



Modelli e Metodi per la Qualità del Software (ITPS)

A.A. 2025 - 2026

Prof.ssa Baldassarre

mariateresa.baldassarre@uniba.it

Presentazione

Dipartimento di Informatica - Università degli Studi di Bari

Via Orabona, 4 - 70125 - Bari

Tel: +39.080.5443270 | Fax: +39.080.5442536

serlab.di.uniba.it



Introduzione

Dipartimento di Informatica - Università degli Studi di Bari
Via Orabona, 4 - 70125 - Bari
Tel: +39.080.5443270 | Fax: +39.080.5442536
serlab.di.uniba.it



You lot start coding...
...I'll go and see what they want.

<http://www.abberley.co.uk/asap/>

Obiettivo Formativo

⇒ Costituire **conoscenze** ed **abilità** per il **Management della Ingegneria del Software**.

⇒ Dove:

- ❑ **Conoscenze** sono le tecniche i metodi ed i principi alla base della Cultura della Qualità nel Software
- ❑ **Abilità** sono le pratiche per usare adeguatamente le conoscenze onde trasformare in **tecnologia** i loro contenuti.
- ❑ **Management della Ingegneria del Software** è il **governo** della qualità dei processi e dei prodotti per raggiungere l'obiettivo strategico della Ingegneria del Software: **Sviluppare software senza sorprese ottimizzando i costi**



Il Corso

Dipartimento di Informatica - Università degli Studi di Bari
Via Orabona, 4 - 70125 - Bari
Tel: +39.080.5443270 | Fax: +39.080.5442536
serlab.di.uniba.it

Obiettivi

⇒ Conoscenze:

- ❑ Modelli di Processo di sviluppo del software
- ❑ Qualità dei processi , Qualità dei prodotti e loro relazione
- ❑ Controllo della qualità
- ❑ La qualità come strumento d'impresa, dalla certificazione al rafforzamento competitivo: ISO, CMMI
- ❑ Gestione di un progetto e delle risorse

⇒ Abilità:

- ❑ Modellare, Verificare e Validare processi
- ❑ Costruire modelli di qualità e piani metrici
- ❑ Monitorare progetti sulla base di un modello di qualità per individuare le iniziative di miglioramento continuo.

Motivazioni

- ⇒ Il corso è mirato a
 - ❑ La qualità del software
 - ❑ Il miglioramento continuo della qualità del software
 - ❑ Elementi di Project Management
- ⇒ La qualità del software ha due componenti:
 - ❑ La qualità dei processi software
 - ❑ La qualità dei prodotti software
- ⇒ Le due componenti della qualità hanno una forte interrelazione tra loro

Prerequisiti del Corso

- ⇒ Principi di produzione e manutenzione del software.
- ⇒ Tecnologie di base per la produzione e la manutenzione del software
- ⇒ **INGEGNERIA DEL SOFTWARE**

Argomenti

- ⇒ **Processi**: rappresentazione di un processo, Verifica e validazione dei processi, diversità dei processi, dal modello di processo al piano esecutivo.
- ⇒ **Qualità**: qualità di processo e qualità del prodotto, Metriche e misure, schemi per la qualità (ISO 9000, CMMI, GQM), technical debt, code review, GenAI nella produzione del sw
- ⇒ **Produttività**: team building, team work, personality traits, trustworthy AI, ruolo degli LLM nella gestione della qualità
- ⇒ **Strumenti per gestire la qualità**: Sonarcloud, Fortify, Git, SVN, Redmine
- ⇒ **Miglioramento dei processi e dei prodotti**: scopo e schemi per il miglioramento dei processi (ISO 9000, TQM, PDCA, QIP).
- ⇒ **Pratiche per la qualità ed il miglioramento**: strumenti per la valutazione della qualità di processi/prodotti; strumenti per la interpretazione corretta e completa delle misure;
- ⇒ **Concetti di Project Management**: elementi di base di project management, scheduling, quality control, Pert, GANTT, CPM, Gestione del Rischio

LIBRI DI TESTO

- ⇒ PRINCIPALE: Fenton, Bieman, "Software Metrics: a rigorous and practical approach", third edition, 2015, ISBN: 978-1-4398-3823-5
- ⇒ Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 10th Edition, Harold Kerzner, ISBN: 978-0-470-27870-3, 2009
- ⇒ **Lecture Specifiche**: Durante le lezioni saranno forniti articoli e capitoli di libri, distribuiti lucidi e dispense utili per lo studio degli argomenti trattati

Tutto il materiale sarà messo a disposizione sulla **piattaforma ADA** del corso.

- ⇒ **Codice Iscrizione (Piattaforma e-Learning):** **MMQS-2526**

Esami ...

- ⇒ l'esame consisterà di una unica **PROVA FINALE** che verifica il livello di acquisizione dei concetti illustrati durante il corso.
- ⇒ La **PROVA FINALE** consiste in domande (orali o scritte) che vertono su **TUTTO** il programma e i contenuti del corso (teoria, esercitazioni, esercizi, esempi di applicazione, uso di strumenti, etc.)
- ⇒ Lo studente dovrà mostrare di aver acquisito i concetti e di saperli applicare utilizzando i metodi, le tecniche, gli strumenti presentati a lezione

... Esami

- ⇒ La PROVA FINALE si svolge nella data corrispondente all'ORALE di ogni appello.
- ⇒ È **OBBLIGATORIO** prenotarsi alla prova finale attraverso il **sistema ESSE3**
- ⇒ **NON SONO AMMESSE ALTRE FORME DI PRENOTAZIONE** (email a docenti, prenotazioni per telefono etc). Non ci saranno eccezioni di alcun tipo.
- ⇒ Gli studenti dovranno **risolvere per tempo eventuali problemi** (mancanza di account, dimenticanza login, password etc etc etc) rivolgendosi a chi di competenza (webmaster del sistema ...).

