

Ingegneria del Software

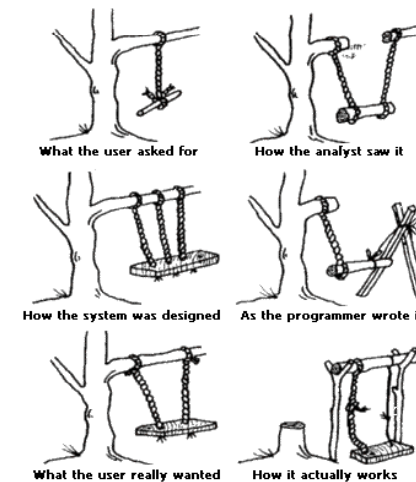
Il Processo Software (II parte)

Antonino Staiano

e-mail: antonino.staiano@uniparthenope.it

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Le attività di processo



Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Attività di processo

- Specifica, sviluppo, convalida ed evoluzione sono organizzate in modo diverso a seconda del modello di processo adottato
 - Nel modello a cascata sono organizzate in sequenza
 - Nel modello evolutivo sono intrecciate
- Il modo in cui ciascuna attività è svolta dipende dal tipo di software da creare e dalle persone e strutture organizzative coinvolte
 - Non c'è un unico modo per organizzarle

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Attività di processo: Specifiche del software

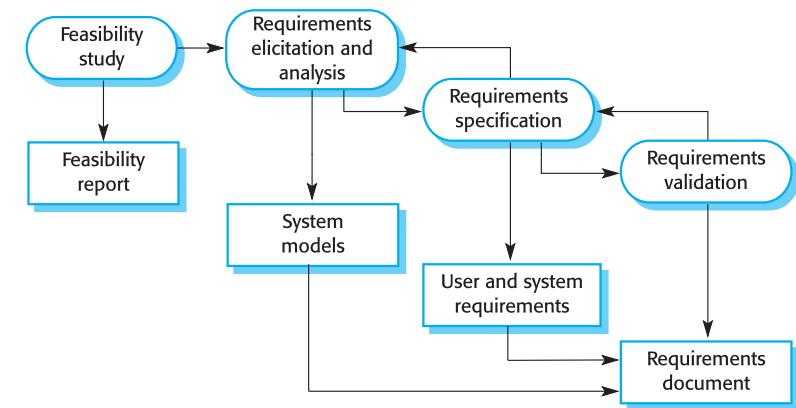
- La creazione delle specifiche del software è il processo (ingegneria dei requisiti)
 - per capire e definire quali funzionalità sono richieste dal sistema
 - per identificare i vincoli operativi e di sviluppo
- Fase critica del processo software
 - Eventuali errori portano ad errori certi nelle fasi di progettazione ed implementazione

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Attività di processo: Specifiche del software

- Il risultato dell'ingegneria dei requisiti è un documento dei requisiti che rappresenta la specifica del sistema
- I requisiti sono presentati con due differenti livelli di dettaglio
 - Clienti ed utenti finali necessitano di una formulazione dei requisiti ad alto livello
 - Gli sviluppatori del sistema necessitano di una specifica dettagliata

Il processo di ingegneria dei requisiti



Specifiche del software: fasi principali

- Studio di fattibilità
 - Si valuta se le necessità degli utenti possono essere soddisfatte con le tecnologie HW/SW correnti
 - Se il sistema può essere sviluppato rispetto ai vincoli economici e se è efficiente nei costi per l'azienda
- Il risultato consente di decidere se continuare o meno con un'analisi più dettagliata

Specifiche del software: fasi principali

- Scoperta ed analisi dei requisiti
 - Processo di derivazione dei requisiti del sistema attraverso l'osservazione dei sistemi esistenti, la discussione con gli utenti e i clienti, l'analisi dei compiti ecc.
 - Può comportare lo sviluppo di uno o più prototipi

Specifiche del software: fasi principali

- Specifica dei requisiti
 - Attività di traduzione delle informazioni raccolte durante la fase di analisi in un documento che definisce un insieme di requisiti
 - **Requisiti utente:** proposizioni astratte dei requisiti del sistema per i clienti e gli utenti finali
 - **Requisiti di sistema:** descrizione dettagliata delle funzionalità che devono essere fornite
- Convalida dei requisiti
 - Si controlla che i requisiti siano realistici, consistenti e completi
 - Si scoprono gli errori (inevitabili) nel documento dei requisiti che dovranno essere corretti

