Modelli di processo per lo sviluppo del software: i modelli orientati alla qualità

Prof. Paolo Ciancarini
Corso di Ingegneria del Software
CdL Informatica
Università di Bologna

Obiettivi di questa lezione

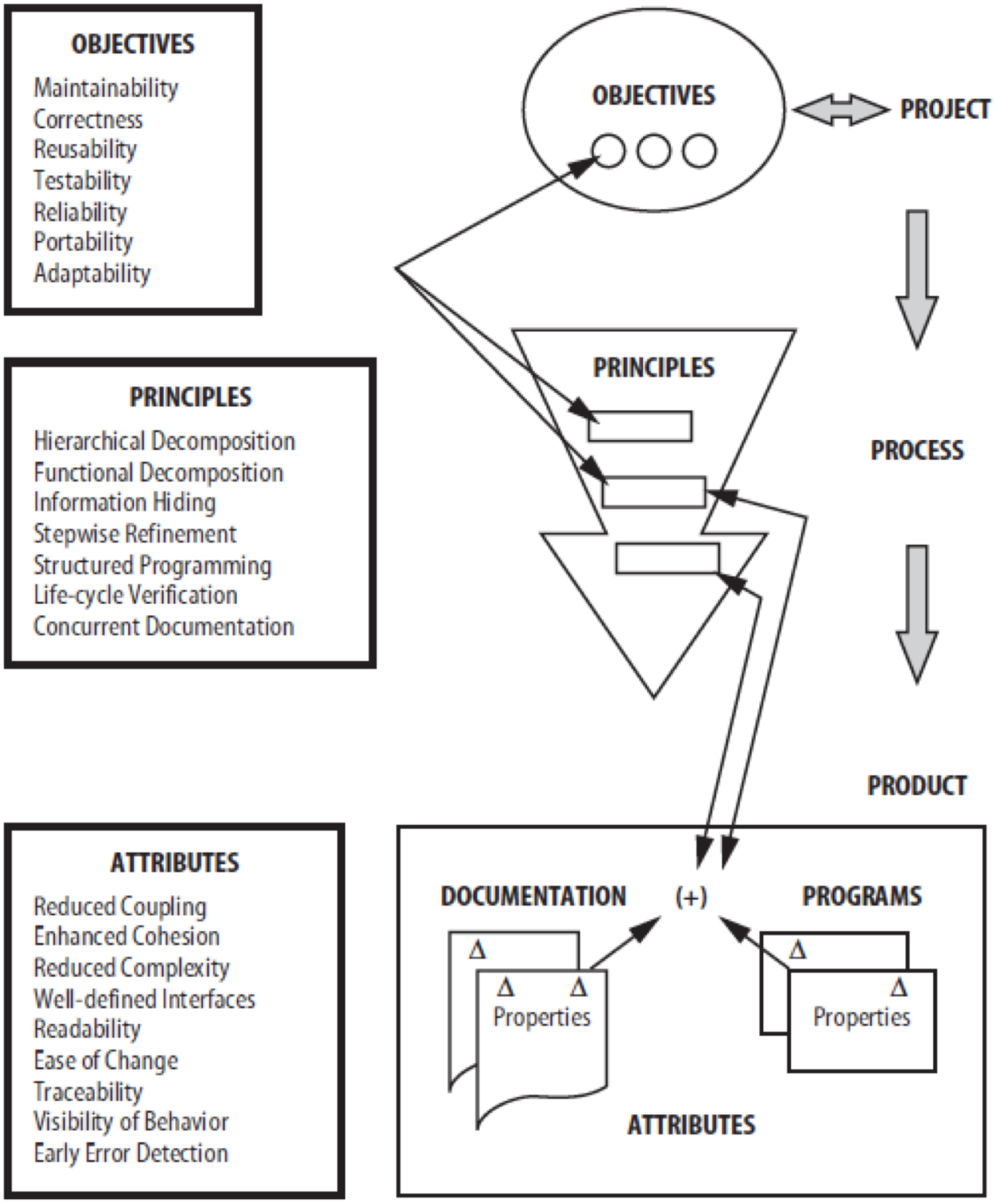
- Indicatori di qualità di processo
- I modelli di processo orientati alla qualità
- I modelli orientati allo sviluppo open source