Orario delle Lezioni



- ❑ Martedì: 13.30 – 15.10; Aula Magna – Piano Terra
- ❑ Giovedì: 15.10 – 18.30; Aula Magna – Piano Terra

Organizzazione del Corso

- ⇒ **Lezione in Classe**: Teoria ed esempi / esercitazioni (dispense e approfondimenti)
 - ⇒ Durante il corso saranno assegnati dei compiti da svolgere in classe/a casa e discutere/presentare in aula.
 - ⇒ Alcune dispense e materiale di supporto potrebbero essere in inglese
- ⇒ **Seminari / Guest Lectures** su argomenti inerenti il corso (i.e. Software Quality) da parte di docenti universitari (italiani e stranieri) e aziende del territorio (seminari on-line e, laddove possibile, in presenza).
 - ** I seminari sono argomento d'esame e di approfondimento

Voto Finale

⇒ Il voto finale è determinato da:

- ❑ Voto scritto della prova finale
- ❑ + eventuali bonus acquisiti per attività svolte durante il corso (**Validi per voti verbalizzati entro l'appello di luglio)


Appelli 2025-2026

- ⇒ 3 appelli tra Gennaio – Febbraio 2026
- ⇒ 1 appello tra Giugno – Luglio 2026
- ⇒ 2 appelli recupero a Settembre 2026
- ⇒ 1 appello a Novembre 2026
- ⇒ 1 appello ad Aprile 2027

Materiale Didattico

- ⇒ I lucidi delle lezioni si possono scaricare a partire dal sito di E-LEARNING del DIB sulla piattaforma ADA:
 - ❑ È necessario **iscriversi al corso 2025-2026**
- ⇒ **Chiave di iscrizione: MMQS-2526**
- ⇒ La frequenza non è obbligatoria, ma fortemente consigliata