Specifiche del software: fasi principali

- Le attività del processo dei requisiti non sono eseguite con una sequenza rigida
 - L'analisi continua durante la fase di definizione e la specifica e durante tutto il processo nascono nuovi requisiti
 - Analisi, definizione e specifica sono intrecciate

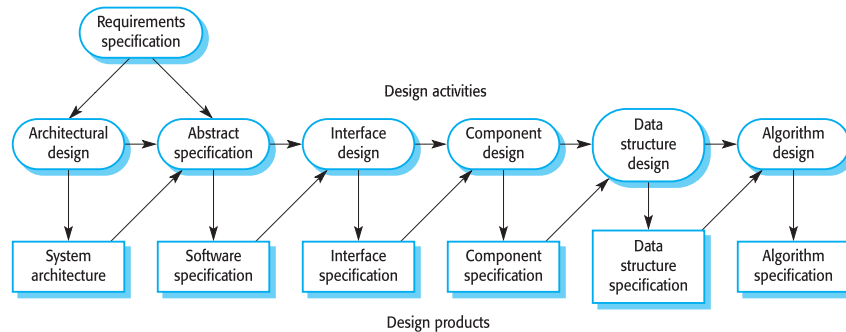
Attività di processo: Progettazione ed implementazione del software

- La fase di implementazione è il processo di conversione delle specifiche in un sistema eseguibile
 - Comprende sempre processi di progettazione e programmazione del software
 - In alcuni approcci (modello evolutivo) può includere un perfezionamento delle specifiche
- Il progetto è la descrizione
 - della struttura del software che si deve implementare
 - dei dati che fanno parte del sistema
 - delle interfacce tra i componenti del sistema
 - degli algoritmi usati

Attività di processo: Progettazione ed implementazione del software

- I progettisti non ottengono alla prima stesura un risultato completo
 - Sviluppano il progetto ciclicamente attraverso una serie di versioni
 - Si aggiungono formalità e dettagli durante lo sviluppo con controlli incrociati per correggere i primi progetti
- Il processo di progettazione può richiedere lo sviluppo di vari modelli del sistema a diversi livelli di astrazione
 - Scomposto un progetto, si scoprono errori ed omissioni nelle prime fasi riportati sui modelli di progetto precedenti al fine di migliorarli

Il processo di progettazione del SW



Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Fasi dell'attività di progettazione

- La figura mostra che gli stadi del processo di progettazione sono sequenziali, in realtà essi si intrecciano
 - Inevitabile che ci sia uno scambio di informazioni da una fase all'altra per ottenere nuove rielaborazioni
 - La specifica per una data fase corrisponde all'output della fase precedente
 - Astratta e formale per chiarire i requisiti o indicare dettagli realizzativi di alcune parti del sistema
- Il risultato del processo è la specifica di algoritmi e strutture dati da implementare

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Fasi dell'attività di progettazione

1. Progettazione dell'architettura
 - Identificazione e documentazione dei sottosistemi e delle loro relazioni
2. Specifiche astratte
 - Per ogni sottosistema è prodotta una specifica astratta dei servizi forniti e dei vincoli a cui è sottoposto
3. Progettazione dell'interfaccia
 - È progettata e documentata l'interfaccia di ciascun sottosistema
 - Specifiche non ambigue in modo da poter usare il sottosistema senza conoscerne il funzionamento

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Fasi dell'attività di progettazione

4. Progettazione dei componenti
 - Progettazione delle interfacce dei componenti e assegnazione dei servizi da fornire
5. Progettazione delle strutture dati
 - Progettate e specificate le strutture dati da usare in fase di implementazione
6. Progettazione degli algoritmi
 - Progettati e specificati gli algoritmi che devono fornire i servizi

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

Attività di progettazione

- Un approccio alla fase di progettazione è rappresentato dai metodi strutturati
 - Producono modelli grafici del sistema dai quali viene generato automaticamente codice
 - Inventati negli anni '70 per la progettazione orientata alle funzioni
 - Negli anni '90 diversi metodi sviluppati per la progettazione orientata agli oggetti furono unificati con la creazione di UML (Unified Modeling Language) e dei relativi processi di progettazione unificati

