

### 1.2 Il ciclo di sviluppo dei programmi

L'obiettivo di questo paragrafo è l'illustrazione delle diverse fasi che compongono il processo di sviluppo dei programmi. La trattazione ha lo scopo di inquadrare la fase di progettazione, e viene quindi condotta a livello generale, senza pretese di completezza.

Il ciclo di sviluppo di un programma viene tradizionalmente visto come composto dalle seguenti fasi:

1. Studio di fattibilità
2. Raccolta e analisi dei requisiti
3. Progettazione / concettuale  
\ realizzativa
4. Verifica
5. Manutenzione

La fase di *studio di fattibilità* ha lo scopo di effettuare una valutazione dei costi e dei benefici del sistema che si intende costruire, con l'obiettivo di stabilire se e quando il progetto deve iniziare, di analizzare quali siano le alternative possibili per lo sviluppo e la realizzazione del sistema, e di individuare le risorse necessarie per l'attuazione del progetto.

La fase di *raccolta e analisi dei requisiti* ha lo scopo di definire con precisione il problema da risolvere e, a fronte delle caratteristiche individuate, di specificare l'ambiente di sviluppo e l'ambiente di realizzazione, sia hardware (sistema di elaborazione, eventuale rete di comunicazione ecc.) che software (linguaggio di programmazione, ambiente di programmazione ecc.). Il risultato di questa fase è, in genere, un documento, detto *documento di analisi*, che specifica tutti gli aspetti sopra menzionati. Il livello di precisione e di formalità di tale documento varia da caso a caso: in un grosso progetto a cui partecipano diversi progettisti e programmatori può essere un documento formale, redatto secondo criteri prestabiliti, mentre in un progetto di piccole dimensioni può essere un testo che specifica informalmente il problema in esame.

Il documento di analisi si compone in genere di due parti:

- Una parte che specifica l'ambiente hardware e software scelto per l'applicazione, e descrive i requisiti raccolti, illustrando, spesso discorsivamente, le caratteristiche del problema elaborativo da risolvere.
- Una parte che specifica, mediante formalismi ad hoc, i diversi aspetti dell'applicazione da realizzare. In genere, gli aspetti che si prendono in considerazione sono:
  - La struttura concettuale dei dati che il sistema deve manipolare.
  - Le caratteristiche generali delle operazioni che il sistema deve effettuare, e le dipendenze che si creano tra esse in virtù dei flussi di informazione che si devono garantire nel sistema.

- L'evoluzione del sistema complessivo in termini di flusso di esecuzione delle varie operazioni al fine del raggiungimento degli obiettivi elaborativi del sistema.

Non è nostro scopo l'approfondimento dei metodi, dei formalismi, e degli strumenti che si utilizzano nella fase di raccolta e analisi dei requisiti. Vogliamo solo sottolineare che il documento di analisi è un prodotto importante nel ciclo di sviluppo, sia perché costituisce il punto di riferimento per tutte le fasi seguenti, sia perché rappresenta il principale input alla fase di progettazione.

La fase di *progettazione* ha lo scopo di concepire la soluzione informatica del problema, di definire l'architettura del programma, di individuare e specificare l'organizzazione del programma in parti ben definite ed interagenti, di specificare le funzioni delle componenti individuate, di scegliere le strutture di rappresentazione degli oggetti da manipolare, e di redigere il programma secondo il linguaggio e l'ambiente di programmazione scelto. Il risultato di questa fase è il programma vero e proprio, spesso corredato da altri documenti (integrati o meno nel testo del programma) che descrivono sia il processo che ha portato al programma, sia il programma stesso. I principi e le tecniche che sono alla base di questa fase costituiscono l'oggetto di indagine di questo testo.

La fase di *verifica* ha lo scopo di analizzare tutta la documentazione prodotta nelle fasi di analisi e di progettazione allo scopo di verificare che il programma prodotto svolga correttamente, completamente, ed efficientemente il compito per il quale è stato sviluppato. La verifica viene condotta sia mediante analisi del testo del programma, sia mediante una serie di prove su dati di test. A seconda del risultato di questa fase, sarà ovviamente possibile tornare a svolgere attività proprie della fase di progetto, ad esempio per modificare scelte dimostrate erranee o non completamente adeguate, o per correggere possibili errori di varia natura.

La fase di *manutenzione* ha lo scopo da una parte di controllare che il programma, durante il periodo di esercizio, produca in ogni situazione i risultati attesi secondo le specifiche, e dall'altra di correggere eventuali errori e disfunzioni, e aggiornare il programma a fronte di eventuali cambiamenti nelle specifiche. Anche in questa fase sarà ovviamente possibile un ritorno ad attività della fase di progetto.