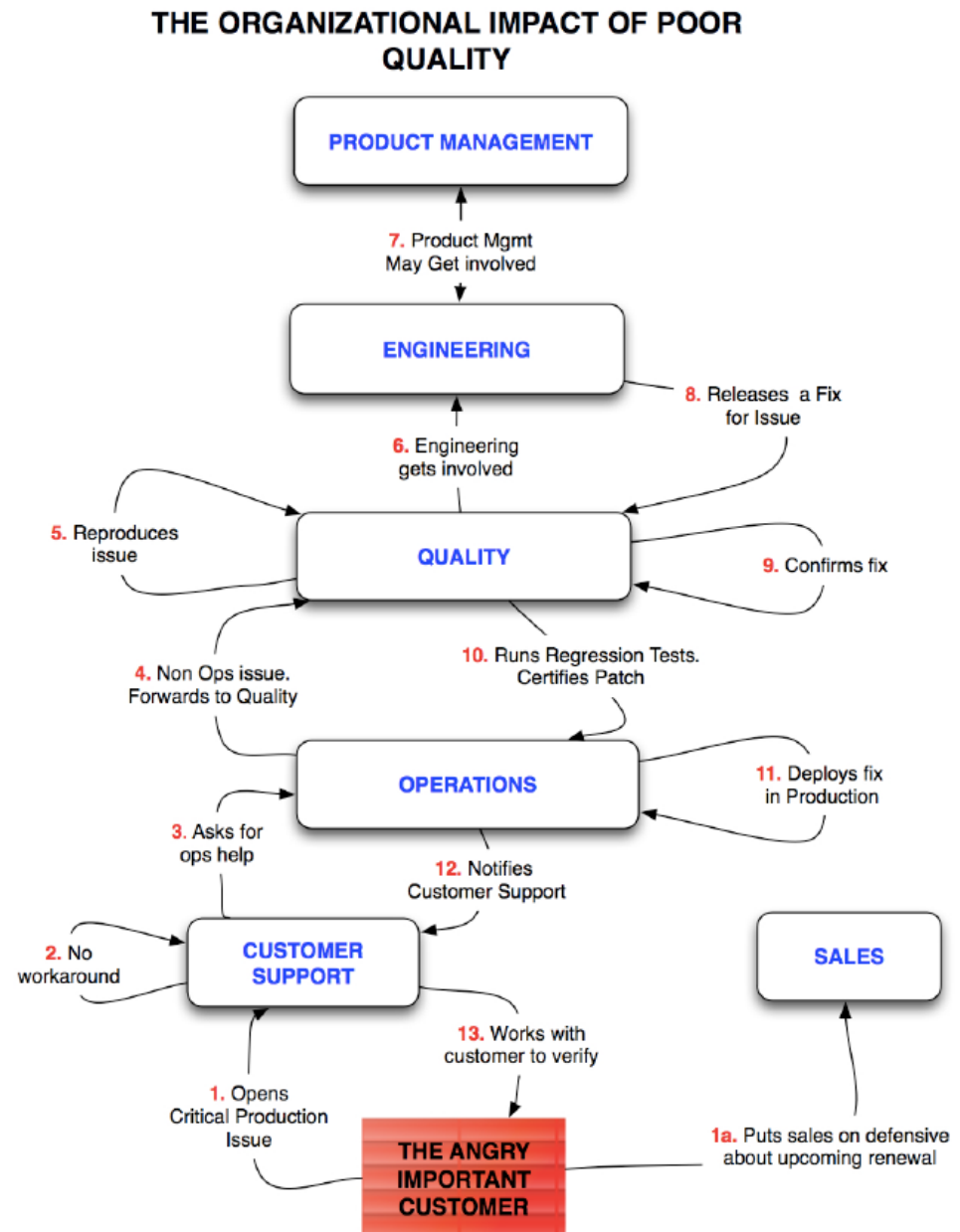




**SOFTWARE AND CATHEDRALS ARE MUCH THE
SAME - FIRST WE BUILD THEM, THEN WE PRAY.**

SAM REDWINE

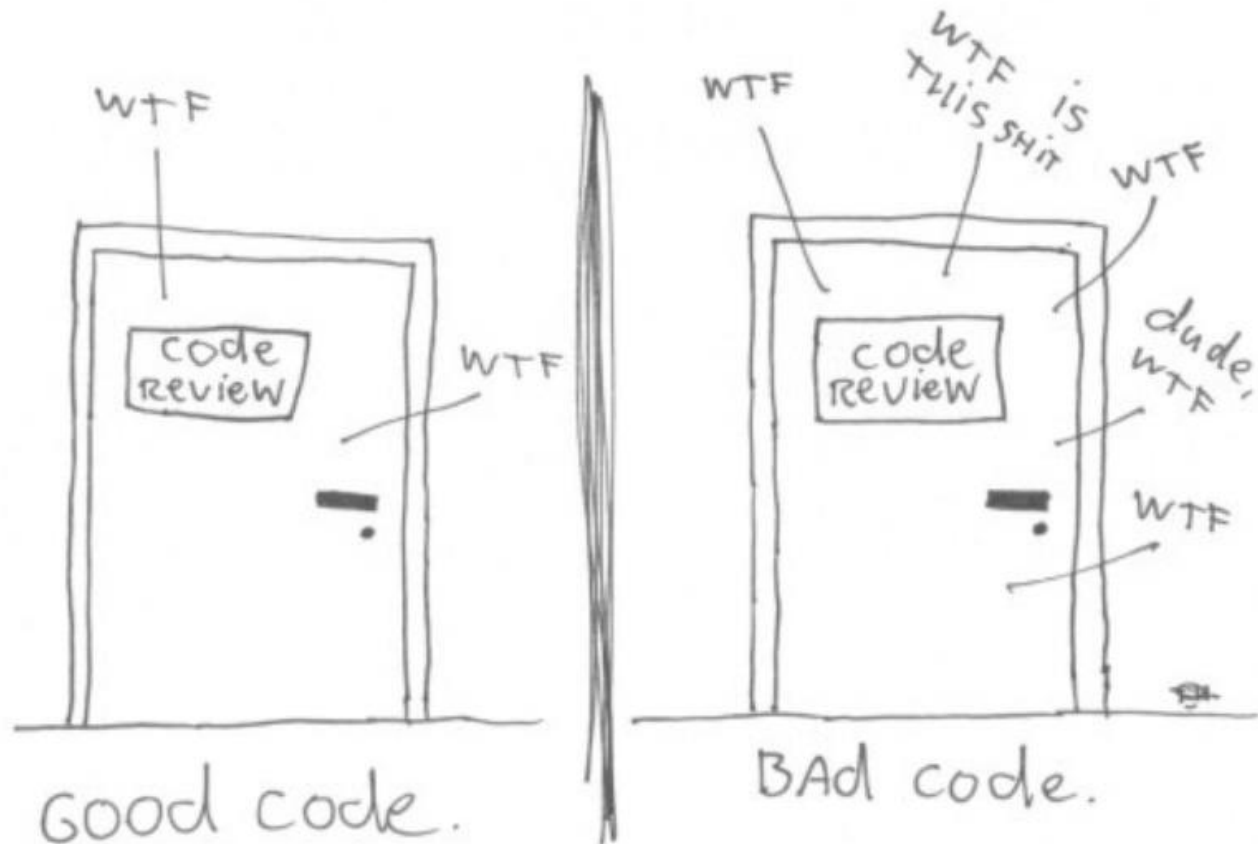
Gestione della qualità nelle grandi aziende



⇒ Come possiamo dire che un prodotto software è di buona qualità?



The ONLY valid measurement
of code quality: WTFs/minute



Cosa si intende per Qualità?

- ⇒ La qualità è un **termine ambiguo**, con molte interpretazioni.
- ⇒ Ciò è dovuto alle seguenti ragioni
 - 1. è un concetto che non è semplice, ma **multidimensionale**. Le dimensioni sono:
 - L'entità di interesse.
 - Il **punto di vista** che abbiamo di tale entità.
 - **Attributi di qualità** in grado di misurare più di tale entità

Cosa si intende per Qualità?

- ❑ 2. Per questo concetto, ci sono diversi **livelli di astrazione**:
 - A volte si usa il concetto di qualità nel senso più ampio del termine.
 - In altri, viene usato per indicare un aspetto molto specifico di un prodotto
- ❑ 3. Si tratta di un concetto che spesso usiamo nella nostra vita quotidiana.
 - Il significato di questo termine è diverso se usato in un ambito popolare (quotidiano) o un professionale.

Cosa è la Qualità: **prospettiva popolare**

- ⇒ Popolarmente, si intende che la qualità sia una **caratteristica immateriale / non tangibile**. Si possono esprimere opinioni, commenti, giudizi, ma non è possibile misurarla concretamente.
- ⇒ Diciamo che qualcosa è di "**buona qualità**", "**cattiva qualità**" o si parla di "qualità della vita"
- ⇒ In questo senso, la qualità non può essere quantificata o gestita.
- ⇒ A volte la qualità esprime anche un senso di "lusso", "classe" o "buon gusto" ed è riservata ai prodotti generalmente **costosi**, con **funzionalità sofisticate**. Prodotti semplici, che non sono costosi, raramente vengono definiti come prodotti di qualità

Cos'è la qualità: **prospettiva Professionale**

- ⇒ Autori rilevanti nel campo della qualità professionale definiscono qualità rispetto a:
- ❑ 1. **conformità con i requisiti**. Ciò implica che i requisiti del prodotto devono essere **chiaramente definiti**. Durante lo sviluppo, è necessario verificare che il prodotto soddisfa tali requisiti. Non conformità con i requisiti, è considerato un difetto.
 - ❑ 2. **adeguatezza circa l'utilizzo**. Ciò implica che le esigenze degli utenti e le aspettative sono note (riportate formalmente nei documenti di progetti) e che il prodotto li soddisfa (verificati attraverso test, collaudi etc).