Attività di progettazione: metodi strutturati

- Un metodo strutturato comprende un modello di processo di progettazione, le notazioni per rappresentare il progetto, i formati dei rapporti, le regole e le linee guida di progettazione
- Un metodo strutturato può prevedere alcuni o tutti i seguenti modelli di sistema:
 - Modello ad oggetti
 - Mostra le classi del sistema e le loro dipendenze
 - Modello a sequenze
 - Mostra l'interazione degli oggetti nel sistema durante l'esecuzione

Attività di progettazione: metodi strutturati

- Modello a transizione di stato
 - Mostra gli stati del sistema e i trigger per la transizione da uno stato all'altro
- Modello strutturale
 - Sono documentati i componenti del sistema e le loro aggregazioni
- Modello a flusso di dati
 - Viene modellato il sistema usando le trasformazioni dei dati che avviene durante l'elaborazione (per sistemi real-time ed aziendali)
- I metodi strutturati sono notazioni standard e delle rappresentazioni di buona prassi
 - Seguendo ali modelli e applicando le linee guida si può avere un progetto valido

Sviluppo

- L'implementazione del sistema segue i processi di progettazione
 - Gli stadi finali della progettazione e quelli iniziali dello sviluppo si intrecciano
 - Si possono usare gli strumenti CASE per la generazione di uno scheletro del programma a partire dalla progettazione
- La programmazione è un'attività individuale
 - Non c'è un processo generico che i programmatori possono seguire

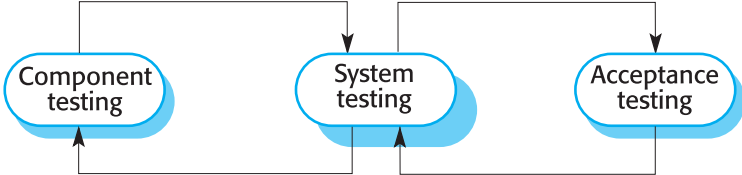
Sviluppo e debug

- I programmatori testano il codice che hanno sviluppato e ne mettono in luce i difetti da rimuovere dal programma (debug)
 - I processi di test e debug sono diversi
 - Il test si preoccupa di stabilire l'esistenza dei difetti, il debug di localizzarli e correggerli
 - Quando si esegue il debug si generano ipotesi sul comportamento visibile del programma che vengono verificate per trovare la causa del problema
 - La verifica delle ipotesi può richiedere il tracciamento manuale del codice oppure la necessità di scrivere alcuni test case per localizzare il problema

Convalida del software

- La verifica e la convalida del software è atta a mostrare che il sistema è conforme alle sue specifiche e che soddisfa le aspettative del cliente
 - Comprende ispezioni e revisioni ad ogni stadio del processo software, dalla definizione dei requisiti utente allo sviluppo del programma
 - La maggior parte dei costi è sostenuta dopo l'implementazione del programma in fase di test dell'operatività del sistema

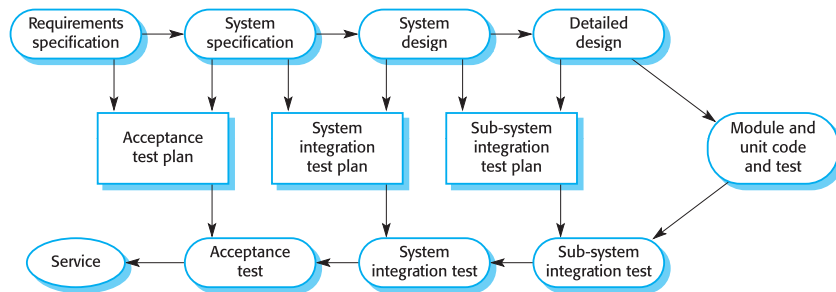
Convalida del software

- I sistemi non dovrebbero mai essere testati come un'unica entità
 - Si testano prima i componenti del sistema, poi il sistema integrato e il sistema con i dati del cliente
- 
- ```
graph LR; CT(Component testing) --> ST(System testing); ST --> AT(Acceptance testing); AT --> CT; AT --> ST;
```
- Il diagramma illustra il processo di convalida del software attraverso tre fasi principali: Component testing, System testing e Acceptance testing. Le frecce indicano un flusso sequenziale da Component testing a System testing, e da System testing a Acceptance testing. Inoltre, ci sono frecce di ritorno da Acceptance testing a Component testing e da Acceptance testing a System testing, suggerendo un ciclo di feedback o una ripetizione delle fasi in caso di problemi.
- I difetti nei componenti sono scoperti nelle prime fasi del processo; i problemi con le interfacce durante l'integrazione
    - I difetti scoperti richiedono un debug del programma il che potrebbe implicare la ripetizione di altri stadi del processo di test
      - Gli errori nei componenti possono apparire durante il test del sistema