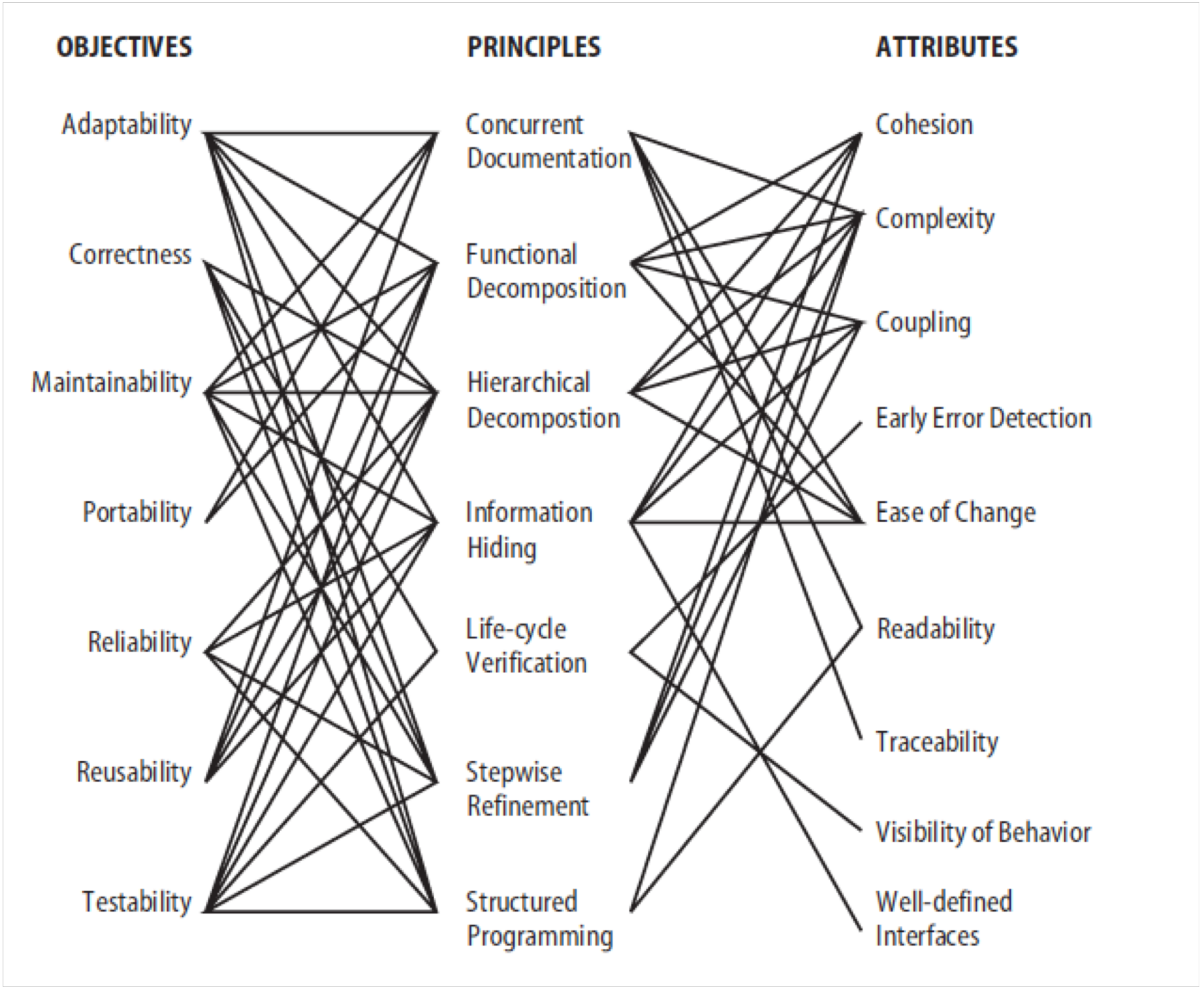
La qualità del software

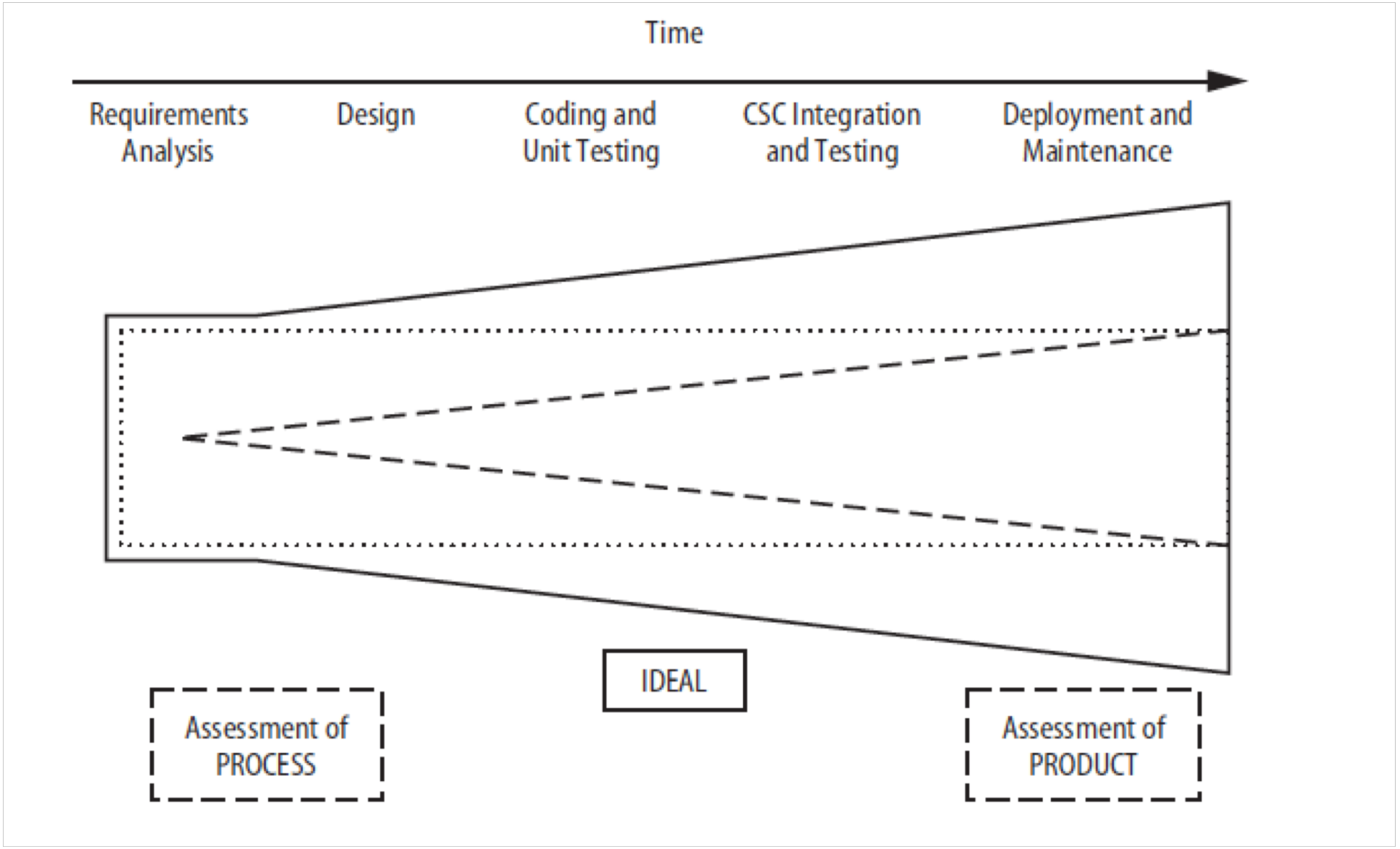
Che cos'è la qualità del software?

- Il software soddisfa tutti i requisiti?
- È stato prodotto nei tempi previsti?
- È stato prodotto col budget previsto?
- Ha < 1 difetto/10KSLOC?
- Ha MTBF > 10.000 ore?

Senza informazioni aggiuntive questa domanda non ha risposta


OBJECTIVES
Maintainability
Correctness
Reusability
Testability
Reliability
Portability
Adaptability
PRINCIPLES
Hierarchical Decomposition
Functional Decomposition
Information Hiding
Stepwise Refinement
Structured Programming
Life-cycle Verification
Concurrent Documentation
ATTRIBUTES
Reduced Coupling
Enhanced Cohesion
Reduced Complexity
Well-defined Interfaces
Readability
Ease of Change
Traceability
Visibility of Behavior
Early Error Detection
OBJECTIVES
PROJECT
PRINCIPLES
PROCESS
PRODUCT
DOCUMENTATION
(+)
PROGRAMS
Δ
Δ Δ
Properties
Δ
Δ
Properties
ATTRIBUTES


OBJECTIVES
PRINCIPLES
ATTRIBUTES
Adaptability
Correctness
Maintainability
Portability
Reliability
Reusability
Testability
Concurrent Documentation
Functional Decomposition
Hierarchical Decompostion
Information Hiding
Life-cycle Verification
Stepwise Refinement
Structured Programming
Cohesion
Complexity
Coupling
Early Error Detection
Ease of Change
Readability
Traceability
Visibility of Behavior
Well-defined Interfaces



Modelli di processo 5

Indicatori di qualità dei processi (a livello aziendale)

- Percentuale dei processi completati a +/- 5% della data stimata
- Tempo medio di ritardo dei processi attivi
- Percentuale di processi in ritardo sulla loro data finale
- Età media di processo
- Percentuale di processi in cui il numero di risorse assegnate è minore del numero pianificato
- Somma dei costi dei processi cancellati
- Tempo medio per completare un compito
- Media in giorni di ritardo ti tutti i progetti attivi

