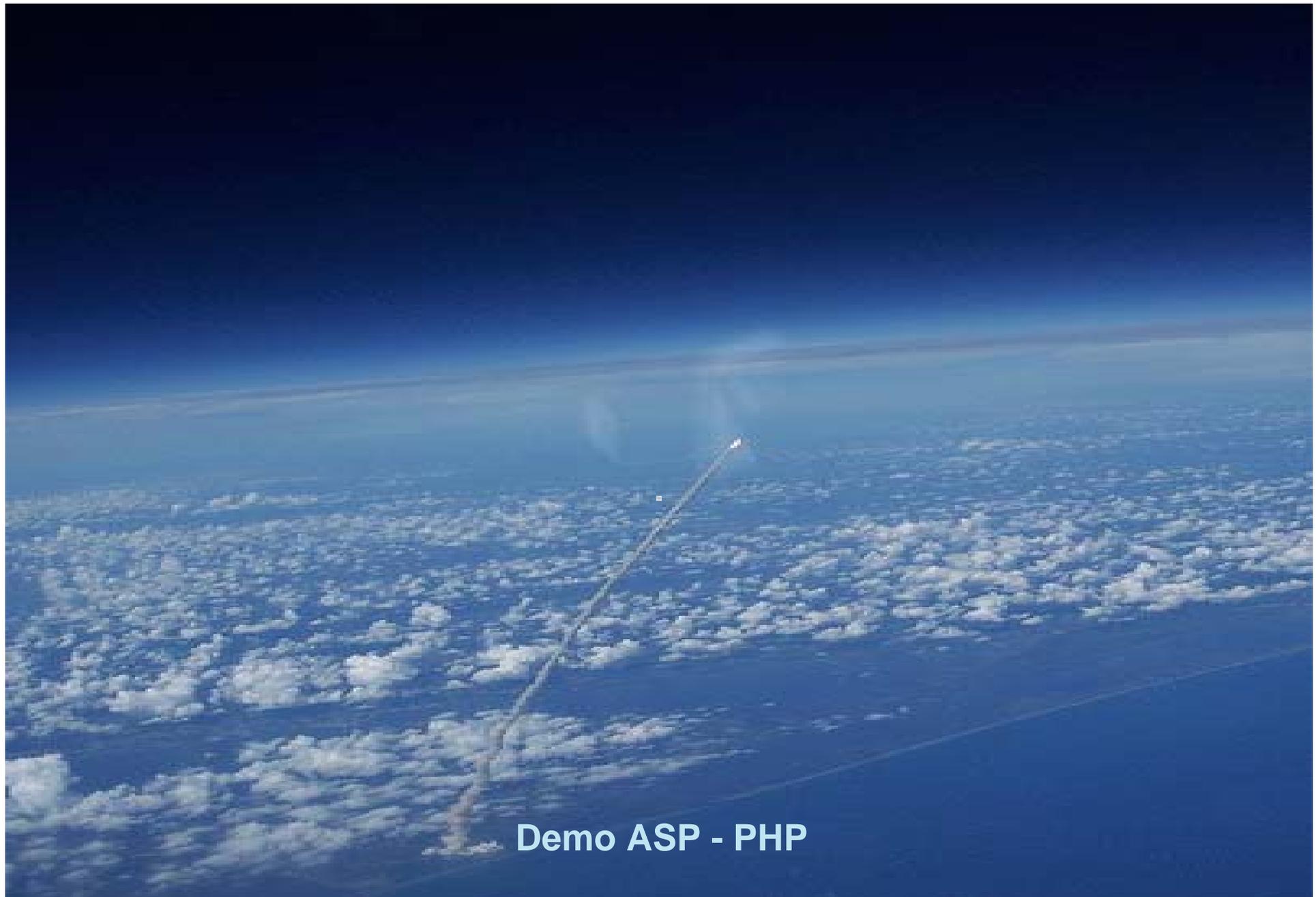




V lezione

Sviluppo sistemi

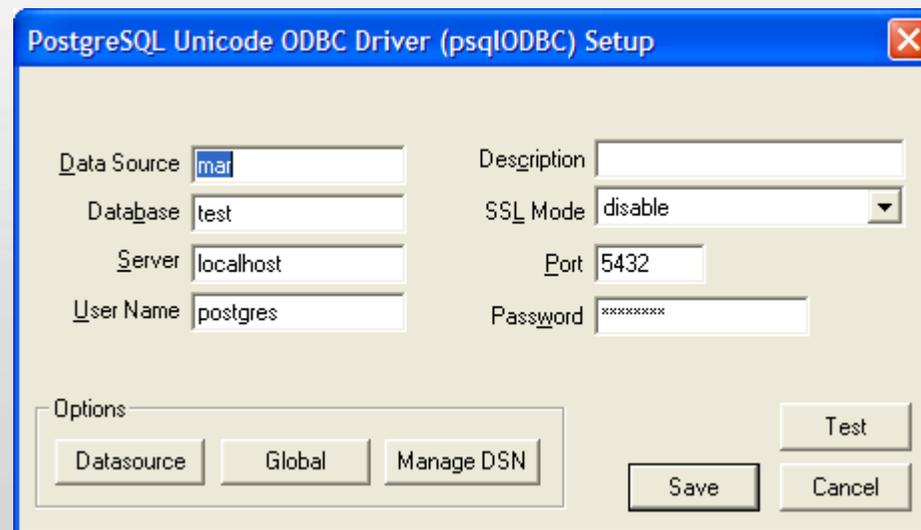
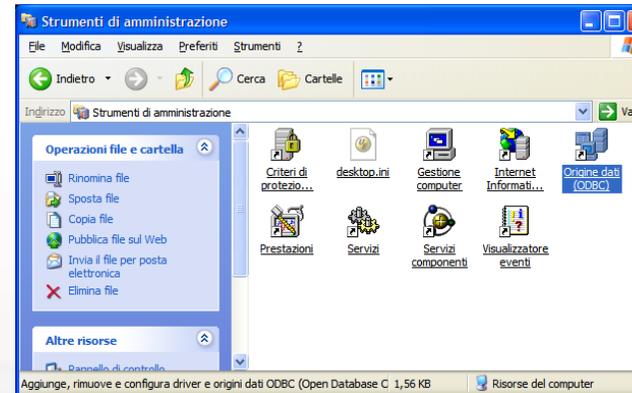
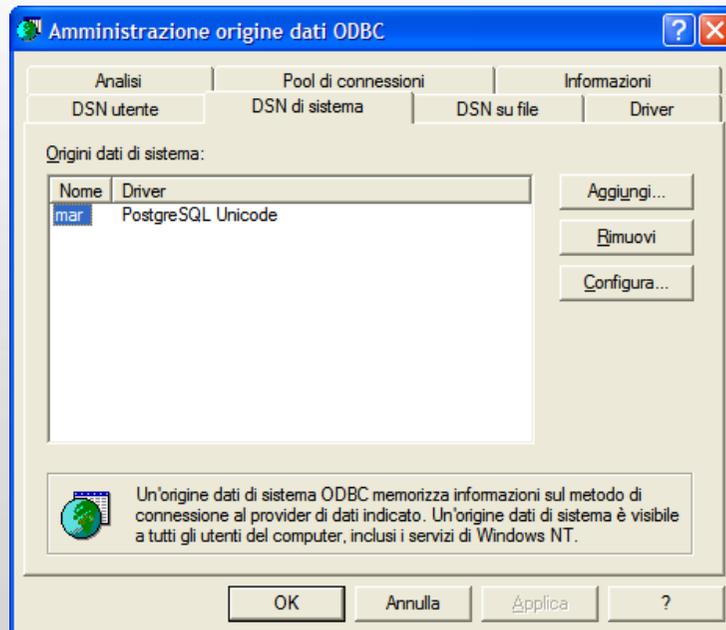
GIS e Geo WEB: piattaforme e architetture