## Convalida del software

- Gli stadi del processo di test sono 3
  1. **Test dei componenti:** si testano i componenti individuali. Ogni componente è testato in modo autonomo
  2. **Test del sistema:** si integrano i componenti per formare il sistema. Ci si preoccupa di trovare gli errori causati da interazioni impreviste tra i componenti. Si provvede a convalidare la conformità dei requisiti e a verificare le proprietà complessive del sistema
  3. **Test di accettabilità:** Il sistema viene testato con le informazioni fornite dai clienti anziché con dati simulati. Il test di accettabilità può rivelare errori ed omissioni nella definizione dei requisiti del sistema
    - I dati reali fanno lavorare il sistema in modo diverso rispetto ai dati di test.

## Fasi di test



## Testing

### ■ alpha test

- Per i sistemi sviluppati per un singolo cliente continua fino a che sviluppatori e cliente concordano che il sistema consegnato è un'implementazione accettabile dei requisiti di sistema

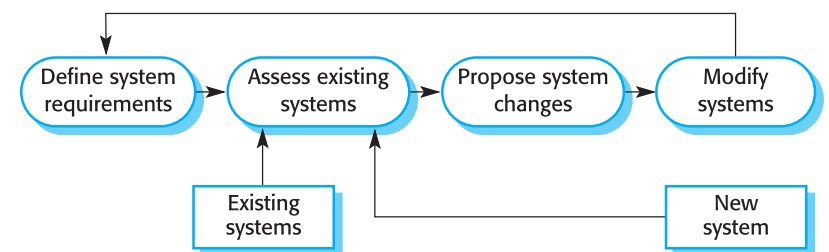
### ■ beta test

- si utilizza quando un sistema viene venduto come prodotto software e comporta la consegna del sistema ad una serie di possibili clienti che accettano di usarlo per un certo periodo e di riferire i problemi agli sviluppatori

## Evoluzione del software

- Il software è un artefatto inerentemente flessibile
  - Soggetto a numerosi cambiamenti
- Storicamente processo di sviluppo e manutenzione sono stati considerati separati
  - In realtà, la distinzione tra sviluppo e manutenzione è quasi del tutto sparita
  - I software completamente nuovi sono pochi
    - Lo sviluppo ed il mantenimento sono visti come un tutt'uno. L'ingegneria del software è un processo evolutivo in cui il SW viene modificato continuamente durante il ciclo di vita in risposta ai cambiamenti dei requisiti e delle necessità del cliente

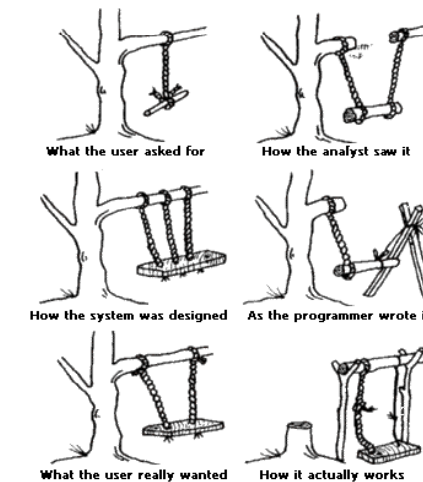
## Evoluzione del sistema



## Attività di processo: distribuzione tipica dello sforzo

- Relativamente alle fasi comprese tra l'analisi dei requisiti e il testing di sistema
  - 18% requisiti
  - 19% progettazione
  - 34% codifica
  - 29% test
- comuni variazioni del  $\pm 10\%$ 
  - ... dati ricavati da 125 progetti in Hewlett Packard

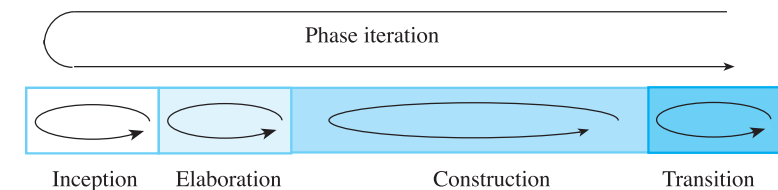
## Rational Unified Process (RUP)