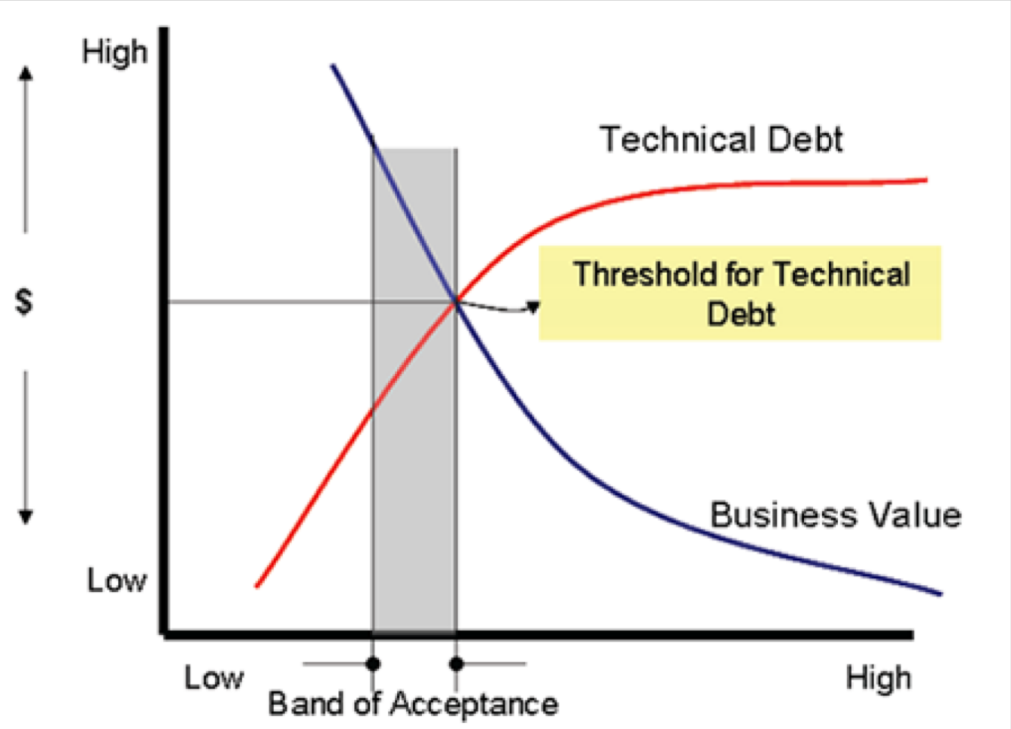
Indicatori di processo di sviluppo software: esempi

- Produttività: numero di LOC per unità di tempo/persona
- Numero di difetti per rilascio o sprint
- Numero di difetti rimossi da un rilascio in una unità di tempo
- Debito tecnico: effort risparmiato rimandando al futuro necessarie fasi di refactoring o correzioni di errori
- Riuso_futuro: numero di LOC progettate per essere riusate in progetti successivi o prodotti analoghi
- Riuso_passato: numero di LOC utilmente riusate da progetti precedenti

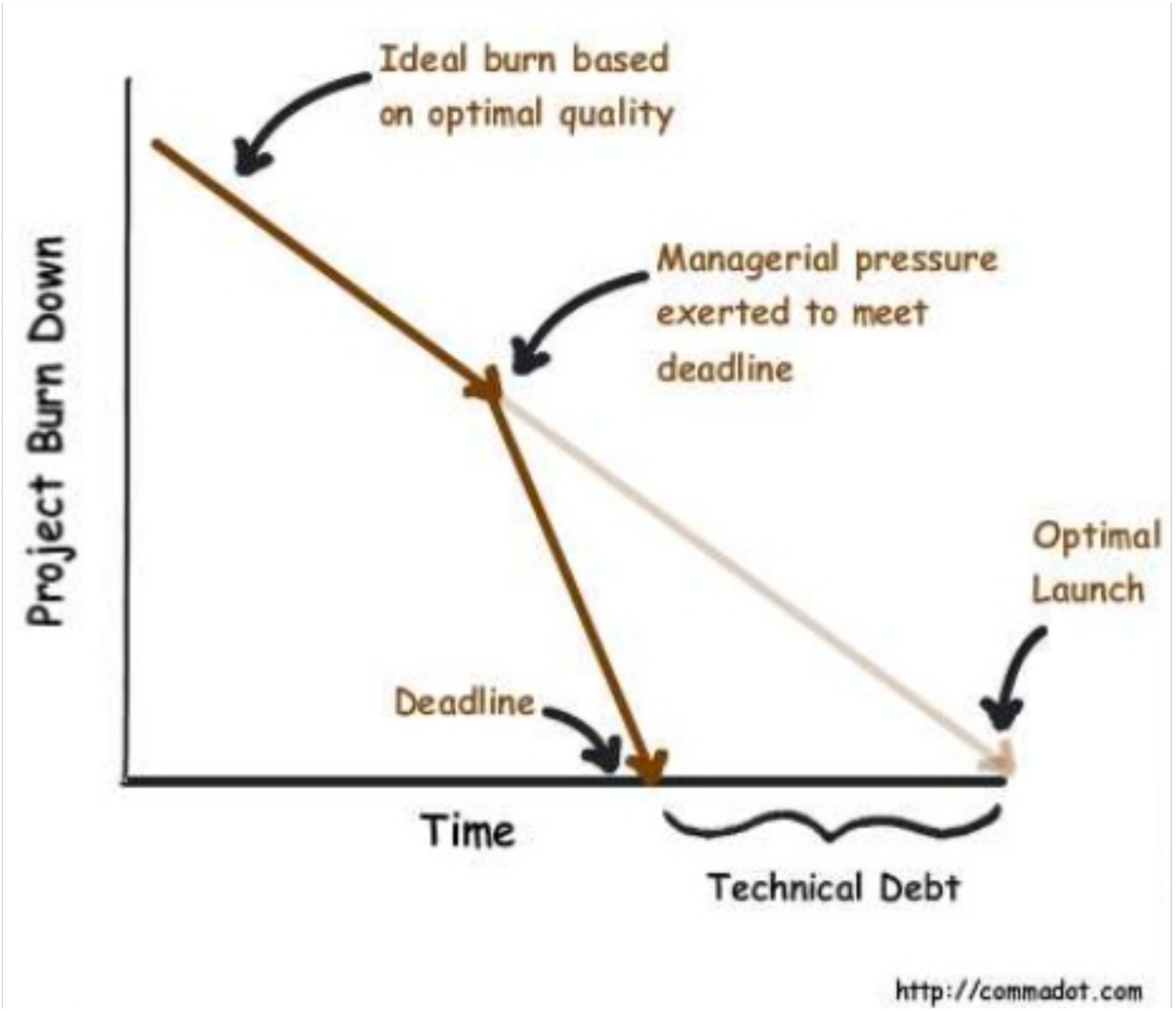
Indicatore agile: debito tecnico

Il debito tecnico è una metafora per descrivere le possibili complicazioni che subentrano in un progetto di sviluppo software, qualora non vengano adottate adeguate azioni volte a mantenerne bassa la complessità