Driver ODBC per PostgreSQL

Installazione driver

- Lanciare psqlodbc.msi
- Configurare un DSN di sistema



Esercitazione PHP

▪ Installazione php su IIS

- <http://www.php.net/downloads.php> (versione 5.2.x.ZIP)
- Creare la cartella C:\php e scompattare lo ZIP dentro
- Copiare il file php5ts.dll nella cartella C:\WINDOWS
- Rinominare il file **php.ini-dist** in **php.ini**
- Integrare il PHP 5 come CGI di IIS:
 - Nella console di configurazione di IIS, andare in 'Home Directory' quindi 'Configuration' e 'aggiungi'.
 - Specificare il path del modulo cgi 'c:\php\php-cgi.exe'.
 - Specificare l'estensione dei file da interpretare come .php.
 - Marcare la checkbox "modulo script".
 - Riavviare IIS.
 - Creare la prima applicazione php

```
<?php
phpinfo();
?>
```
 - Salvare il file in **C:\inetpub\wwwroot**
 - Lanciare <http://localhost:83/info.php> per verificare l'installazione

Php su Apache

Esercitazione PHP

- Installazione Apache
- Installazione php su Apache
 - <http://www.php.net/downloads.php> (versione 5.2.x.ZIP)
 - Creare la cartella C:\php e scompattare lo ZIP dentro
 - Copiare il file php5ts.dll nella cartella C:\WINDOWS
 - Copiare e rinominare il file **php.ini-dist** in **php.ini**
 - Integrare il PHP 5 in Apache
 - Cercare il file httpd.conf in C:\Programmi\Apache Software Foundation\Apache2.2\conf
 - Aprirlo con un editor, posizionarsi alla fine della sezione LoadModule e inserire le righe

```
LoadModule php5_module "c:/php/php5apache2_2.dll"  
AddType application/x-httpd-php .php  
PHPIniDir "C:/php"
```
 - Salvare e Chiudere il file httpd.conf
 - Riavviare Apache
 - Creare la prima applicazione php

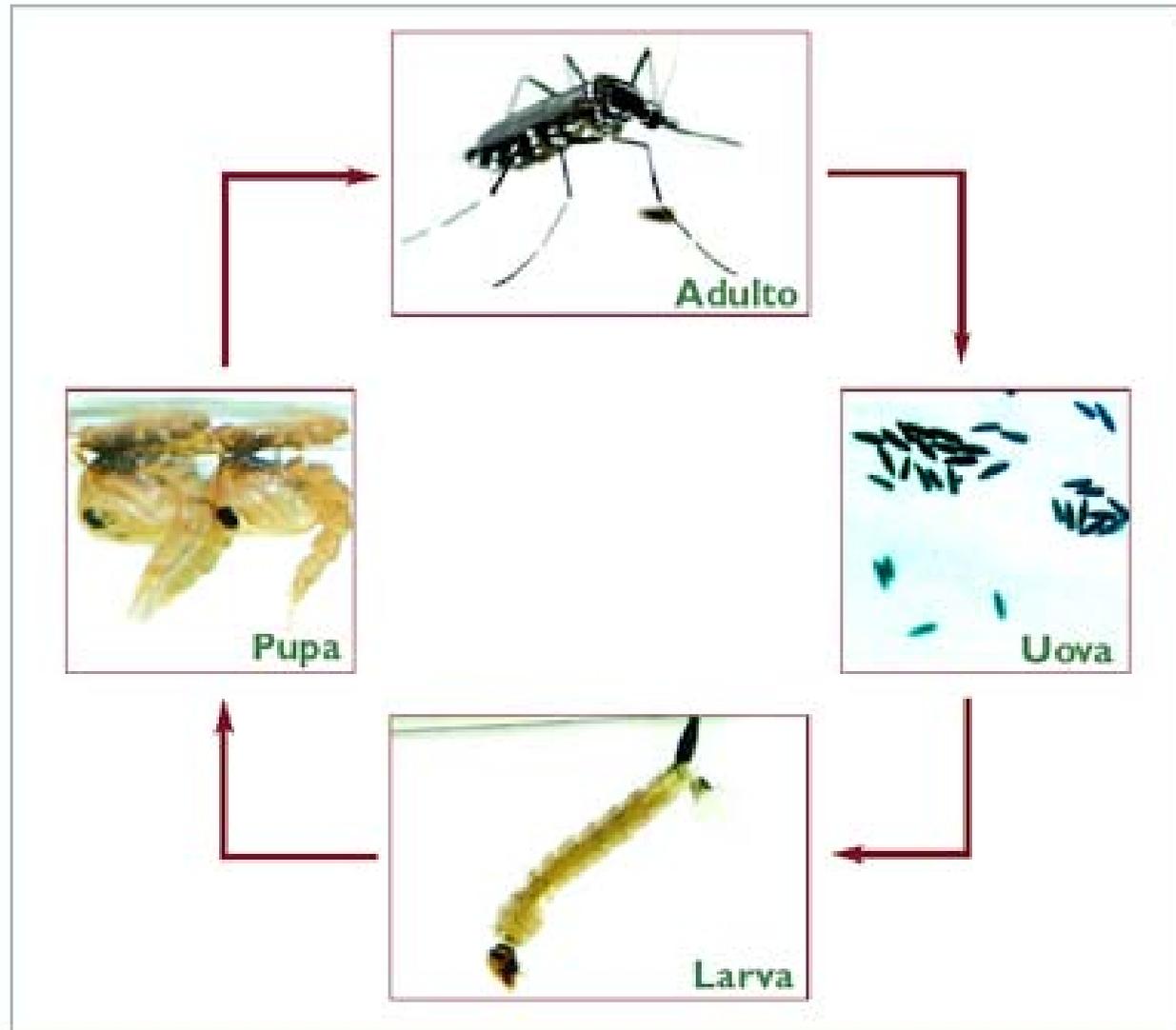
```
<?php  
phpinfo();  
?>
```
 - Salvare il file in **C:\Programmi\Apache Software Foundation\Apache2.2\htdocs**
 - Lanciare <http://localhost/info.php> per verificare l'installazione

