

## Processo – parte VII

Leggere Cap. 9 Ghezzi et al.



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

1

## Strumenti

- Strumenti software che assistono gli ingegneri del software in tutte le fasi del progetto; in particolare
  - progettazione
  - codifica
  - test
- Evoluzione dovuta a
  - sviluppo della tecnologia
  - comprensione del processo software

Progetto 2B



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

2

## Sviluppo tecnologico

- Progressi nelle interfacce utente grafiche (GUI)
  - Editor grafici
  - Linguaggi visuali
- Progresso delle tecnologie basate su sistemi distribuiti
  - Strumenti per la gestione delle configurazioni
  - Gestione del lavoro di squadra (groupware)

Progetto 2B



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

3

## Maggiore integrazione

1. Strumenti a supporto di singole attività (codifica, debugging, test)
2. Ambienti integrati di sviluppo (IDE), insiemi di strumenti che funzionano insieme (in genere a supporto di un linguaggio)
3. Ambienti aperti: strumenti dotati di interfacce standard che consentono di interagire con altri strumenti

Progetto 2B



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

4

## Confronto fra strumenti

- Modalità di interazione: batch o interattivi
- Interfaccia: testuale o grafica
- Livello di formalità: sintassi e/o semantica formale
- Dipendenza dalla fase del ciclo di vita
- Grado di standardizzazione
- Dipendenza dal linguaggio: mono o multilinguaggio
- Statici o dinamici
- Sviluppo o gestione
- Mono-utente o multi-utente
- Supporto di sistemi distribuiti

## Strumenti Computer Aided Software Engineering (CASE)

- Molto diffusi, soprattutto per OOD
- Consentono la creazione e modifica specifiche di progetto in linguaggi grafici (spesso UML)
- Esempio: class diagram
- Generalmente consentono la generazione di scheletri di codice; i più evoluti anche il reverse engineering (dal codice al progetto).

## Editor

- Stand-alone o integrati in IDE
- Testuali o grafici
- Dedicati ad un formalismo o generali

## Editor di testi

- Funzionalità desiderabili:
  - syntax highlighting
  - autocompletamento
  - indentazione
  - macrolinguaggio

## Integrazioni di editor testuali e grafici

- Possibilità di inserire informazioni in modo testuale o grafico a seconda dei casi
- Possibilità di avere una vista testuale e una grafica sullo stesso documento (ad esempio, usando TDN e GDN)

## Editor di GUI

- Riducono drasticamente il tempo necessario allo sviluppo di GUI
- Alcuni generano unidirezionalmente il codice sorgente corrispondente alla GUI disegnata
- I più evoluti supportano l'editing concorrente:
  - nell'editor grafico
  - nell'editor di testo del codice corrispondente

## Linker

- Collegano tra loro oggetti correlati al fine di ottenere oggetti di maggiori dimensioni
- Risoluzione dei nomi
- Tradizionalmente, collegano file oggetto per ottenere file eseguibili
- Monolinguaggio o multilinguaggio
- In generale, collegamento di moduli

## Interpreti

- Strumenti per eseguire azioni specificate in una notazione formale
- Possono eseguire anche specifiche dei requisiti
- Importante che tollerino anche specifiche incomplete
- Si ottiene un prototipo del sistema
- A volte, animazione dei requisiti: esecuzione di un modello (ad es, FSM o rete di Petri)

## Generatori di codice

- Sviluppo del software: trasformazione da una specifica a una implementazione
- Strumenti per aiutare in questa trasformazione, in particolare nelle parti più ripetitive
- Ad es., dal diagramma delle classi UML ad uno scheletro Java

## Strumenti di 4a generazione

- Generano codice da un linguaggio di altissimo livello (linguaggio di 4a generazione, DBMS)
- Esempio: data la struttura di un DB relazionale, generazione di
  - maschere di immissione dati
  - vista griglia-dettaglio
  - pulsanti di navigazione fra tabelle in base a relazioni di chiave esterna

## Debugger

- Casi particolari di interpreti. Consentono di
  - ispezionare il valore delle variabili del programma
  - effettuare esecuzione passo-passo o fino al prossimo breakpoint
  - verificare asserzioni intermedie
  - valutare espressioni
  - watch (break se il valore di una variabile cambia)
  - senza modificare il codice sorgente
- Può servire anche per comprendere un programma

## Strumenti per test

- Documentazione dei test
  - form per definizione, archiviazione, recupero (test di regressione)
- Determinazione dei casi di test
  - white box: basati su criteri di copertura o esecuzione simbolica se è disponibile il codice
  - black box: se è disponibile la specifica
- Valutazione dei test
  - copertura, grado di successo
- Test di altre qualità: ad es. prestazioni

## Analizzatori statici

- Analisi di flusso
- Identificano possibili errori o flussi sospetti
  - uso di variabili non inizializzate
  - accesso a file non aperti
- Spesso integrati in
  - compilatori
  - IDE

## Strumenti per le interfacce grafiche

- User Interface Management System (UIMS): insieme di astrazioni di base (finestre, menu, barre di scorrimento, ecc.)
- Libreria di routine da collegare con l'applicazione
- Spesso collegamento a run-time
- Modello model-view-controller

## Strumenti di gestione delle configurazioni

- Strumenti di controllo delle versioni:
  - CVS, SVN, git, Mercurial
- Strumenti per la costruzione del prodotto:
  - make

## Strumenti per il tracciamento

- Aiutano a mantenere informazioni sul processo
- Esempio importante: strumenti per il tracciamento di difetti
- Consentono di gestire i difetti:
  - segnalazione da parte di un utente riempiendo un form
  - decisione da parte del management riguardo alla gestione
  - correzione

## Strumenti di reverse engineering

- Strumenti di *program understanding*
- Ricostruzione del processo di sviluppo all'indietro, dal codice sorgente ai requisiti
- Ad esempio: costruzione del grafo di controllo o del flusso dei dati

## Strumenti di reengineering

- Migrazione di sistemi esistenti verso nuove tecnologie
- Ad es., conversione da linguaggio procedurale a linguaggio ad oggetti

## Strumenti di supporto al processo

- Per organizzare e gestire il processo
- Memorizzazione delle liste di compiti dei progettisti
- Controllo dei compiti svolti dai progettisti
- Process-Oriented Software Engineering Environments (PSEE): gestiscono il processo a partire da una sua descrizione in un linguaggio formale

## Strumenti di management

- Strumenti per il disegno di diagrammi PERT e di Gantt
- Strumenti per la previsione dei costi