⇒ Qual è il prodotto di maggiore qualità?



Cos'è la Qualità del Software

- ⇒ Il significato più comune è che **un software è di qualità quando non ha difetti (bug) di tipo generalmente funzionali**
- ⇒ Per esprimere questa definizione di qualità, generalmente si adottano misure/metriche [quantitative - qualitative]:
 - ❑ **Tasso di difetti**: No. difetti / PF, n difetti / KLOC
 - ❑ **Affidabilità**: No. Fallimenti (operazioni nn andate a buon fine) / Nr. ore di funzionamento, tempo medio tra i fallimenti.
 - ❑ **Soddisfazione degli utenti**: % di utenti soddisfatti (misurato attraverso indagini)

Definizione di Qualità del software

- ⇒ (ISO) Insieme di proprietà e di caratteristiche desiderate per un processo, un prodotto, un servizio software allo scopo di :
 - ❑ Soddisfare il committente, attraverso i requisiti impliciti od espliciti
 - ❑ Realizzare il ritorno economico degli investitori
 - ❑ Realizzare trade - off tra qualità tecnica, costo e tempo di esecuzione del processo.
- ⇒ La misurazione cattura informazioni sugli attributi delle entità. Un'entità è un oggetto o un evento del mondo reale (processo, prodotto, risorsa).
 - ❑ L'entità è descritta attraverso caratteristiche specifiche
 - ❑ Un attributo è un aspetto o una proprietà specifica di un'entità

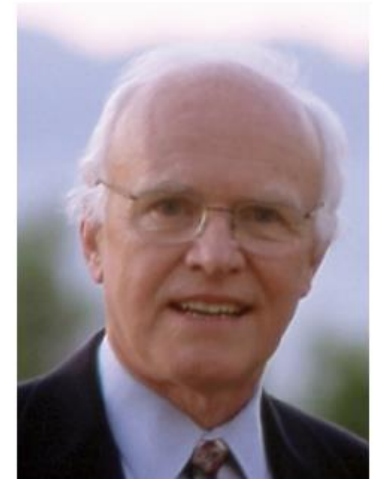
⇒ Non solo ...



Qualità del software

- ⇒ *Qualità funzionali*: grado di soddisfazione dei **requisiti funzionali**, valutato mediante **testing**
- ⇒ *Qualità strutturali*: grado di soddisfazione dei **requisiti non funzionali** da parte **dell'architettura**, valutato mediante analisi architetturali

Principio di Tom DeMarco



Non puoi controllare quel che
non puoi misurare!

Controlling Software Projects, Management Measurement & Estimation, 1982

Quanto è importante misurare?

Progetti	Con misurazioni	Senza misurazioni
puntuali	75%	45%
In ritardo	20%	40%
cancellati	5%	15%
Rimozione difetti	95%	<85%
Stima risorse necessarie	Accurata	Ottimistica
Soddisfazione del cliente	Alta	Bassa
Morale del team	Alto	Basso

Fonte: Capers Jones, Measurement, Metrics and Industry Leadership, 2009 e
Software Engineering Best Practices, McGraw Hill, 2010

Ragioni per cause legali sul software

Requisiti instabili, modificati continuamente	95%
Controllo di qualità inadeguato, senza misurazioni	90%
Cattivo monitoraggio del processo di sviluppo	85%
Stime sbagliate dei costi o dei tempi	80%
False promesse dei venditori	80%
Stime ottimistiche o arbitrarie	75%
Sviluppo informale, non strutturato	70%
Clienti inesperti incapaci di definire i propri requisiti	60%
Project manager inesperti	50%
Mancato uso di tecniche di analisi statica o ispezioni	45%
Riuso di software con errori	30%
Team di progettisti inesperto o non qualificato	20%

Fonte: Capers Jones

- Sviluppatori individuali: **A chi interessano le misure del sw**
 - Distribuzione dello sforzo
 - Durata e sforzo a preventivo (stimati) e a consuntivo
 - Codice coperto da testing di unità
 - Numero di difetti trovati dal test di unità
 - Complessità del progetto e del codice
- Team di progetto
 - Dimensione del prodotto
 - Distribuzione dello sforzo
 - Stato dei requisiti (#approvati, #implementati, #verificati)
 - % dei casi di test superati
 - Durata stimata e reale tra due milestone principali
 - Livelli di staff stimati e reali
 - #difetti trovati dai test di integrazione e di sistema
 - #difetti trovati dalle ispezioni
 - Stato dei difetti
 - Stabilità dei requisiti (#requisiti modificati durante lo sviluppo)
 - Numero di task pianificati e completati
- Organizzazione che sviluppa software
 - Livelli dei difetti rilasciati (*critical, major, average, minor, exception*)
 - Tempo di ciclo di sviluppo del prodotto
 - Accuratezza della pianificazione e dello sforzo stimati
 - Riuso effettivo
 - Costo preventivato e reale

Misurare la qualità

⇒ In un qualsiasi ciclo di vita del software le entità visibili e misurabili sono i processi (le attività), le risorse impiegate e alcuni attributi dei prodotti