Esercitazione PHP

- Scrivere del testo, richiamare funzioni lato server
 - Php1.php, php2.php, php3.php
- Condizionali
 - Php4.php
- Risultati da una form
 - Action.htm, Action.php
- Formattare il testo
- Creare variabili
- Connettersi al database
 - Conn.php
- Risorse PHP
 - <http://www.php.net/>
 - <http://php.resourceindex.com/>
 - <http://www.hotscripts.com/PHP/>
 - <http://www.phpworld.com/>
 - <http://www.morpheusweb.it/html/manuali/php.asp>

Esercitazione ASP

- Scrivere del testo, creare variabili, cicli, formattare il testo
 - aspvb.asp
- Connettersi al database, condizionali
 - utenti.asp
- Risorse
 - <http://www.asp.net>
 - <http://msdn.microsoft.com>
 - <http://www.w3schools.com/asp/>
 - <http://www.aspitalia.com/>
 - <http://www.morpheusweb.it/html/manuali/asp.asp>



Il ciclo di vita

Un caso particolare: ECSS

European Cooperation for Space Standardization

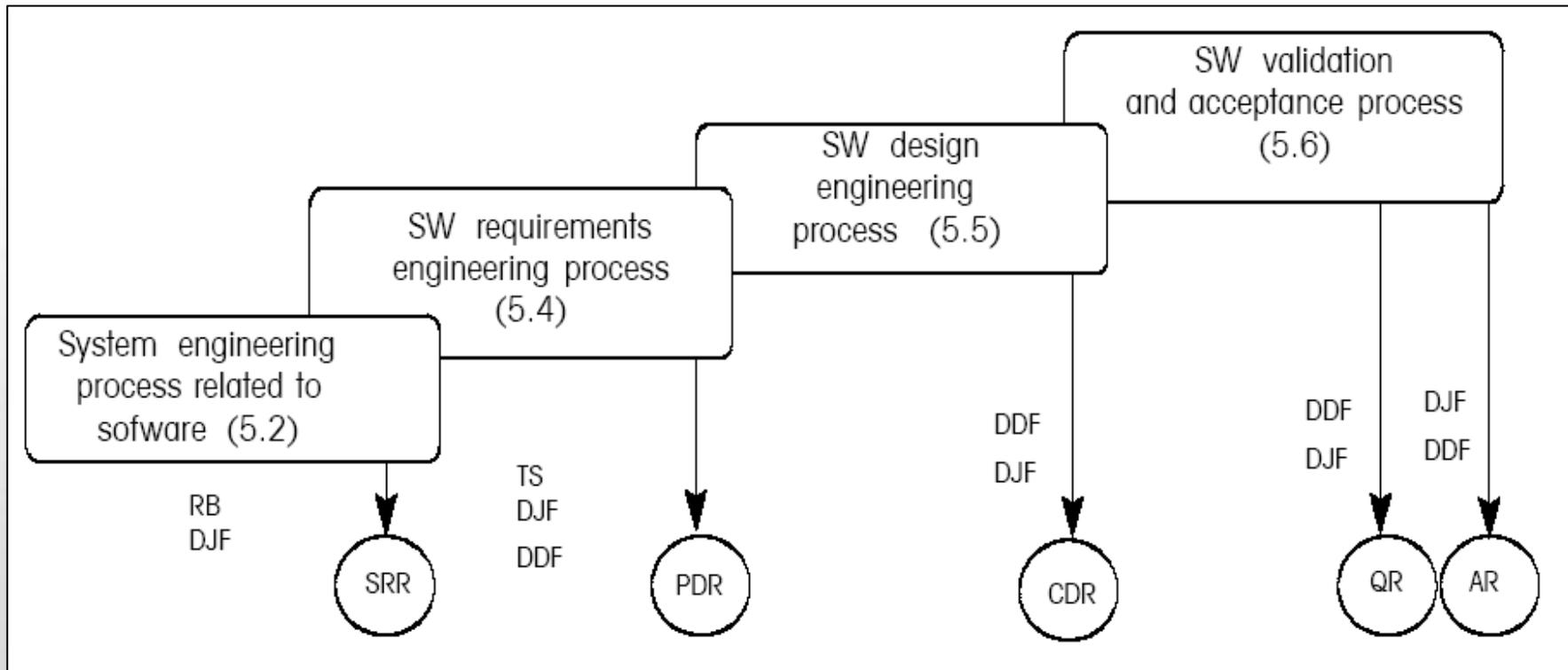
■ Fasi del progetto

- SRR: System Requirement Review
- PDR: Preliminary Design Review
- CDR: Critical Design Review
- QR: Qualification Review
- AR: Acceptance Review