## Il Rational Unified Process (RUP)

- Un moderno modello di processo derivato dal lavoro su UML e del processo associato
  - Modello di processo ibrido: contiene elementi di tutti i modelli di processo generici
  - Aiuta i cicli e illustra le buone prassi nella specifica e nella progettazione
- Descritto normalmente da tre prospettive
  - Una prospettiva dinamica che mostra le fasi del modello nel tempo
  - Una prospettiva statica che mostra le attività di processo coinvolte
  - Una prospettiva pratica che suggerisce le buone prassi da seguire durante il processo

## Fasi del RUP





## Fasi del RUP

- Il RUP organizza il progetto del processo software per fasi sequenziali e identifica quattro fasi discrete
  - Sono strettamente correlate agli aspetti aziendali anziché a quelli tecnici
- 1. Avvio (inception)
  - L'obiettivo è stabilire un business case per il sistema.
  - Si identificano tutte le entità esterne (persone e sistemi) che interagiranno con il sistema e definire tali interazioni
  - Sulla base di queste informazioni si stimano i contributi che il sistema può dare all'azienda
    - Se è minimo il progetto può essere annullato

## Fasi del RUP

- 2. Elaborazione (Elaboration)
  - Sviluppare una comprensione del dominio del problema
  - Stabilire una struttura architetturale del sistema
  - Sviluppare il piano del progetto
  - Identificare i rischi chiave
- Al completamento di tale fase si dovrebbe avere un modello dei requisiti del sistema (diagrammi UML use case), una descrizione architetturale e un piano di sviluppo del software

## Fasi del RUP

- 3. Costruzione (Construction)
  - Progettazione, programmazione e test del sistema
    - Più parti del sistema sono sviluppate parallelamente e poi integrate
    - Al completamento si dovrebbe avere un sistema software funzionante e la relativa documentazione pronta per gli utenti
- 4. Transizione (Transition)
  - si sposta il sistema dalla comunità dello sviluppo a quella utente e lo si fa funzionare nell'ambiente reale
  - Al completamento si dovrebbe avere un sistema software documentato che funziona correttamente nel suo ambiente operativo

## Fasi del RUP

- La ciclicità e l'incremento sono supportati nel RUP in due modi
  - Ogni fase può essere eseguita in modo ciclico con risultati sviluppati in modo incrementale
  - L'intero insieme delle fasi può essere eseguito in modo incrementale

## RUP: Prospettiva statica

- La prospettiva statica del RUP si concentra sulle attività di produzione del software chiamate *workflow*
- Sei sono i workflow principali identificati nel processo e tre di supporto
- Poiché il RUP è stato progettato insieme a UML la descrizione dei workflow è orientata ai modelli associati a UML

## RUP: Workflow statici

| Workflow                                        | Descrizione                                                                                                                                                                                     |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modellazione delle attività aziendali           | I processi aziendali sono modellati utilizzando business case                                                                                                                                   |
| Requisiti                                       | Vengono identificati gli attori che interagiscono col sistema e sviluppati casi d'uso per modellare i requisiti del sistema                                                                     |
| Analisi e progettazione                         | Viene creato e documentato un modello di progetto utilizzando modelli architetturali, dei componenti, degli oggetti                                                                             |
| Implementazione                                 | I componenti del sistema sono implementati e strutturati nell'implementazione dei sottosistemi. La generazione automatica del codice a partire dai modelli di progetto accelera questo processo |
| Test                                            | Il test è un processo iterativo eseguito in parallelo all'implementazione. Il test di sistema segue il completamento di questa                                                                  |
| Rilascio                                        | Viene creata una release del prodotto, viene distribuita agli utenti e installata nelle loro postazioni di lavoro                                                                               |
| Gestione della configurazione e delle modifiche | Questo workflow di supporto gestisce i cambiamenti del sistema                                                                                                                                  |
| Gestione del progetto                           | Questo workflow di supporto gestisce lo sviluppo del sistema                                                                                                                                    |
| Ambiente                                        | Questo workflow rende disponibili al team di sviluppatori del software gli strumenti adeguati                                                                                                   |

## RUP: prospettiva pratica

- Descrive la buona prassi di ingegneria del software che si raccomanda di usare nello sviluppo dei sistemi
- Sono sei le pratiche fondamentali
  - **Sviluppare il software ciclicamente**
    - Pianificare gli incrementi del sistema basati sulle priorità del cliente. Le funzioni con priorità maggiore devono essere sviluppate e consegnate all'inizio del processo di sviluppo
    - Gestire i requisiti
      - **Documentare esplicitamente i requisiti del cliente e i cambiamenti effettuati. Analizzare l'impatto dei cambiamenti prima di accettarli**

