



Università degli Studi di Udine

Qualità

prof. Maurizio Pighin

Dipartimento di Matematica e Informatica



Software quality management

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Managing the quality of the software process and products
- Concerned with ensuring that the required level of quality is achieved in a software product
- Involves defining appropriate quality standards and procedures and ensuring that these are followed
- Should aim to develop a 'quality culture' where quality is seen as everyone's responsibility

Slide 2





What is quality?

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Quality, simplistically, means that a product should meet its specification
- This is problematical for software systems
 - *Tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.)*
 - *Some quality requirements are difficult to specify in an unambiguous way*
 - *Software specifications are usually incomplete and often inconsistent*

Slide 3



The quality compromise

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- We cannot wait for specifications to improve before paying attention to quality management
- Must put procedures into place to improve quality in spite of imperfect specification
- Quality management is therefore not just concerned with reducing defects but also with other product qualities

Slide 4





Quality management activities

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

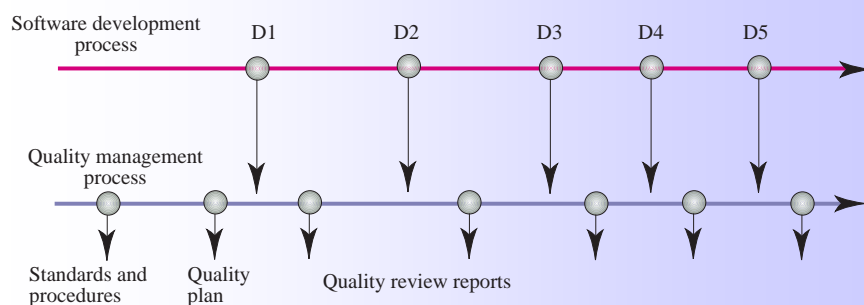
- Quality assurance
 - *Establish organisational procedures and standards for quality*
- Quality planning
 - *Select applicable procedures and standards for a particular project and modify these as required*
- Quality control
 - *Ensure that procedures and standards are followed by the software development team*
- Quality management should be separate from project management to ensure independence

Slide 5



Quality management and software development

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin



Slide 6





Quality assurance and standards

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Standards are the key to effective quality management
- They may be international, national, organizational or project standards
- **Process standards** define how the software process should be enacted
- **Product standards** define characteristics that all components should exhibit e.g. a common programming style

Slide 7



Importance of standards

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Encapsulation of best practice
 - *avoids repetition of past mistakes*
- Framework for quality assurance process
 - *it involves checking standard compliance*
- Provide continuity
 - *new staff can understand the organisation by understand the standards applied*

Slide 8





Problems with standards

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Not seen as relevant and up-to-date by software engineers
- Involve too much bureaucratic form filling
- Unsupported by software tools so tedious manual work is involved to maintain standards

Slide 9



Standards development

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Involve practitioners in development.
 - *Engineers should understand the rationale underlying a standard*
- Review standards and their usage regularly.
 - *Standards can quickly become outdated and this reduces their credibility amongst practitioners*
- Detailed standards should have associated tool support.
 - *Excessive clerical work is the most significant complaint against standards*

Slide 10





Process and product quality

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- The quality of a developed product is influenced by the quality of the production process
- Particularly important in software development as some product quality attributes are hard to assess
- However, there is a very complex and poorly understood relationship between software processes and product quality

Slide 11



Process improvement

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Understanding existing processes
- Introducing process changes to achieve organisational objectives which are usually focused on quality improvement, cost reduction and schedule acceleration
- Most process improvement work so far has focused on defect reduction. Reflects increasing attention paid by industry to quality