Il ciclo di sviluppo che abbiamo appena descritto viene spesso chiamato *a cascata*, per sottolineare che la struttura del processo si compone di fasi che, almeno a livello schematico, dovrebbero essere eseguite sequenzialmente. Osserviamo, però, che esso rappresenta più un modello di riferimento che una vera metodologia rigida e prescrittiva. Nella realtà, è infatti opportuno condurre il processo di sviluppo in modo flessibile, per garantire che il prodotto finale sia effettivamente costruito secondo le esigenze. A tale scopo è spesso opportuno prevedere che, durante la fase di progetto, o almeno subito dopo, si costruisca un prototipo del programma, cioè una versione che, pur non avendo tutte le caratteristiche di completezza della soluzione finale, evidenzii le scelte più critiche e prefiguri il comportamento del prodotto quando sarà nella fase di esercizio. È evidente che, una volta prodotto il prototipo, la fase di verifica potrà svolgersi più efficacemente interagendo con l'utente finale, che avrà a disposizione elementi concreti per giudicare la bontà delle scelte effettuate. Tenendo conto di questo ulteriore elemento, il processo di sviluppo si comporrà di una successione di rilasci di versioni sempre più accurate, complete, e rispondenti alle esigenze dell'utente. In questo modo, la suddivisione nelle fasi di progetto, verifica e manutenzione è meno netta di quella rappresentata dal modello a cascata. Per una descrizione più dettagliata di altri modelli di descrizione del ciclo di vita del software rimandiamo a [22].

---

## Capitolo 2

### La fase di progettazione

La fase di progettazione è la più critica di tutto il ciclo di sviluppo di un programma, e lo scopo principale di questo testo è l'approfondimento delle problematiche relative a questa fase. Questo capitolo illustra le nozioni fondamentali relative alla fase di progettazione, che si possono così riassumere:

- i *principi* di progettazione, che sono le linee guida generali che si devono seguire per produrre software di qualità;
- le *tecniche* di progettazione, che sono i metodi mediante i quali si produce il programma in modo coerente con i principi;
- gli *strumenti* per la progettazione, intesi come mezzi espressivi (ad esempio i linguaggi di programmazione) e ausili di varia natura utilizzati nella progettazione.

#### 2.1 Principi di progettazione

I principi di progettazione sono le linee guida generali che dovrebbero costituire la base di ogni scelta e strategia di progetto. In questo paragrafo presentiamo le considerazioni sui principi limitando la discussione a tre di essi, che reputiamo i più significativi e importanti, e dai quali tutti gli altri possono essere in qualche modo derivati. Essi sono:

- distinzione tra il livello della *concettualizzazione* e il livello della *realizzazione*,
- uso dell'*astrazione*,
- *decomposizione* del problema e della soluzione in parti indipendenti.

##### 2.1.1 Distinzione tra concettualizzazione e realizzazione

Il primo principio di progettazione di cui ci occupiamo suggerisce di considerare la fase di progettazione come composta di due sottofasi distinte:



Figura 2.1 La struttura della fase di progettazione

- la concettualizzazione della soluzione, il cui prodotto è una descrizione astratta della soluzione del problema elaborativo, chiamata *schema concettuale di progetto*,
- la realizzazione in termini della tecnologia informatica, il cui prodotto è il programma redatto in un particolare linguaggio di programmazione.

La figura 2.1 mostra sinteticamente la struttura della fase di progettazione che risulta da questa suddivisione.

A parte la distinzione in due vere e proprie sottofasi, il principio di progettazione suggerisce di procedere durante tutto il progetto muovendosi su due piani distinti: il piano della concettualizzazione, in cui ci si concentra su cosa deve fare il programma, senza ancora affrontare il problema relativo a come il programma svolge le sue funzioni, ed il piano della realizzazione, in cui ci si concentra su come il programma realizza quanto deciso. Nel seguito di questo paragrafo discuteremo separatamente la concettualizzazione e la realizzazione.

### *Concettualizzazione*

Appartengono alla fase di concettualizzazione tutte quelle attività relative al concepimento della soluzione informatica del problema, dalla specifica degli oggetti da manipolare, fino alla ideazione delle scelte algoritmiche su cui si baserà poi il programma finale.

Come si vede dalla figura 2.1, l'input alla fase di concettualizzazione è l'input all'intera fase di progettazione, e cioè il documento di analisi, prodotto nella fase di raccolta e analisi dei requisiti. L'output della concettualizzazione è lo schema concettuale di progetto, ovvero la *specifica* di cosa deve fare il programma, ancora espressa in modo indipendente dal linguaggio di programmazione. A livello generale, lo schema concettuale di progetto è costituito da:

- l'esplicitazione dei dati di interesse per il progetto dell'applicazione, e la specifica delle loro proprietà;
- l'esplicitazione delle operazioni di interesse per il progetto dell'applicazione, e la specifica dei corrispondenti dati di input e di output.