- **Qualità di un prodotto software:**

- la qualità di un prodotto software è la misura in cui il prodotto soddisfa la sua specifica

- **Attributi di qualità interni ed eterni**

- **Attributi esterni:** visibili all'utente
- **Attributi interni:** visibili agli sviluppatori

Esempi di misure usate in Sw Eng

Entità

Attributo

Metrica

codice	lunghezza	LOC
	funzionalità	Function Points
	complessità	Indice McCabe
specifiche	lunghezza	#pagine
	riuso	#pagine
codifica	sforzo	Mesi/persona
testing	Fase rilevazione	%difetti trovati
	volume	#test schedulati
manutenzione	costo	€

Tipi di problemi nel software

- ⇒ **Failure**: comportamento del software non previsto dalla sua specifica
- ⇒ **Fault**: difetto del sorgente (detto anche bug), causa di un failure
- ⇒ **Error**: causa di un difetto; esempio: un errore umano d'interpretazione della specifica o nell'uso di un metodo

Da IEEE Standard Glossary of SE Terminology

Perché il software è difettoso?

- ⇒ Gli **esseri umani commettono errori**, specie quando eseguono compiti complessi; ciò è inevitabile
- ⇒ I programmatori esperti commettono in media **un errore ogni 10 righe**
- ⇒ Circa il **50%** degli errori di codifica vengono catturati a tempo di compilazione
- ⇒ Altri errori vengono catturati col testing
- ⇒ Circa il **15%** degli errori sono ancora nel sistema quando viene consegnato al cliente

(W. Humphrey: „What if your life depended on software?” EuroSPI conference, Copenhagen, April 2000)

Da dove arrivano i difetti?