Analogamente a quanto avviene nel mondo finanziario, in cui per sanare un debito occorre pagare gli interessi, lo sforzo per recuperare un progetto eseguito male può aumentare anche considerevolmente nel tempo, se non si interviene tempestivamente



Debito tecnico



SonarQube: uno strumento per misurare il debito tecnico

Modelli orientati alla qualità

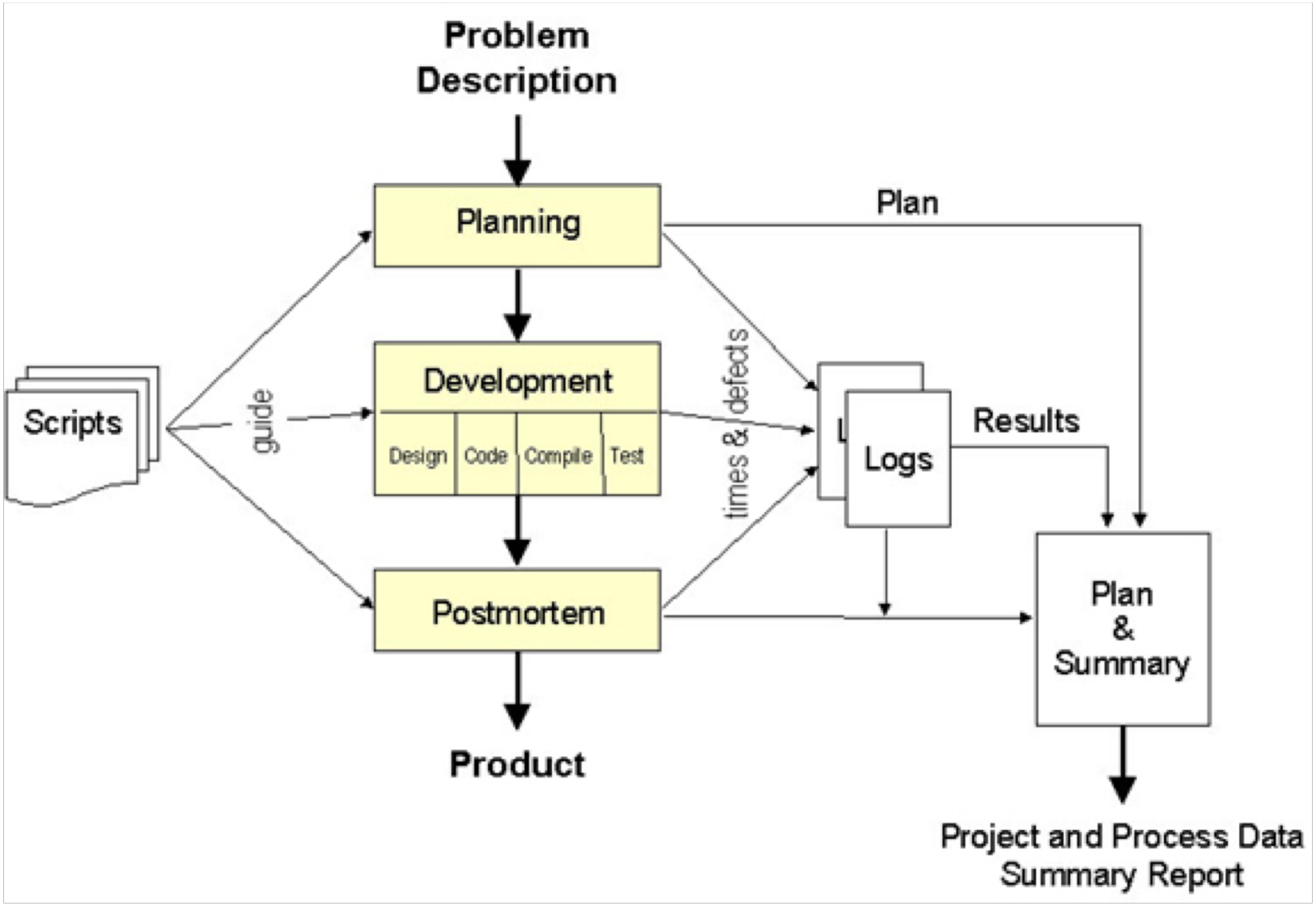
I modelli di processo orientati alla **qualità** sono sempre pianificati (= non agili) e si basano sulla misurazione sistematica di alcuni indicatori di qualità di processo

- PSP (personal sw process) e TSP (team sw process)
- ISO 9000 e CMM (Capability Maturity Model)

Personal Software Process (PSP)