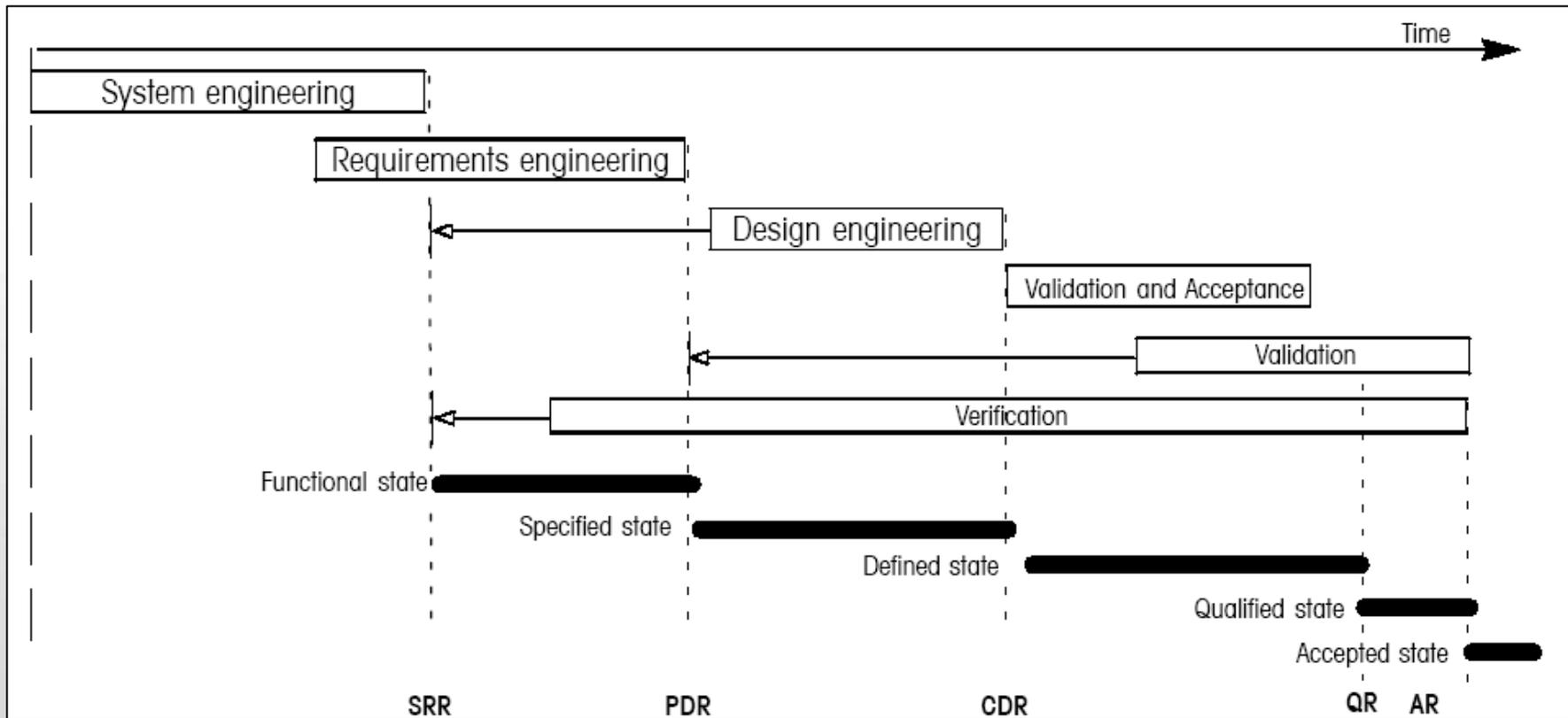
■ Organizzazione del progetto

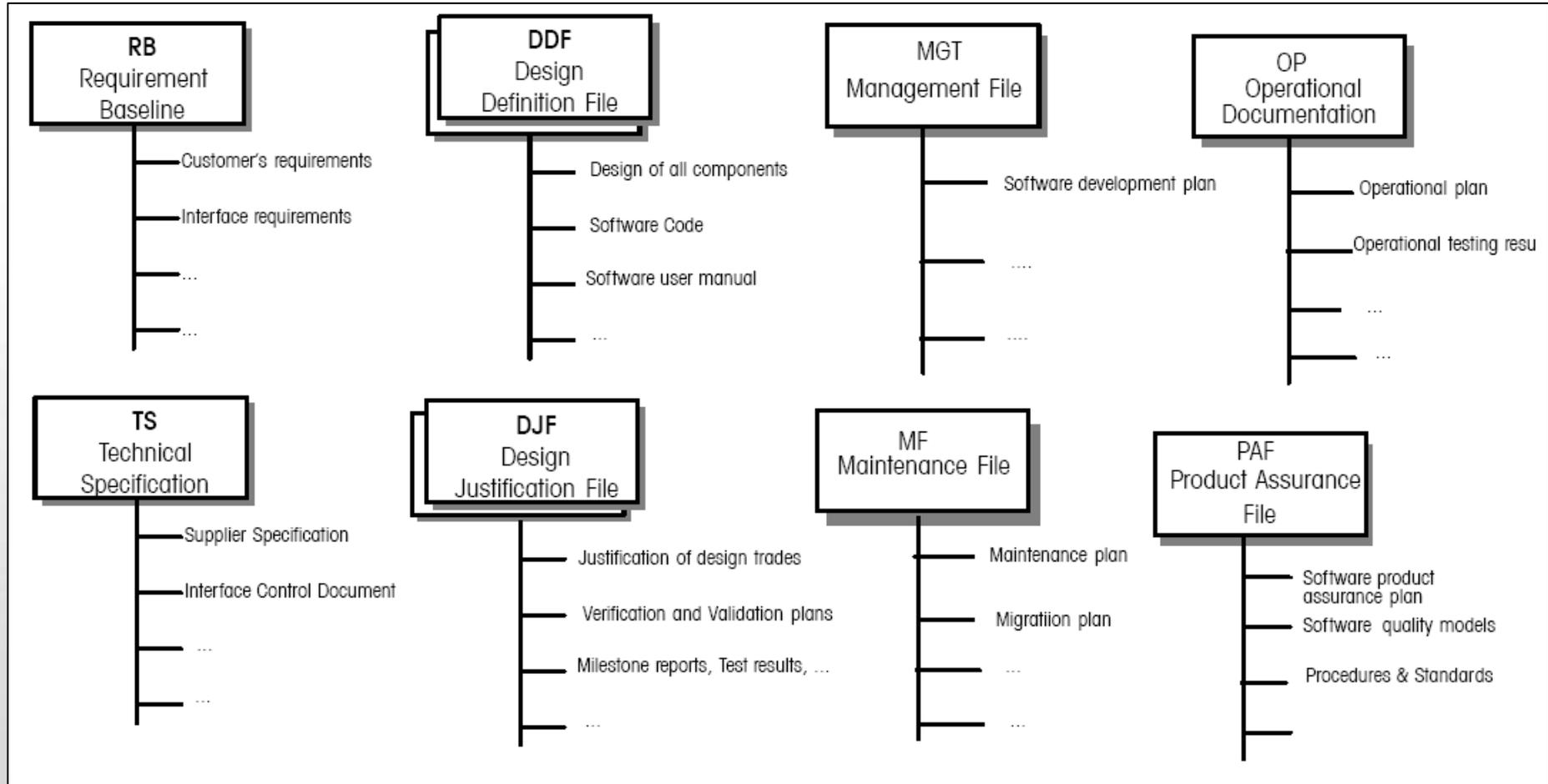
- Requirements Baseline
- Technical Specification
- Design Definition File
- Design Justification File
- Management File
- Maintenance File
- Operational Documentation
- Product Assurance File

European Cooperation for Space Standardization



European Cooperation for Space Standardization





Alcuni documenti d'esempio

- [System Requirement document example](#)
- [Architectural design document example](#)
- [Test Plan example](#)
- [Test Procedure Example](#)



