverso le classi.

Riepilogo

- **Software gestionale**: programma per un ente che tratta dati e flussi di informazioni; di rado si scrive da zero.
- **Attori**: committente, esperti del dominio, analista, progettista e programmatore, utente finale, manutentore.
- **Ciclo di vita**: requisiti, analisi (che cosa), progetto (come), verifica, manutenzione; in pratica ciclico, con passi all'indietro.
- **Dimensione e durata** impongono comunicazione fra programmatori e programmatori che cambiano.
- Il software dev'essere **semplice, documentato e standardizzato**.
- **Qualità esterne** per gli utenti, **interne** per gli sviluppatori; quali contano dipende dal contesto.
- Nel gestionale contano correttezza e affidabilità più dell'efficienza.
- La programmazione a oggetti favorisce le qualità interne nei programmi grandi.

7 Esercizi

Esercizio 1 – Gli attori di un sistema per le contravvenzioni

Il Comune di XYZ intende automatizzare la gestione delle informazioni relative alle contravvenzioni elevate sul suo territorio. Vuole dotare ogni vigile di un dispositivo palmare con cui comunicare al sistema informatico il veicolo multato, il luogo e la natura dell'infrazione; il sistema notificherà la contravvenzione al cittadino per posta ordinaria. Il Comune bandisce una gara per la realizzazione e la manutenzione del sistema, vinta dalla Ditta ABC. Quali sono gli attori coinvolti?

Soluzione a pagina 6 →

8 Soluzioni

Soluzione dell'Esercizio 1 – Gli attori di un sistema per le contravvenzioni

- Committente: il Comune di XYZ.
- Esperto del dominio: un funzionario del Comune, o un altro professionista designato, esperto del Codice della Strada.
- Utenti finali: i vigili.
- Analista, progettista, programmatore, manutentore: personale della Ditta ABC.

[*← Torna all'Esercizio 1, pagina 5*](#)