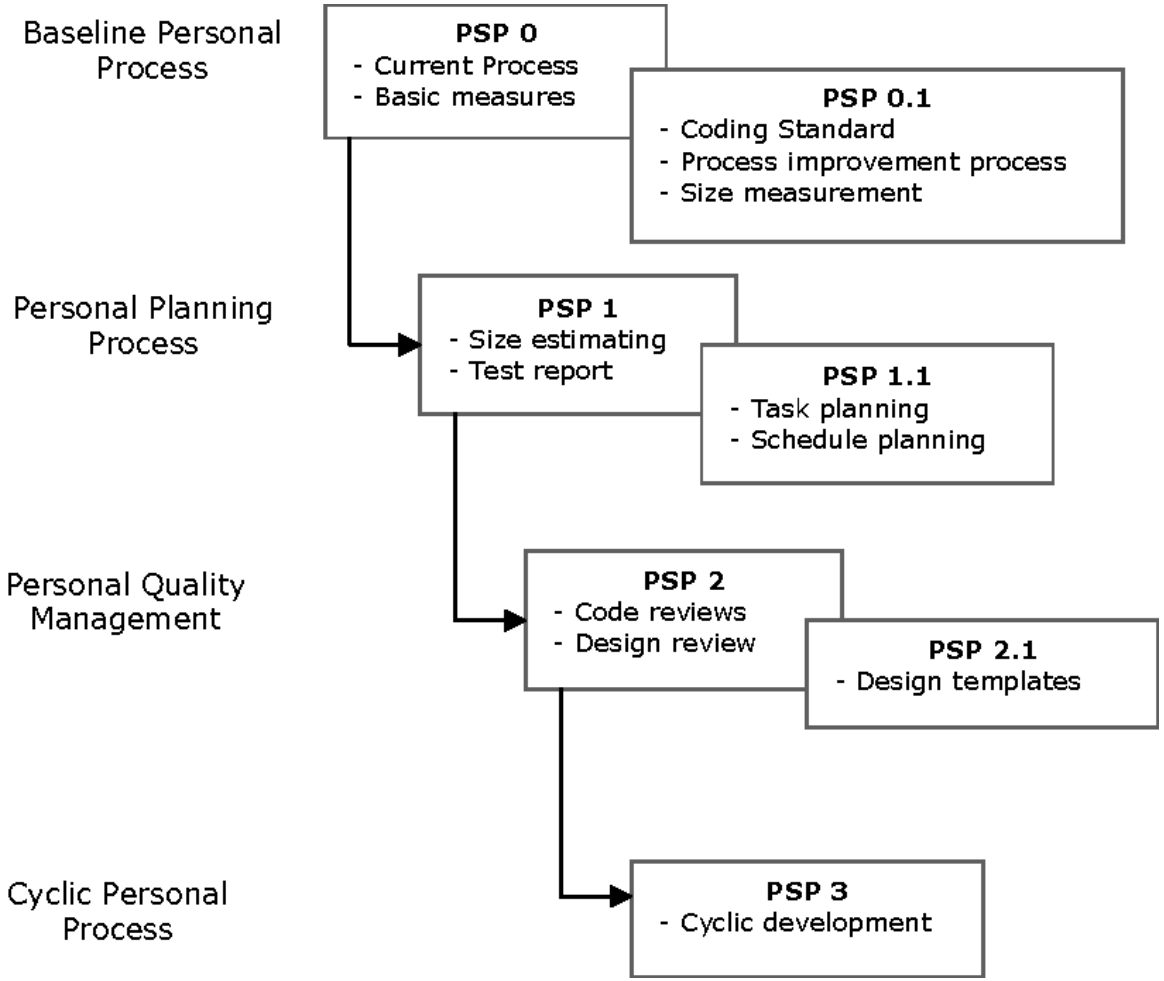
- Il PSP si basa sulla misurazione sistematica delle attività di sviluppo allo scopo di migliorare la produttività del singolo sviluppatore
- Alcuni parametri tipici da misurare: Size, Effort, Quality, Schedule
- PSP: famiglia di modelli *in-the-small*
- I modelli PSP scalano *in-the-many* (a livello di gruppo di persone) nei modelli di processo Team Software Process (TSP)

Esempio: PSP



www8.cs.umu.se/~jubo/UPSP/

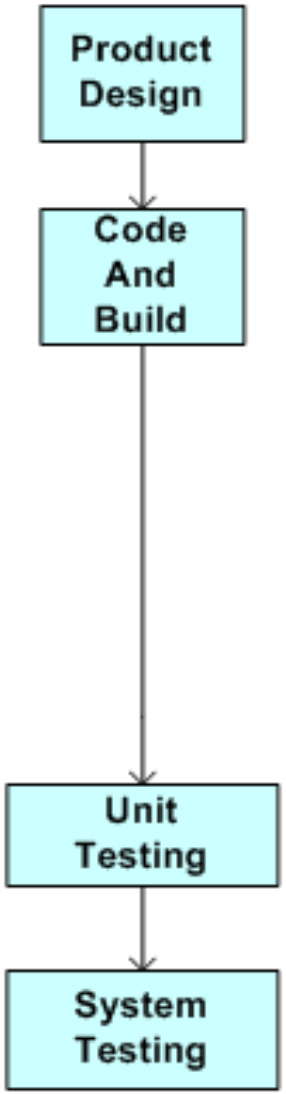
La famiglia PSP



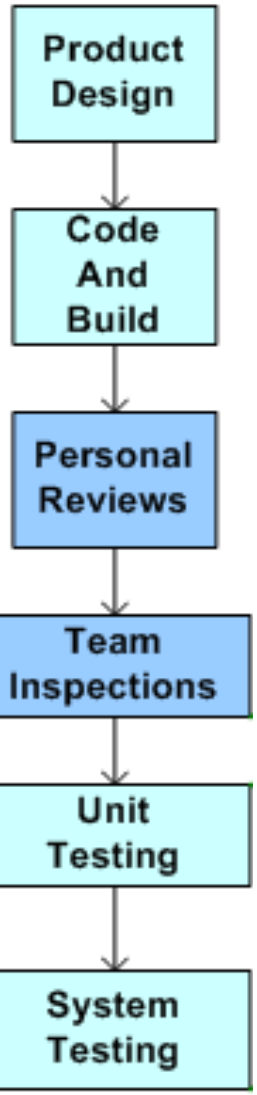
Principi di PSP

- Ogni sviluppatore è diverso; per essere efficaci, gli sviluppatori debbono pianificare il lavoro e basare i piani su dati personali.
- Per migliorare sistematicamente il proprio processo, gli sviluppatori debbono usare personalmente processi ben definiti e misurabili.
- Per costruire prodotti di qualità, gli sviluppatori debbono sentirsi personalmente responsabili della qualità dei propri prodotti. I prodotti di qualità supeiore non si ottengono per errore; gli sviluppatori debbono sforzarsi di lavorare con buona qualità.
- In un processo di sviluppo costa meno prima che dopo trovare e aggiustare i difetti.
- È più efficiente prevenire i difetti piuttosto che cercarli e aggiustarli.
- Il modo giusto di fare un lavoro è sempre il più veloce ed economico

Team Software Process



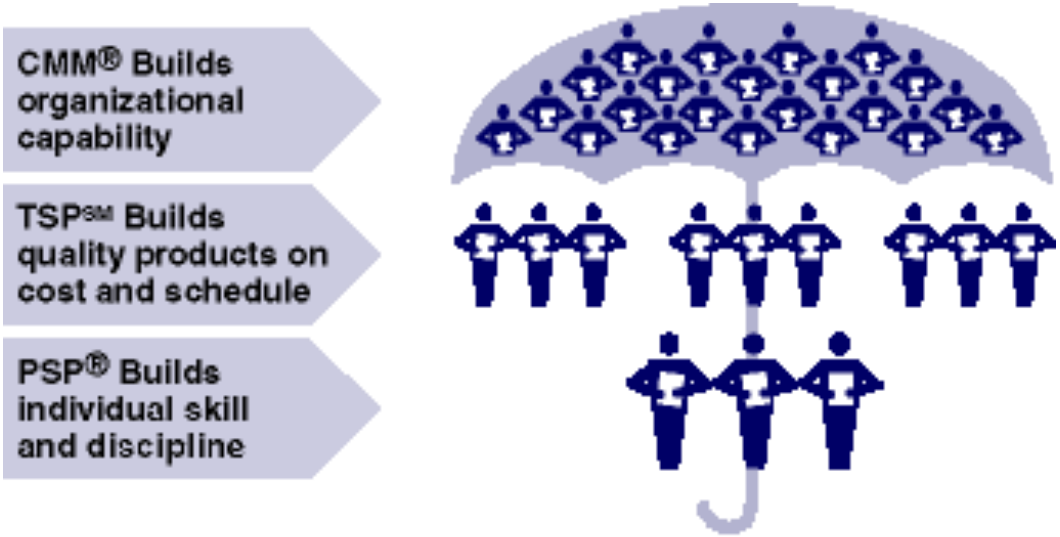
Today's Typical SW Quality Process



The TSP SW Quality Process

www.sei.cmu.edu/tsp/

PSP-TSP-CMM



CMM focus: management

TSP focus: team performance

PSP focus: individual discipline

Capability Maturity Model for Software (SW-CMM©)