Rimettiamo a posto i guai

Test del software

Perché test e analisi

Il software non ha garanzia

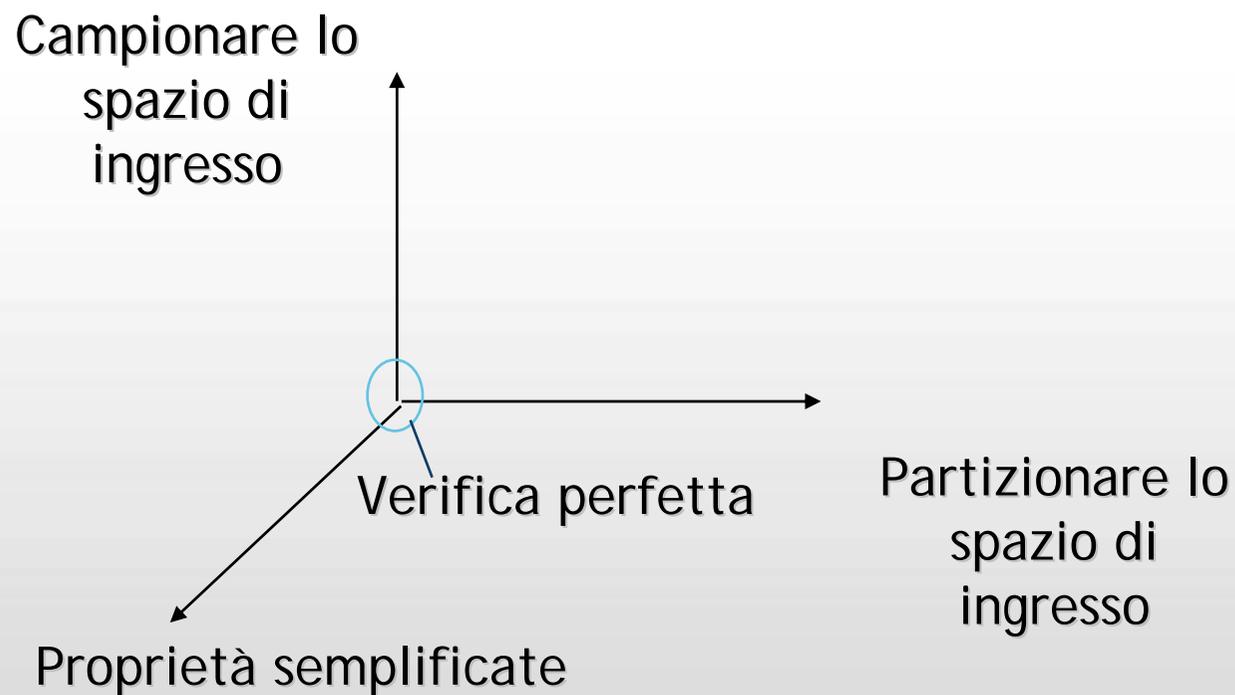
- Il software non è mai corretto a priori
 - Indipendentemente dalla tecnica di sviluppo utilizzata
- Ogni software deve essere verificato
- Test e analisi del software sono
 - Importanti per controllare la qualità del prodotto (e del processo)
 - Costose
 - Difficili ma stimolanti

Cosa può succedere

Terminologia

- Errore
 - Fattore (umano) che causa una deviazione tra il software prodotto e il programma ideale
 - Esempio: errore di analisi dei requisiti, progetto, battitura,...
- Difetto
 - Elemento del programma non corrispondente alle aspettative
 - Esempio: il programma somma contiene un operatore di prodotto anziché un operatore di somma.
- Malfunzionamento
 - Comportamento del codice non conforme alle specifiche.
 - Esempio: il programma somma usa i dati 4 e 3 produce 12.

Cosa possiamo fare



Dobbiamo ammettere un certo livello di inaccuratezza per poter affrontare il problema

Migliorare il prodotto o il processo

Due risultati

- Migliorare il prodotto:
 - Il difetto è
 - Rilevato (a valle di ispezione, test, rapporto degli utenti..)
 - Diagnosticato
 - Riparato
- Migliorare il processo:
 - Il difetto è
 - Rilevato (ed eventualmente riparato)
 - Registrato
 - Analizzato per migliorare il processo

Severità dei difetti

Severity

- Tipica classificazione dei fallimenti (causati dai difetti)
 - Critico: il prodotto non è usabile
 - Severo: Alcune funzionalità del prodotto non possono essere usate, e non c'è modo di sostituirle
 - Moderato: Alcune funzionalità del prodotto non possono essere usate, ma possono essere sostituite (calo di efficienza, affidabilità, convenienza,...)
 - Inconveniente minore

Attività di test

Varie tipologie

- Ispezione di codice
- Test di integrazione
- Test di sistema
- Test di accettazione/robustezza/usabilità
- Documentazione di test e test di regressione
- Controllo modifiche e rimozione difetti
- Controlli di copertura

Vantaggi di un gruppo di test autonomo

Un team dedicato

- Aspetti tecnici
 - Maggior specializzazione
 - Conoscenza delle tecniche
 - Conoscenza degli strumenti
- Aspetti psicologici
 - Distacco dal codice
 - Test svincolato dalla conoscenza del codice
 - Attenzione ad aspetti dimenticati
 - Indipendenza della valutazione
 - Il lavoro è valutato rispetto al numero di difetti rilevati ancora presenti nel codice, non in base alla presunta correttezza iniziale

Svantaggi di un gruppo di test autonomo

Un team dedicato

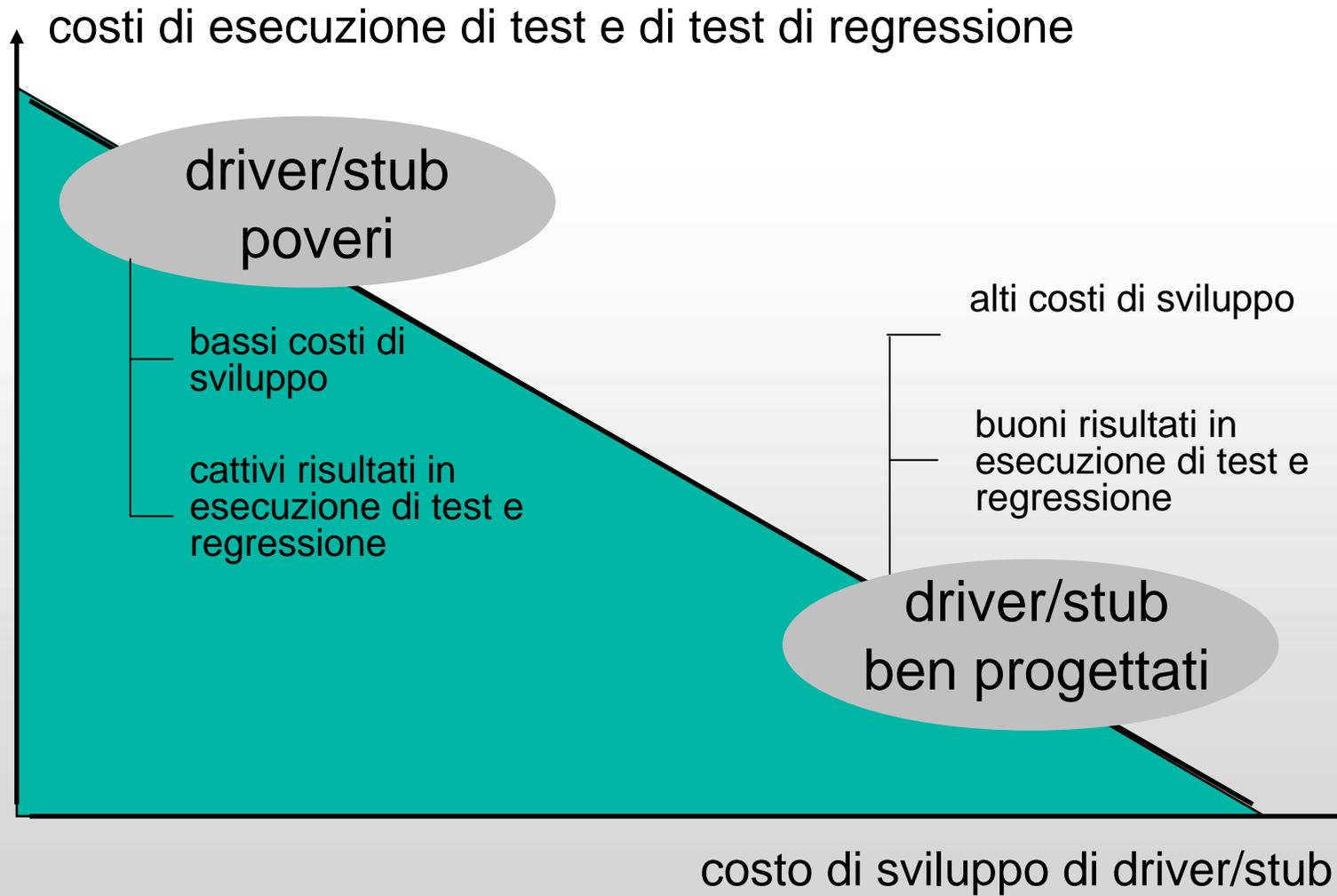
- **Aspetti tecnici**
 - Progressiva perdita di capacità di progetto e codifica
 - Minor conoscenza dei requisiti
- **Aspetti psicologici**
 - Possibile pressione negativa sui progettisti e programmatori
 - Possibile gestione scorretta delle responsabilità
 - Possibile svuotamento dell'importanza dei singoli ruoli