Il modo in cui lo schema concettuale viene prodotto, formalizzato e specificato dipende dalle tecniche e dalle metodologie usate. Illustreremo nei capitoli 9 e 10 un metodo generale per il disegno dello schema concettuale di progetto, basato sulla tecnica della modularizzazione.

Consideriamo, ad esempio, un banale problema di ricerca dei numeri primi all'interno di una collezione di valori interi positivi. L'astrazione è quello strumento concettuale che ci consente di ignorare, ad un primo livello di dettaglio, l'organizzazione dettagliata dei dati in esame, di considerare in prima istanza la collezione degli elementi come un insieme senza struttura, e di concepire la soluzione come un'analisi degli elementi dell'insieme allo scopo di individuare tra essi quelli cercati, raccogliendoli in una seconda collezione. Già a questo livello di dettaglio, possiamo dire di avere ideato gli aspetti fondamentali della soluzione, ossia:

- la rappresentazione dei valori su cui il programma opererà mediante l'oggetto matematico "insieme",
- la realizzazione della soluzione algoritmica mediante l'operazione di analisi degli elementi dell'insieme,
- la rappresentazione del risultato, ancora mediante un insieme.

Si noti che, nelle considerazioni precedenti, abbiamo implicitamente assunto di avere a disposizione sia le operazioni fondamentali sugli insiemi (ad esempio l'analisi dei suoi elementi), sia un'operazione che, dato un valore intero, verifichi se esso sia primo. Abbiamo, cioè compiuto una strutturazione del nostro processo di concettualizzazione, decomponendo il problema in sottoproblemi, e astruendo dalle caratteristiche peculiari di questi sottoproblemi. È evidente che, al momento di dettagliare maggiormente la soluzione, ci occuperemo dei singoli sottoproblemi, ma lo faremo con la convinzione che, una volta trovata la soluzione ad essi, sappiamo come combinare queste soluzioni in modo corretto per ottenere il risultato globale.

Nell'esempio precedente sono presenti, sia pure in forma molto semplificata, gli altri principi fondamentali che riteniamo più rilevanti nella fase di concettualizzazione e che discutiamo nei prossimi due paragrafi: l'astrazione e la decomposizione. Ciò mette in evidenza che i principi di progettazione non sono tra loro indipendenti.

### ***Realizzazione***

Come già detto, intendiamo per realizzazione di un programma quel processo che, partendo dallo schema concettuale di progetto, permette di ottenere uno specifico programma che realizza gli elementi di tale schema. Ricordiamo che la fase successiva

al progetto è la verifica, che ha lo scopo di stabilire, tra le altre cose, se il programma sia corretto.

La fase di realizzazione viene spesso vista come scomposta in due attività:

1. scelta dell'architettura del programma in termini di un insieme di parti indipendenti, dette *moduli software*;
2. realizzazione dei singoli moduli, e generazione del programma complessivo.

La prima attività riguarda la necessità di scomporre il sistema complessivo in un insieme di moduli, stabilire le relazioni tra essi, e individuare con precisione le funzionalità che tali moduli devono assicurare. La scelta su quali e quanti moduli realizzare viene fatta sulla base di un'analisi di come aggregare gli elementi dello schema concettuale in moduli software.

La seconda attività è invece una attività prettamente di programmazione, in cui i vari moduli vengono realizzati nel linguaggio di programmazione scelto, vengono provati singolarmente, e vengono poi integrati in un unico programma. Una buona realizzazione di un programma, dopo una buona concettualizzazione, è quindi legata al corretto uso del linguaggio di programmazione. Per ottenere questo risultato è necessario avere una buona conoscenza del linguaggio di programmazione, cioè interpretare ogni sua forma espressiva esattamente allo stesso modo della macchina, e far uso delle strutture del linguaggio per esprimere efficacemente quanto si è ottenuto con la concettualizzazione. L'attività di realizzazione sarà poi facilitata dalla capacità del linguaggio di programmazione di ridurre il lavoro intellettuale richiesto per realizzare il programma a partire dal risultato di una concettualizzazione.

### 2.1.2 Astrazione

L'astrazione è un procedimento mentale che consente da una parte di evidenziare le caratteristiche pregnanti di una situazione o di un problema, e dall'altra di offuscare o addirittura ignorare gli aspetti che si ritengono secondari rispetto ad un determinato obiettivo. L'astrazione è un concetto molto generale, che comunemente viene utilizzato in tutte le attività mentali dell'uomo, e che si caratterizza in modo preciso quando si specializza ad attività informatiche.

Nell'ambito del progetto dei programmi, l'astrazione consente di giungere ad una soluzione in modo razionale e controllato.