Livello	**Caratteristiche**	**Principali strategie di miglioramento del processo**
Ottimizzante	Migliorare il feedback al processo	Identificare indicatori di processo
Gestito	(Quantitativo) Processo misurato	Raccolta automatica di dati di processo per analizzarlo e modificarlo
Definito	(Qualitativo) Processo definito e istituzionalizzato	Misurazione del processo Analisi del processo Piani di qualità quantitativi
Ripetibile	(Intuitivo) Processo dipendente dagli individui	Creare un Gruppo di Processo Identificare un'architettura di processo Introdurri metodi e strumenti di Sw Eng
Iniziale	Ad hoc/ Caotico No stima dei costi, pianificazione, o gestione	Project management Pianificazione del progetto Controllo di qualità del software

http://www.sei.cmu.edu/cmm/

ISO 9000

- ISO9000-3 è ISO9001 per le fabbriche del sw

ISO 9000	Quality management and quality assurance standards; guidelines for selection and use
ISO 9001	Quality systems model for quality assurance in design, development, production, installation, and servicing
ISO 9002	Quality systems model for quality assurance in production and installation
ISO 9003	Quality systems model for quality assurance in final inspection and test
ISO 9004	Quality management and quality systems elements. Guidelines

Lo sviluppo del software Open Source

- Ridistribuzione libera del sorgente
- Codice sorgente sempre incluso nella distribuzione
- Altri software derivabili con licenza
- Protezione dell'integrità del sorgente dell'autore
- Nessuna discriminazione contro persone o gruppi
- Nessuna discriminazione contro campi applicativi
- Distribuzione della licenza, che si applica a tutti
- Licenza non specifica di prodotto
- Licenza che non vincola altri prodotti sw
- Licenza tecnologicamente neutrale

`www.opensource.org/docs/definition.php`

La filosofia Open Source

- Ogni buon prodotto sw inizia da un problema personale
- I bravi programmatori sanno cosa scrivere. I migliori sanno cosa riscrivere
- Quando hai perso interesse in un programma che hai costruito, è tuo dovere passare le consegne ad un successore competente
- Trattare gli utenti come sviluppatori è la strada migliore per debugging efficace e rapidi miglioramenti del codice
- Distribuisci presto e spesso; e presta ascolto agli utenti
- Stabilita una base di betatester e cosviluppatori sufficientemente ampia, ogni problema verrà presto individuato e qualcuno troverà la soluzione adeguata

E. Raymond, The Cathedral and the Bazaar, O'Reilly, 1997

Famosi Progetti Open Source

- Linux, OpenBSD, FreeBSD, NetBSD
- Progetto Apache. HTTP server, ANT, Tomcat
- PHP, Perl
- Progetto GNU. GCC, GDB, EMACS, GNOME
- JBOSS
- Vim
- Postfix, SpamAssassin, Clam AntiVirus
- …e molti altri

Alcuni repository di prodotti software openvsource:

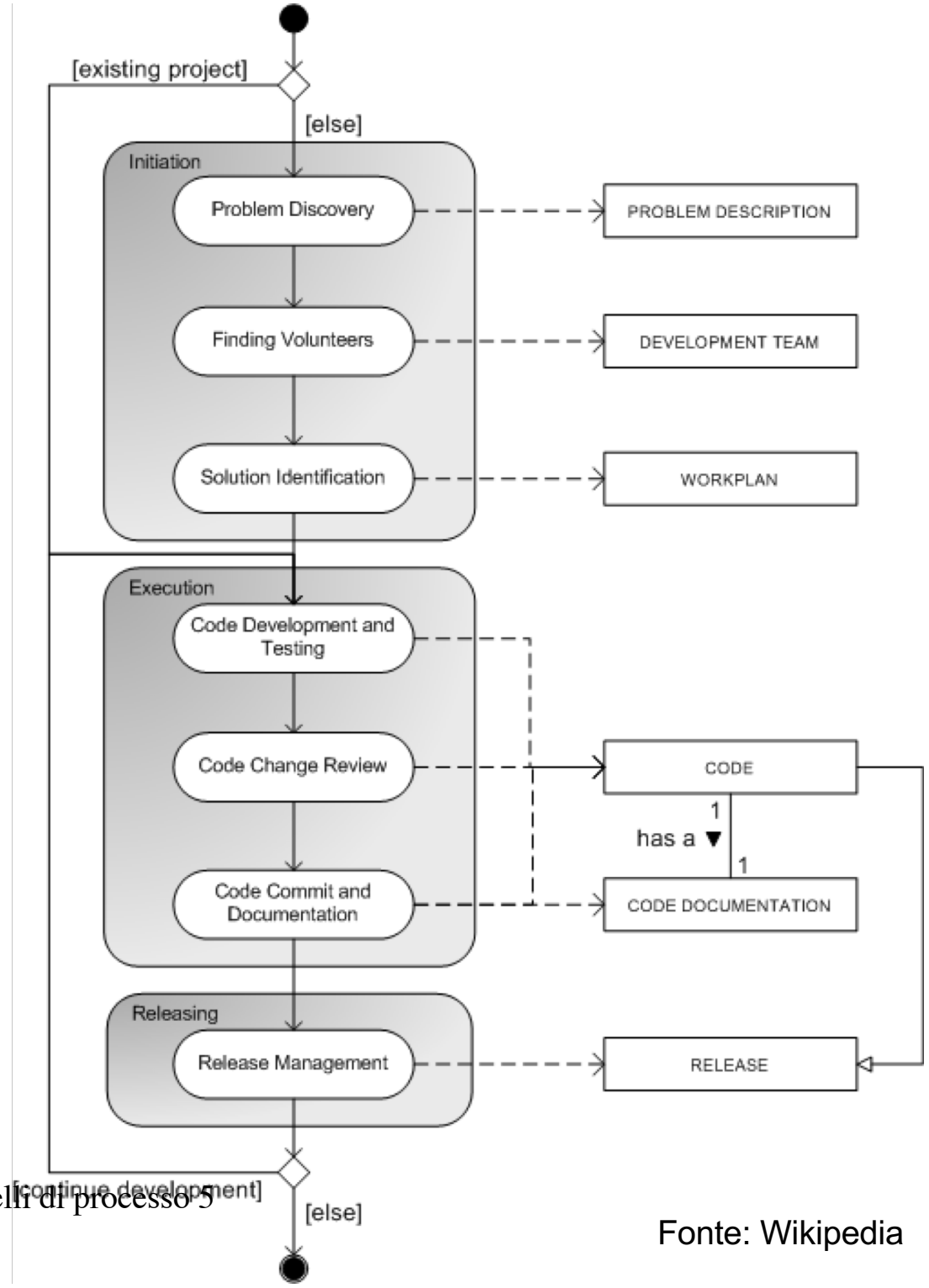
```
https://github.com/
http://sourceforge.net/
```