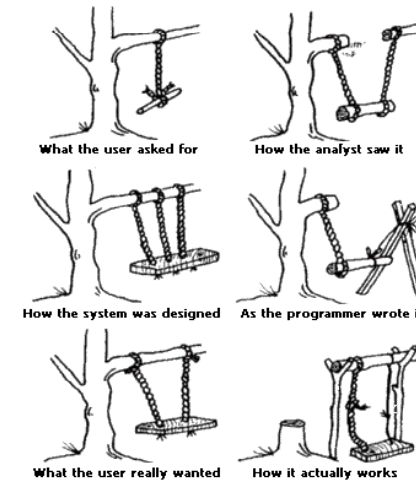
## RUP: prospettiva pratica

- **Usare architetture basate sui componenti**
  - Strutturare l'architettura del sistema con un approccio a componenti
- **Creare modelli visivi del software**
  - Usare modelli grafici UML per rappresentare le visioni statiche e dinamiche del software
- **Verificare la qualità del software**
  - Assicurarsi che il software raggiunga gli standard di qualità dell'organizzazione
- **Controllare le modifiche al software**
  - Gestire i cambiamenti del software usando un sistema per la gestione delle modifiche e procedure e strumenti di gestione della configurazione

## RUP: considerazioni

- Non è un processo adatto a tutti i tipi di sviluppo
  - Rappresenta una nuova generazione di processi generici
  - Le maggiori innovazioni
    - Separazione delle fasi e i workflow
    - Accettazione della distribuzione del software nell'ambiente dell'utente come parte del processo
  - Le fasi sono dinamiche ed hanno obiettivi
  - I workflow sono statici e sono attività tecniche non associate a una singola fase
    - Utilizzate durante tutto lo sviluppo per il perseguimento degli obiettivi di ciascuna fase

## Computer-Aided Software Engineering (CASE)



## Computer-aided software engineering

- Computer-aided software engineering (CASE) è il software usato per il supporto ai processi di sviluppo del sistema e di evoluzione dello stesso
- Consentono di automatizzare le varie attività del processo software
  - Editor grafici per lo sviluppo del modello del sistema
  - Dizionario di dati per gestire le entità del progetto
  - Costruttore di interfacce utente grafiche per la realizzazione delle interfacce utente
  - Programmi per il debug che supportano l'individuazione degli errori nel codice
  - Traduttori automatici per generare nuove versioni di un programma

## Tecnologia CASE

- La tecnologia dei CASE ha portato notevoli miglioramenti nel processo software. Tuttavia, i risultati ottenuti sono inferiori a quelli previsti
  - L'ingegneria del software richiede creatività la quale è difficile da automatizzare
  - L'ingegneria del software è un'attività svolta in gruppo e, per grandi progetti soprattutto, la maggior parte del tempo i membri del team la impiegano interagendo con gli altri membri del team, la qual cosa non è supportata dalla tecnologia CASE

## Classificazione CASE

- La classificazione aiuta a comprendere i differenti strumenti CASE e il loro ruolo nel supporto alle attività del processo software
- Prospettiva funzionale
  - I tool sono classificati in base alla rispettive funzioni specifiche
- Prospettiva di processo
  - I tool sono classificati in base alle attività di processo che sono supportate
- Prospettiva di integrazione
  - I tool sono classificati in base alla loro organizzazione in unità integrate che forniscono supporto ad una o più attività del processo

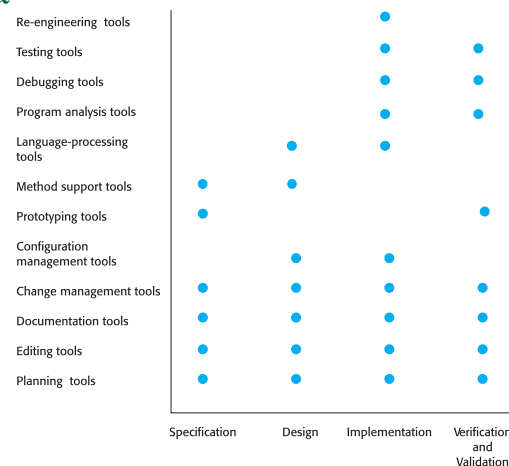
Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Classificazione funzionale degli strumenti CASE