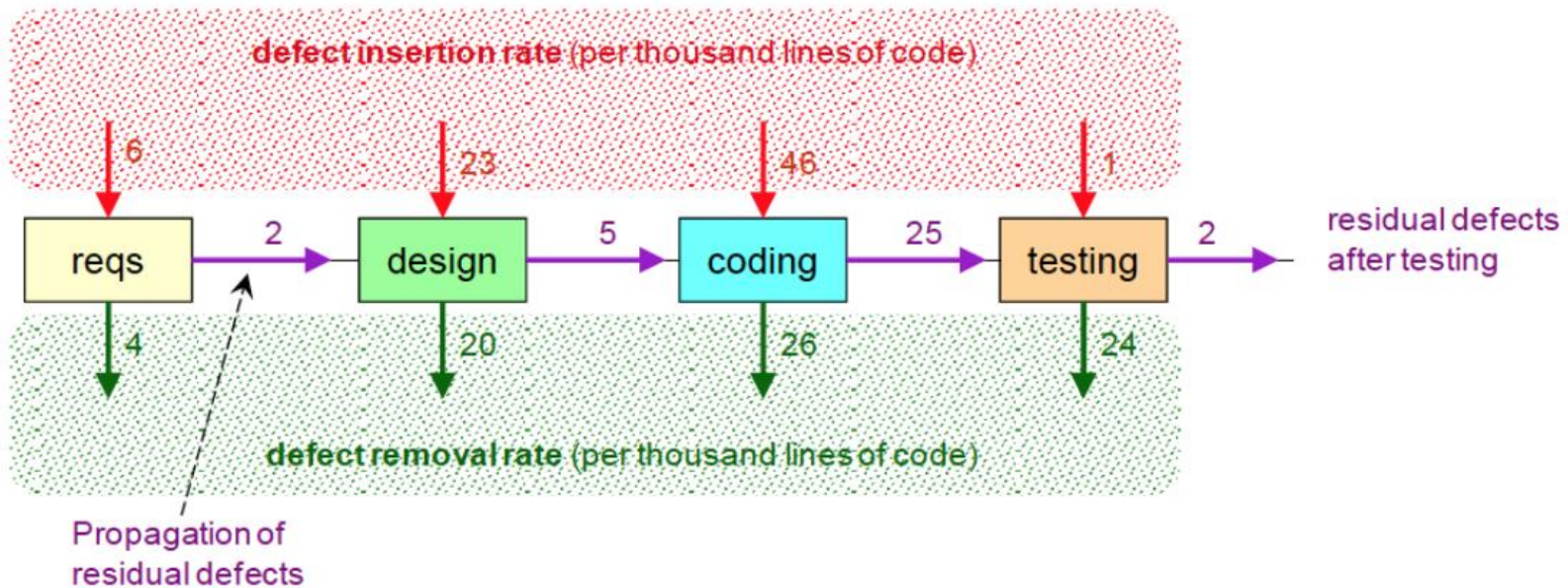


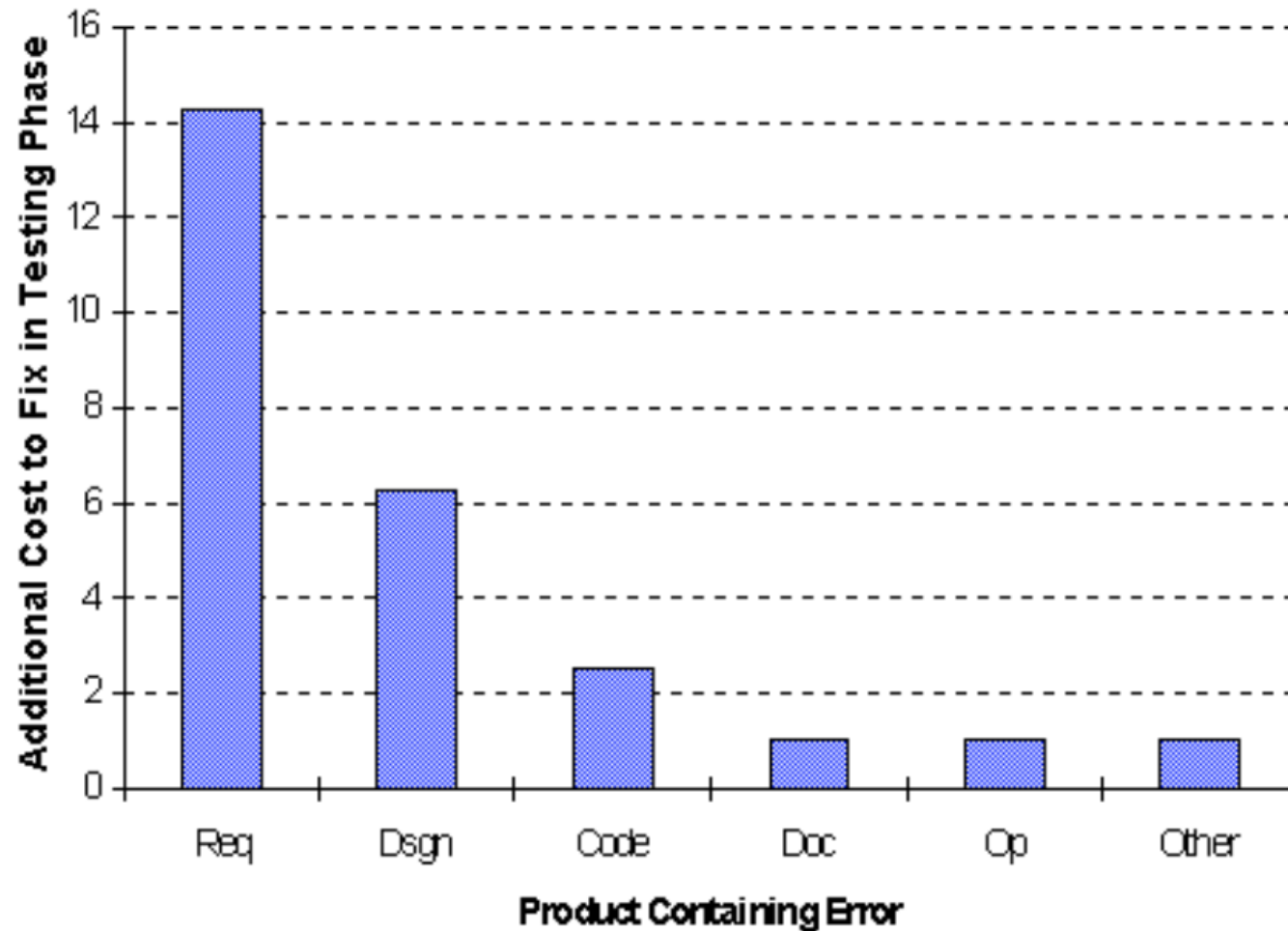
Figure 11. Defects are inserted and removed at different rates in each part of the development lifecycle.

The difference between insertion and removal rates determines defect propagation rate.

This example shows that 2 defects per thousand lines of code remain after testing.

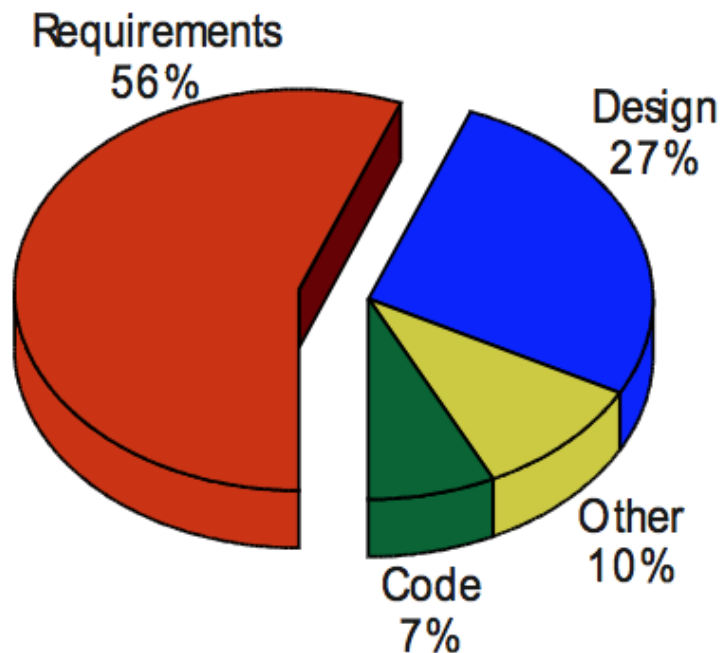
(S.G. Eick, C.R. Loader et al., Estimating software fault content before coding, Proc. 15th Int. Conf. on Software Eng., Melbourne, Australia, 1992, pp. 59-65)