Il processo di sviluppo del sw open source

- Definizione del problema (problem discovery)
- Ricerca di volontari
- Identificazione della soluzione
- Sviluppo del codice e testing
- Code change review
- Code commit and documentation
- Release management

A Framework for creating hybrid-open source software communities. Srinarayan Sharma et. al. Info Systems (2002)

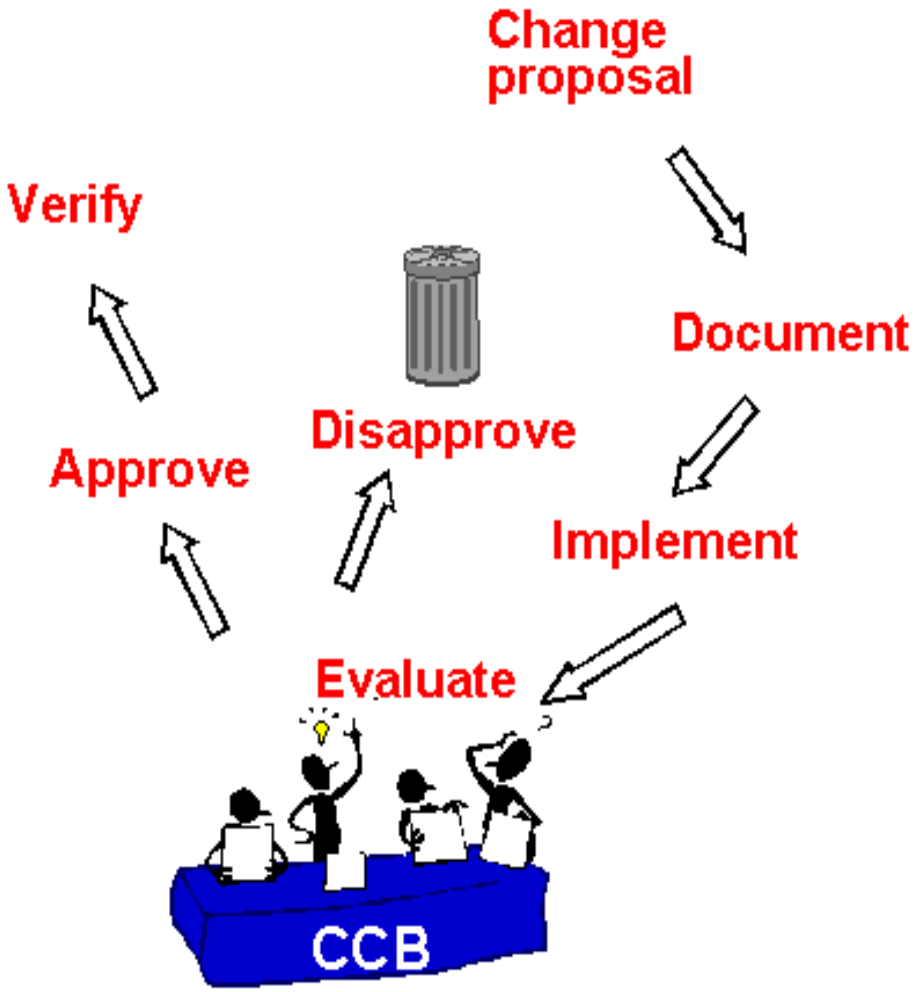
Processo open source



Fonte: Wikipedia

Change control board per OSS

- Il processo è ”pubblico”
- Le implementazioni sono controllate da un board (CCB) che revisiona e testa il codice proposto
- modifiche moderate
- build frequenti
- proprietà collettiva
- ”no maintenance”



Change process for OSS

Reputation-driven

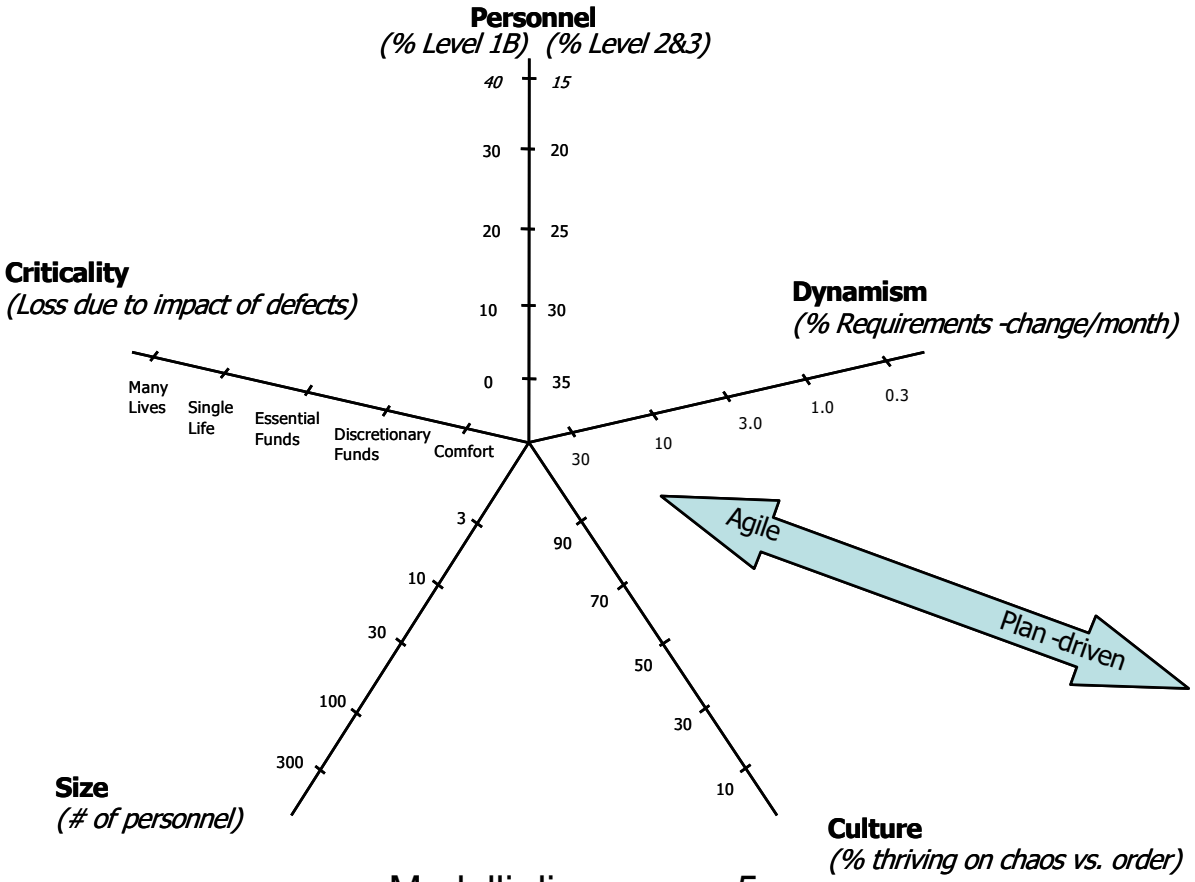
- I sistemi OS sono quasi sempre costuiti da piccoli gruppi di sviluppatori volontari
- Siccome non sono pagati, qual è l'incentivo?
- "*The 'utility function' Linux hackers are maximizing is not classically economic, but is the intangible of their own ego satisfaction and reputation among other hackers*."
  Raymond, La cattedrale e il bazaar, 1999