Un buon compromesso

Non fare solo test

- Rotazione del personale
 - Permette di evitare il progressivo depauperamento tecnico dovuto alla eccessiva specializzazione
 - Permette di evitare lo svuotamento dei ruoli
 - Aumenta i costi di formazione
 - Aumenta le difficoltà di pianificazione
- Condivisione del personale
 - Permette di supplire alla scarsa conoscenza del prodotto in esame
 - Aumenta le difficoltà di gestione dei ruoli

Driver e Stub



Documentazione di test

Lista

- Documentazione di insiemi di test (test suite):
 - Software collaudato
 - Versione
 - Obiettivi
 - Risultati globali
 - Autore
- Documentazione di casi di test:
 - Obiettivi
 - Ambiente (driver, stub,)
 - Ingressi
 - Uscite attese
 - Uscite ottenute
 - Risultato
 - Osservazioni

Test di sistema

test di unità	test di integrazione	test di sistema
dalle specifiche dei moduli	dalle specifiche delle interfacce	dalle specifiche dei requisiti
visibilità dei dettagli del codice	visibilità di aspetti di integrazione	nessuna visibilità del codice
richiesti scaffolding complessi	richiesti scaffolding parziali	non sono richiesti driver/stub
attenzione al comportamento del singolo modulo	attenzione alle interazioni tra moduli	attenzione alle funzionalità di sistema

Ambienti di Test

Un passo per volta

- Test di sistema
 - Il software è confrontato con le specifiche dei requisiti (verifica)
 - Di solito eseguito dal fornitore
- Test di accettazione
 - Il software è confrontato con i requisiti dell'utente (convalida)
 - Di solito eseguito dal cliente
- Test di Integrazione
 - componenti testati possono causare problemi a causa di:
 - Cattivo uso di interfacce (cattiva specifica/implementazione di interfacce)
 - Cattive ipotesi sul comportamento dei moduli
 - Uso di driver/stub poveri durante il test di unità



In maniera ordinata

Configuration Management

CM e SCM

- Configuration management (CM): gestione delle versioni e delle configurazioni dei prodotti e dei semilavorati
- È un insieme di tecniche e metodologie
 - applicate allo sviluppo di software da parte di team di analisti, progettisti, sviluppatori
 - che consentono di:
 - identificare (identificare per poter controllare)
 - controllare le modifiche (controllare l'evoluzione)
 - conoscere (registrare e diffondere informazioni)
- Software Configuration Management (SCM): CM applicato al processo di sviluppo software

A che serve

- massimizzare la produttività riducendo gli errori nel trattamento dei semilavorati
- mantenere l'integrità del prodotto durante tutta la sua vita, dalle specifiche al design, allo sviluppo, all'utilizzo
- supportare le attività di sviluppo e manutenzione

Attività di CM

- Configuration identification
 - selezione di ciascun elemento da porre sotto CM
 - identificazione (attribuzione di un identificatore univoco)
- Configuration control
 - definizione delle politiche secondo cui attuare le modifiche
 - controllo e verifica dell'attuazione delle politiche
- Configuration status accounting
 - raccolta e distribuzione di informazioni sullo stato degli elementi posti sotto CM
- Configuration audit and review
 - verifica di conformità (interna ed esterna)

Esempi di identificazione

- Documento di specifica
 - <codice ufficio> - <codice progetto> - <versione> - <stato>
 - <titolo> - <tipo di documento> - <versione> + <stato> - <codice progetto>
 - PROTOCOLLO/DES/0102B/A-001-NTW-BB
 - DISPATCHER/MAN/0100T/R-023-AS1-AA
- Elemento + modifica = elemento in una nuova versione
- Versione
 - stato di un elemento in un istante di tempo
- Configurazione
 - insieme di moduli software, documentazione, ..., utilizzati per costruire un prodotto (famiglia di prodotti = diverse configurazioni)
- Analogie
 - edizioni di un libro
 - versioni di un'auto
 - configurazione di un PC: tipo di processore, quantità di memoria, tipo di hard disk, ...