Costo relativo di correzione dei difetti

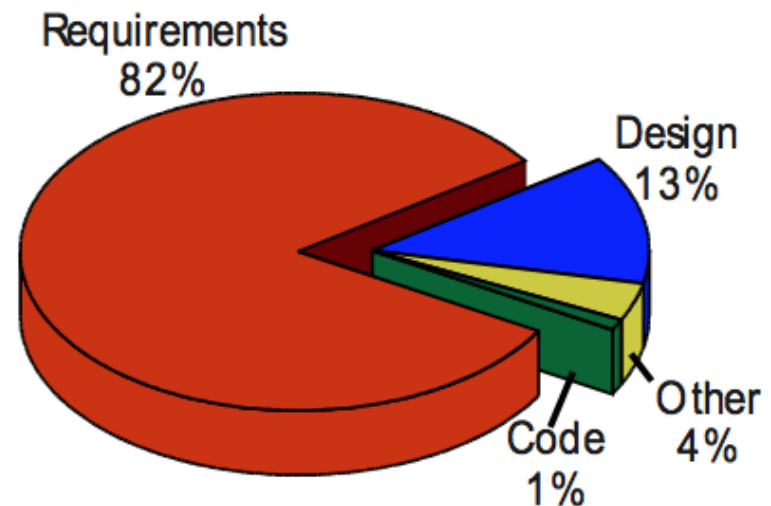


Lo sforzo per correggere i difetti

Distribution of Bugs

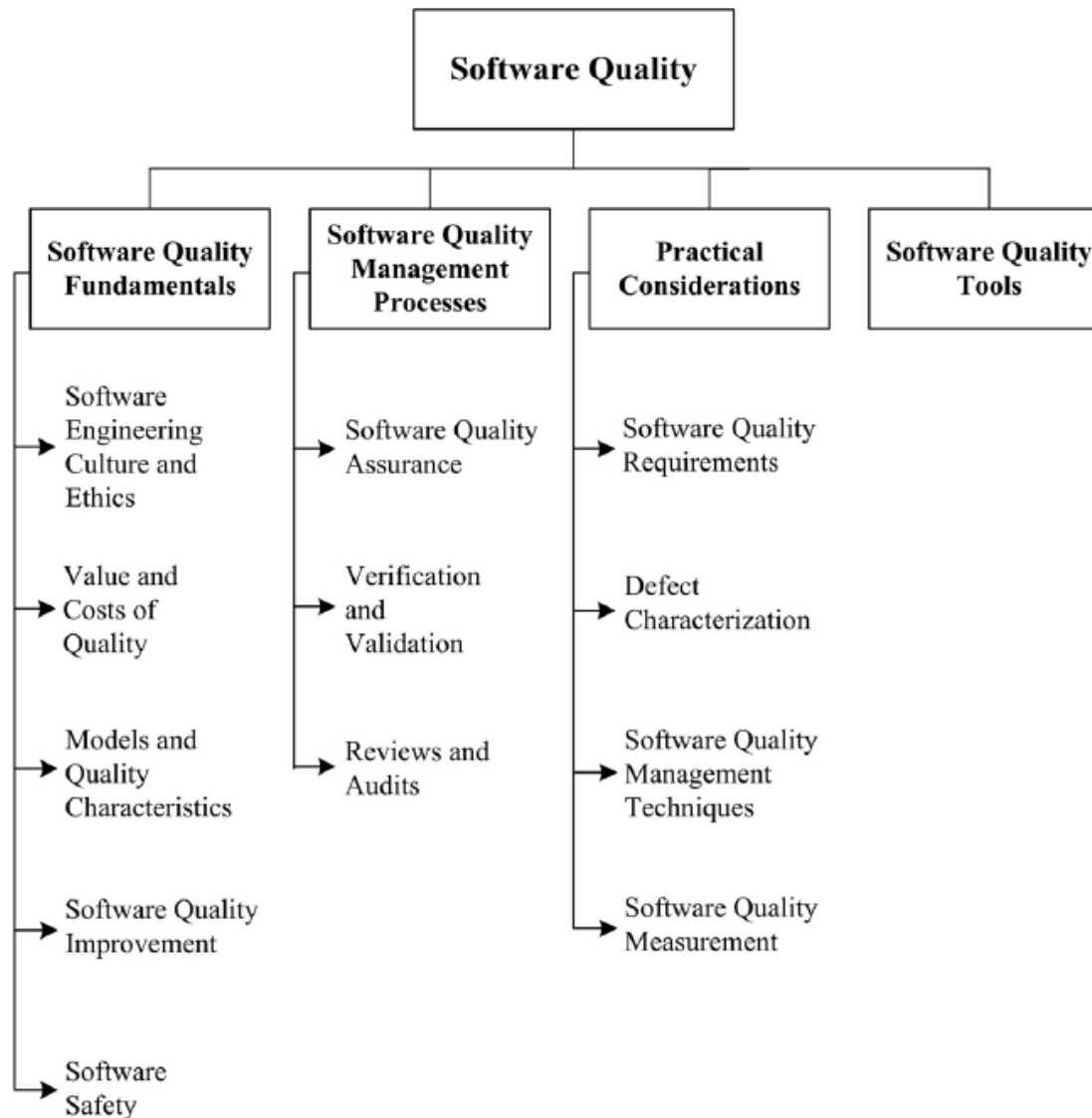


Distribution of Effort To Fix Bugs



(James Martin)

La qualità del SW nel SWEBOK



Il problema della qualità del software

- ⇒ Assicurare la qualità di un prodotto o servizio è difficile. Nel caso del software valutarla o garantirla è particolarmente complesso
- ⇒ Esistono
 - ❑ **Attività** legate alla qualità del software (es. testing)
 - ❑ **Metodi** orientati alla **qualità di prodotto** (es. cleanroom)
 - ❑ **Metodi** orientati alla **qualità di processo** (es. ISO9000)

Come introdurre la qualità ...

- ⇒ Qualsiasi valutazione di qualità inizia dallo **scopo** (**goal**) che ha chi la vuole valutare
- ⇒ Chi valuta la qualità di una entità (processo, prodotto, servizio, risorsa ...) dovrebbe:
 - ❑ **Avere chiari i propri obiettivi**
 - ❑ **Legarli a domande specifiche sull'entità** (prodotto o processo) oggetto di analisi
 - ❑ **Definire metriche capaci di analizzare e quantificare** le qualità richieste ai prodotti rispetto agli obiettivi



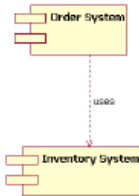


... come introdurre la qualità

- ⇒ Le **metriche** che vengono definite debbono **correlarsi** agli **obiettivi** (goals) dell'organizzazione
- ❑ Le **misurazioni** dovrebbero essere **utili e costruttive**, perché l'organizzazione deve **imparare analizzandoli**
 - ❑ Le metriche e le loro definizioni dovrebbero **riflettere il punto di vista di diverse parti interessate** (es. sviluppatori, utenti, progettisti etc).