| Tipo di tool                               | Esempi                                                                              |
|--------------------------------------------|-------------------------------------------------------------------------------------|
| Strumenti di pianificazione                | Strumenti PERT, strumenti di stima, fogli di calcolo                                |
| Strumenti di editing                       | Editor di testi e di diagrammi, elaboratori di testi                                |
| Strumenti di gestione delle modifiche      | Strumenti per il tracciamento dei requisiti, strumenti di controllo delle modifiche |
| Strumenti di gestione della configurazione | Sistemi per la gestione delle versioni, strumenti per la costruzione del sistema    |
| Strumenti di prototipizzazione             | Linguaggi di altissimo livello, generatori di interface utente                      |
| Strumenti di supporto ai procedimenti      | Editor di progetto, dizionari di dati, generatori di codice                         |
| Strumenti di elaborazione del linguaggio   | Compilatori, interpreti                                                             |
| Strumenti di analisi dei programmi         | Generatori di riferimenti incrociati, analizzatori statici, analizzatori dinamici   |
| Strumenti di test                          | Generatori di dati di test, comparatori di file                                     |
| Strumenti di debug                         | Sistemi interattivi di debug                                                        |
| Strumenti di documentazione                | Programmi di impostazione di pagina, editor di immagini                             |
| Strumenti di re-ingegnerizzazione          | Sistemi di riferimento incrociato, sistemi di ristrutturazione del programma        |

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Classificazione dei tool CASE basata sulle attività



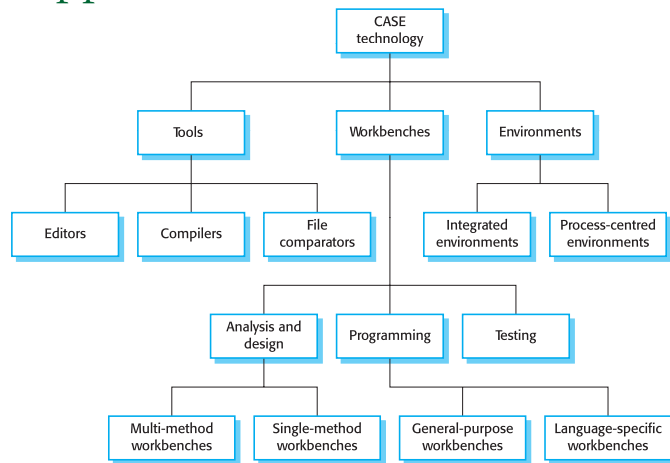
Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Integrazione CASE

- Strumenti
  - Supportano singoli compiti di processo, come la verifica della coerenza di un progetto, la compilazione di un programma, ecc.
- Workbench
  - Supportano fasi o attività del processo come la specifica, la progettazione. Solitamente consistono di un insieme di strumenti più o meno integrati
- Ambienti di sviluppo
  - Supportano tutto il processo software (o una gran parte) e includono diversi workbench integrati

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Strumenti, workbench e ambienti di sviluppo



Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Sommario

- I processi software sono attività coinvolte nella produzione e nell'evoluzione di un sistema software
- I modelli di processo software sono rappresentazioni astratte di tali processi
- Le principali attività di un processo software sono la specifica, la progettazione e l'implementazione, la validazione e l'evoluzione
- I modelli di processo generico descrivono l'organizzazione dei processi software. Esempio sono il modello a cascata, lo sviluppo evolutivo e l'ingegneria del software basata su componenti
- I modelli di processo ciclico rappresentano il processo software come un ciclo di attività. Vantaggio: evita gli impegni prematuri nella specifica o nella progettazione. Esempi sono lo sviluppo incrementale ed il modello a spirale

Ingegneria del Software, a.a. 2008/2009 – A. Staiano

## Sommario

- L'ingegneria dei requisiti è il processo di sviluppo delle specifiche del software
- I processi di progettazione e di implementazione trasformano una specifica in un programma eseguibile
- La validazione coinvolge il controllo che il sistema aderisca alle sue specifiche ed alle necessità degli utenti
- L'evoluzione riguarda la modifica del sistema dopo che è messo in opera
- Il Rational Unified Process è un modello di processo generico che separa le attività dalle fasi
- La tecnologia CASE supporta le attività del processo software

Ingegneria del Software, a.a. 2008/2009 – A. Staiano