Configuration management

Change request management

- Tool di Change Request Management (CRM)
 - Strumento che affianca i tool di CM
 - Gestisce le richieste di cambiamento (es. form on-line)
 - Offre adeguati meccanismi di approvazione e classificazione (es. priorità, severità, stato)
 - Tiene traccia dell'iter della modifica (status)
 - Genera report sullo stato delle richieste e sul lavoro svolto

Configuration control board

- Una CCB si riunisce periodicamente per
 - analizzare lo stato delle richieste di modifica
 - coordinare e autorizzare le modifiche da apportare a una baseline per cui è competente

Analisi d'impatto del cambiamento di requisiti

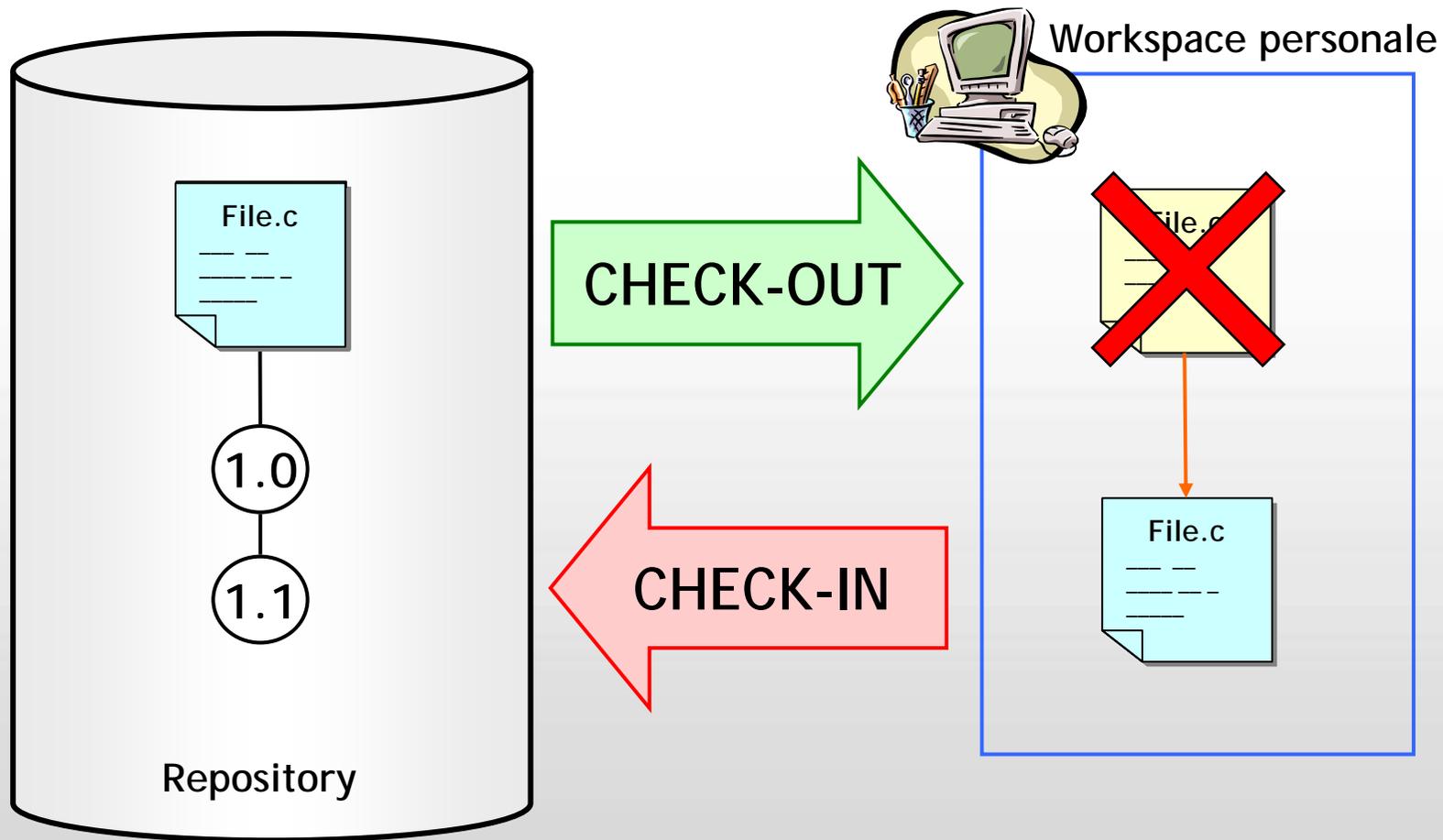
- Un cambiamento nei requisiti può impattare su
 - Fattibilità
 - Altri requisiti esistenti
 - Vincoli tecnologici
 - Tempistica
 - Effort e risorse
 - Gestione dei rischi e degli effetti collaterali
 - Codice
- Alcuni tool di CM supportano (o si integrano con tool che supportano) l'automazione della gestione dei requisiti

Paradigma check-out/check-in

Prendere e rilasciare

- Check-out
 - preleva una copia di una versione di un modulo da un repository e la deposita nello spazio di lavoro
 - criteri per la selezione della versione (ultima, x.y, data, ...)
 - possibilità di porre o meno un lock sulla specifica versione (lock = modifica possibile o meno)
- Modifica della copia nello spazio di lavoro (isolamento)
- Check-in
 - deposita una nuova versione di un modulo nel repository (se consentito, se presente lock, ...)
 - criteri per stabilire identificatore nuova versione (vecchia + 1, manualmente, regole, ...)
- Possibile una stretta serializzazione del lavoro (single writer, multiple readers)

Esempio: check-out/check-in



- N.B.: *commit* è sinonimo di check-in

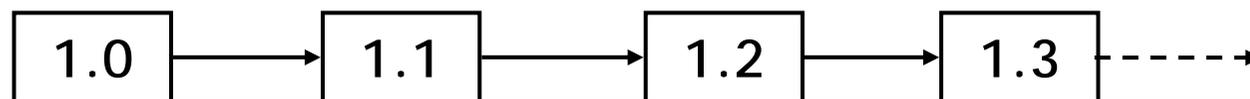
Politiche di locking

lock-modify-unlock

- **Lock esclusivo:**
 - al momento del check-out viene posto un lock sul file
 - non è ammesso il check-out di file su cui è già posto un lock
 - solo un programmatore per volta può modificare il file
 - il lock viene automaticamente rilasciato al momento del check-in del file
- **Reserved/unreserved check-out**
 - reserved: chi ha operato il check-out del file sarà il primo a poterne fare il check-in
 - unreserved: non c'è garanzia che chi ha operato il check-out del file sia il primo a poterne fare il check-in
 - dello stesso file possono esserci contemporaneamente più unreserved check-out ma un solo reserved check-out
 - un reserved check-out può essere cancellato o rilasciato (diventando unreserved)

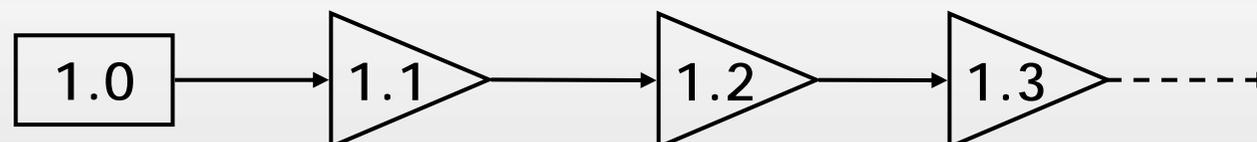
Memorizzazione delle versioni

- *Completa*: ogni versione di un modulo in un file separato

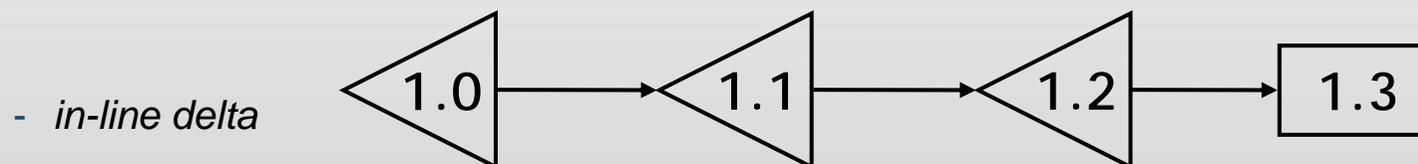


- *Delta*: memorizzare in forma completa solo una versione; di tutte le altre versioni si conservano solo le differenze