Pianificazione di un Modello di Qualità: Passi da Seguire

1. Sviluppo degli obiettivi da raggiungere e delle misure associate alla qualità
2. Generazione di domande che definiscono gli obiettivi quantificandoli
3. Specificare le misure da collezionare in conformità a ciascun obiettivo
4. Sviluppare i meccanismi operativi di collezione delle misure
5. Raccogliere i dati e analizzarli onde sviluppare azioni correttive
6. Analizzare i dati post-mortem per raccomandazioni sul futuro.

Attività di sviluppo del software: la qualità in ogni fase

Raccolta dei requisiti	Analisi dei requisiti	System Design	Object design	Implementazione	Testing
					
	espressi mediante	mappati su	realizzati con	definiti da	verificati con
				<div>Class A...</div> <div>Class B...</div> <div>Class C...</div>	
Modello dei casi d'uso	Oggetti del dominio	Sotto-sistemi	Oggetti del sistema	Sorgente	Casi di test

Obiettivi di Verifica e Validazione

- ⇒ **Verifica**: confronto di un prodotto con la sua specifica (ovvero, confronto di un prodotto con i suoi requisiti)
- ⇒ **Validazione**: accettazione del prodotto da parte del committente

Verifica

- ⇒ L'attività di verifica, così come quella di documentazione, non è una fase separata del processo
- ⇒ Tuttavia è opportuno che sia effettuata da **persone diverse** da quelle coinvolte nel design o nella codifica
- ⇒ Ogni **documento prodotto** dovrebbe essere **controllato** (possibilmente da persone diverse dagli autori del documento) e **sistematicamente documentato** esso stesso
- ⇒ Esistono due tipi di verifica:
 - ❑ **Ispezione/Code Review**: basata sull'analisi statica
 - ❑ **Testing**: basata sull'analisi dinamica, esecuzione

Attività legate alla qualità

- ⇒ **Testing**: processo di investigazione sui rischi connessi all'esecuzione di un sistema software
- ⇒ **Misurazione**: indicatori di qualità sia mediante ispezione che mediante esecuzione
- ⇒ **Verifica**: analisi delle funzioni rispetto alla specifica
- ⇒ **Validazione**: accettazione da parte degli stakeholder
- ⇒ **Certificazione**: analisi delle funzioni rispetto ai requisiti di legge da certificare

Costo della qualità (ISO)

⇒ Due costi principali:

- ❑ Costo per fare bene le cose (**conformità**)
- ❑ Costo per rimettere a posto le cose sbagliate (**non conformità**)

⇒ I costi della qualità nel processo produttivo sono i costi che si sostengono per adeguare la qualità del prodotto alla qualità richiesta

Costo della conformità (COC) (per soddisfare tutte le esigenze espresse ed implicite)

costi di prevenzione: oneri sostenuti per prevenire gli insuccessi

costi di accertamento: oneri per controlli e collaudi

Costo della non conformità (CNC) (per insuccessi interni ed esterni)




costi per insuccessi interni: oneri connessi ad un prodotto che non soddisfa i requisiti di qualità prima ancora della sua consegna

costi per insuccessi esterni: oneri connessi ad un prodotto che non soddisfa i requisiti di qualità dopo la sua consegna (costi di manutenzione e riparazione, costi di garanzia e resi, costi per il richiamo dei prodotti, costi per la responsabilità da prodotto,...)

Standardizzazione

- ⇒ Diversi **enti di standardizzazione** (eg ISO) hanno cercato di integrare vari approcci alla definizione della qualità, partendo dalla consapevolezza che la qualità è un attributo che varia in funzione del:
 - ❑ Percettore
 - ❑ Del contesto di percezione
 - ❑ Dello scopo e costo del prodotto
- ⇒ ISO/IEC 9126 (prodotti software)
- ⇒ ISO 9000 – ISO 12207 – CMM (processi)

ISO e altre Organizzazioni

		<u>Settori vari</u>	<u>Settore elettrotecnico</u>
<u>A livello internazionale</u>		<i>ISO</i>	<i>IEC</i>
<u>A livello europeo</u>		<i>CEN</i>	<i>CENELEC</i>
<u>A livello nazionale</u>		<i>UNI</i>	<i>CEI</i>

Sigle

ISO	International O rganization for S tandardization
IEC	International E lectrotechnical C ommission
CEN	Comitato E uropeo di N ormazione (sigla sui documenti EN)
CENELEC	Comitato E uropeo di N ormazione E lettrotecnica (sigla sui documenti EN HD)
UNI	E nte N azionale Italiano di U nificazione
CEI	Comitato E lettrotecnico Italiano

Tra gli organismi ISO a livello nazionale non europeo:

ANSI	A merican N ational S tandard Institute, per gli Stati Uniti;
JISC	J apan I ndustrial S tandards C ommittee, per il Giappone;
SA	S tandards A ustralia, per l'Australia;
SCC	S tandard C ouncil of C anada, per il Canada.

I 18 organismi "National Standard Bodies" del CEN rappresentano tutti i paesi europei, con l'aggiunta di 7 affiliati di cui 5 paesi limitrofi, più Cipro e Turchia (tra questi rientra l'**UNINFO** - Ente di normazione Federato all'UNI per le Tecniche Informatiche e le loro applicazioni - per l'Italia).