Confrontare i modelli

CMM	RUP	XP
Cos'è importante?		
Stabilità	Adattabilità	Flessibilità
Ripetibilità	Precisione	Agilità
Perché standardizzare?		
Predicibilità	Componentistica	Velocità
Efficienza	Integrazione	Semplicità
La qualità come viene fuori?		
Rimozione difetti	API predefinite	Integrazione continua
Prevenzione difetti	Integrazione semplice	Presenza cliente
Progetto per qualità	Gestione dei rischi	

Agile vs pianificati: fattori per la scelta

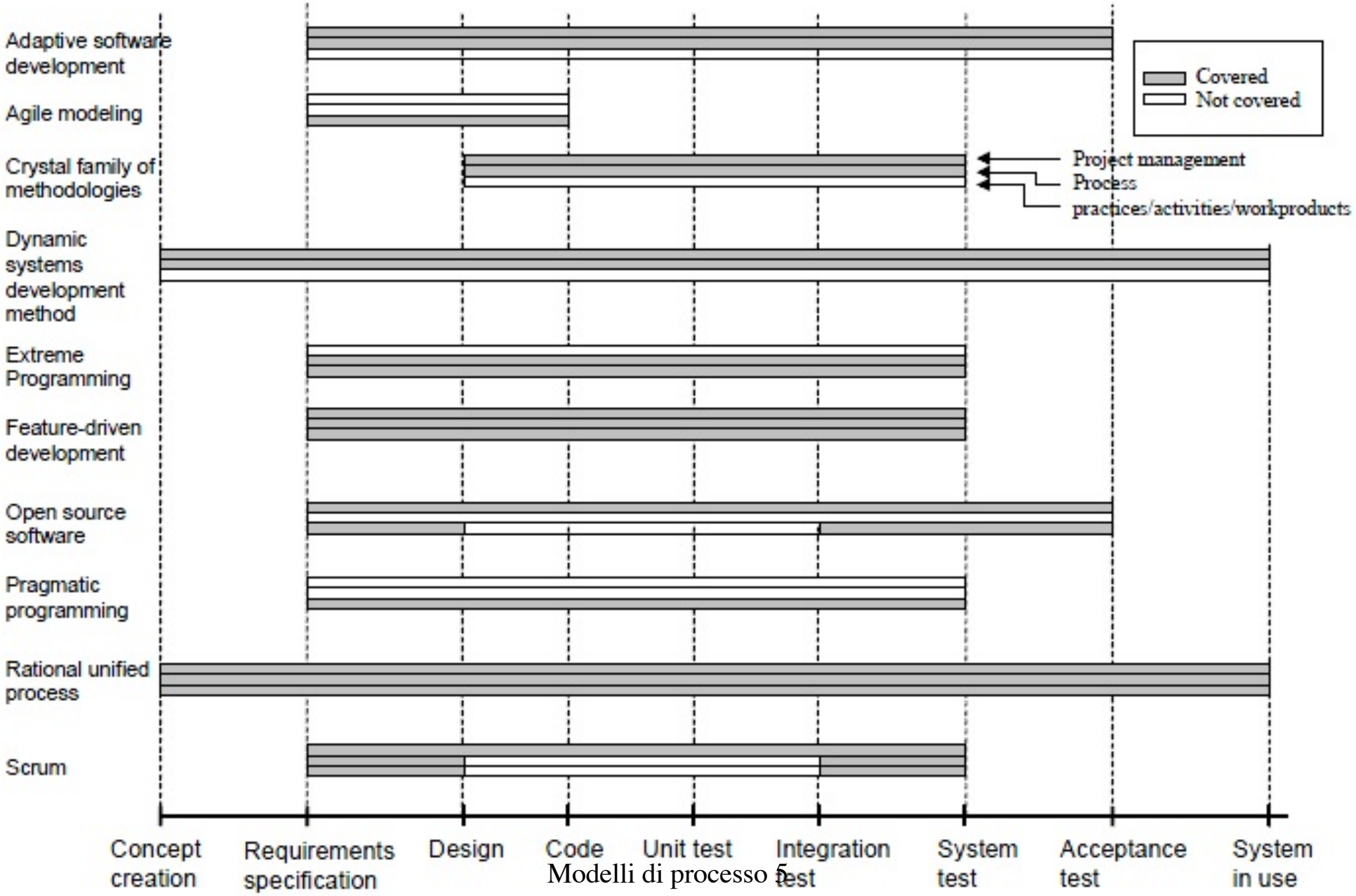
Dimensione, Criticalità, Dinamismo, Personale, Cultura



Livelli di personale (Boehm e Turner)

Livello	Criterio
-1	Non collaborativo
1B	Buon lavoratore poca esperienza necessita di guida
1A	Buon lavoratore esperienza limitata accetta guida
2	Funziona bene in piccoli team in progetti con precedenti
3	Funziona bene in grandi team in progetti senza precedenti

Confrontare i modelli



Conclusioni

- No “Silver bullets” nel processo di sviluppo
- Processi waterfall: pianificati, rigidi
- Processi iterativi: pianificati, flessibili
- Processi agili: non pianificati, test-driven
- Processi per la qualità: misurati
- Processi open-source: reputation-driven

- Ogni organizzazione dovrebbe esplorare più modelli e scegliere quelli più efficaci

Riferimenti

- Nance, Managing Software Quality, 1990
- Galin, Software Quality, IEEE 2018

Siti

- Personal Sw Process e Team Sw Process (SEI) `www.sei.cmu.edu/tsp/introducing.html`
- `www8.cs.umu.se/~jubo/UPSP/`
- Tool PSP e TSP `www.processdash.com`
- Linux open source `www.kernel.org`

Domande?


DID I REMEMBER TO TELL YOU BEFORE YOU FINISHED THE CODING THAT THE USER'S SPECIFICATIONS CHANGED?
Dilbert.com DilbertCartoonist@gmail.com
AAAIII-YIIII-YIIII-YIIIIII!!!!!!
5-16-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick.
SO, NO—ISH?
BAM! BAM! BAM!