- *forward delta*: prima versione e differenze rispetto alle successive



- *backward delta*: versione più recente e differenze rispetto alle precedenti



- *in-line delta*

Panoramica sui tool di CM

Nome	Vendor	Sito	Licenza
RCS	-	www.gnu.org/software/rcs	Open Source
CVS	-	www.cvshome.org	Open Source
ClearCase	Rational	www.rational.com/products/clearcase	Commerciale
SourceSafe	Microsoft	msdn.microsoft.com/ssafe	Commerciale
PVCS	Merant	www.merant.com/PVCS	Commerciale
Perforce	Perforce Software	www.perforce.com/perforce	Commerciale
BitKeeper	BitMover	www.bitkeeper.com	Commerciale
AllFusion Harvest	Computer Associates	ca.com/products	Commerciale
Source Integrity	MKS	www.mks.com/products	Commerciale
CM Synergy	Telelogic	www.telelogic.com/products/synergy	Commerciale
AllChange	Intasoft	www.intasoft.co.uk	Commerciale
StarTeam	Starbase	www.starbase.com	Commerciale
TRUEchange	McCabe & Associates	www.truesoft.com	Commerciale
ChangeMan	Serena	www.serena.com/product	Commerciale
Aegis	-	aegis.sourceforge.net	Open Source
PRCS	-	prcs.sourceforge.net	Open Source

Politica di locking in CVS

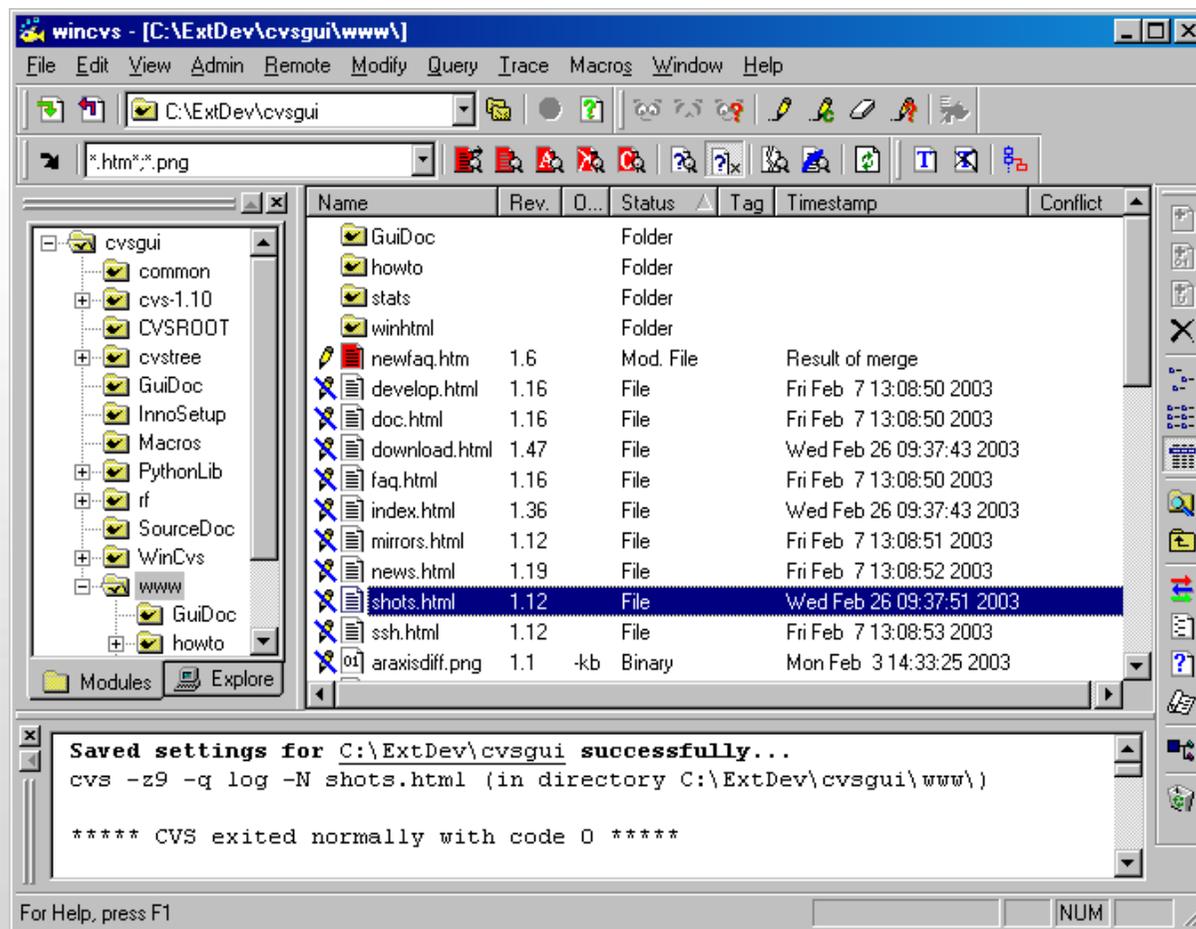
- CVS consente modifiche simultanee allo stesso modulo (paradigma copy-modify-merge)
 - più persone possono fare check-out (no locking)
 - CVS verifica le inconsistenze all'atto del check-in e impone di risolverle (semi-automaticamente)
- Volendo, CVS può supportare il locking
- CVS supporta anche un meccanismo (watches) per sapere chi sta editando un certo file
- In generale, CVS permette di generare notifiche e compiere azioni in base al verificarsi di determinati eventi (es. commit, check-out, update, tag,...)

Cosa CVS non è

- Non è un sistema di build
- Non sostituisce l'attività di management del progetto
- Non sostituisce la comunicazione tra gli sviluppatori
- Non supporta il change control (es. bug-tracking)
- Non è un tool per l'automazione del test
- Non ha un modello di processo

Front-end per CVS

- Interfacce grafiche per varie piattaforme
 - WinCVS
 - SmartCVS
 - MacCVS
 - TkCVS
 - gCVS
 - jCVS
 - Cervisia
- Plug-in per tool di sviluppo e altri ambienti
 - CvsIn
 - TortoiseCVS





Fine V lezione Sviluppo sistemi

GIS e Geo WEB: piattaforme e architetture