Slide 12





Process improvement stages

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

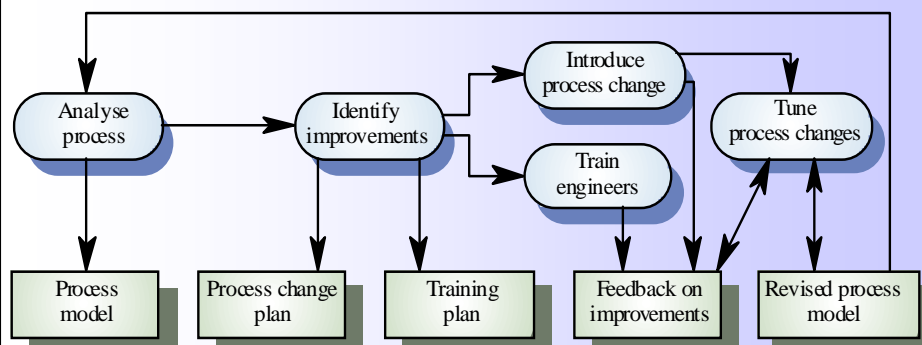
- Process analysis
 - Model and analyse (quantitatively if possible) existing processes
- Improvement identification
 - Identify quality, cost or schedule bottlenecks
- Process change introduction
 - Modify the process to remove identified bottlenecks
- Process change training
 - Train staff involved in new process proposals
- Change tuning
 - Evolve and improve process improvements

Slide 13



The process improvement process

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin



Slide 14





The Software Engineering Institute (SEI)

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- US Defence Dept. funded institute associated with Carnegie Mellon
- Mission is to promote software technology transfer particularly to defence contractors
- Maturity model (CMM) proposed in mid-1980s, refined in early 1990s.
- Work has been very influential in process improvement

Slide 15



The SEI process maturity model

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Vengono definiti 5 livelli di Capacità dell'organizzazione
 - *Eseguita, Gestita, Definita, Gestita quantitativamente, Ottimizzata*
- Per ogni processo viene definito un insieme di aree da valutare
 - *Fasi di processo (requisiti, progettazione, implementazione, test di modulo, di integrazione, rilascio...)*
 - *Documenti di progetto (casi d'uso, casi di test, sorgenti, ...)*
 - ...
- Ciascuna area del processo viene valutata rispetto ad una serie di
 - *Obiettivi Specifici (Specific Goals – SG) definiti tramite pratiche specifiche (Specific Practices – SP)*
 - *Obiettivi Generici (Genreic Goals – GG) definiti tramite pratiche generiche (Specific Practices – GP)*

Slide 16





Capability Maturity Model (CMM)

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Livelli
 - **Livello 0: incompleta**
 - L'area del processo non è trattata o è trattata parzialmente
 - **Livello 1: iniziale**
 - L'area del processo è trattata, ma con processi ad hoc o con processi poco definiti
 - **Livello 2: gestita**
 - Soddisfa il Livello 1
 - L'area del processo è trattata con processi conformi alla politica definita a livello dell'organizzazione
 - Le persone che svolgono il lavoro hanno accesso a risorse adeguate
 - Gli stakeholder interessati sono coinvolti attivamente nell'area del processo
 - Le attività e i prodotti sono monitorati, controllati e revisionati; ne viene valutata l'adesione alla descrizione dei processi

Slide 17



Capability Maturity Model

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Livello 3: definita**
 - Soddisfa il Livello 2.
 - I processi dell'area sono definiti, documentati, standardizzati ed integrati con gli altri processi dell'organizzazione
 - Vengono raccolte informazioni per il miglioramento dei processi
 - Sono definite ed applicate procedure per il controllo della qualità dei processi
- **Livello 4: gestita quantitativamente**
 - Soddisfa il Livello 3.
 - La valutazione della qualità e delle prestazioni dei processi viene condotta con riferimento ad obiettivi quantitativi.
- **Livello 5: ottimizzata**
 - Soddisfa il Livello 4.
 - Vengono condotte azioni di ottimizzazione dei processi dell'area basati sull'analisi statistica dei dati storici.

Slide 18





Capability Maturity Model

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Livello	Attività	Aree del processo
Ottimizzata	Miglioramento continuo del processo	Innovazione e miglioramento aziendale Analisi delle relazioni causa-effetto
Gestita quantitativamente	Gestione quantitativa	Prestazione dei processi aziendali Gestione quantitativa del progetto
Definita	Standardizzazione del processo	Sviluppo dei requisiti Soluzione tecnica Integrazione del prodotto Verifica Convalida Coinvolgimento nel processo aziendale Definizione del processo aziendale Addestramento aziendale Gestione integrata del progetto Gestione integrata dei fornitori Valutazione dei rischi Analisi e risoluzione delle decisioni Ambiente aziendale per l'integrazione Definizione integrata del team di sviluppo
Gestita	Gestione base del progetto	Gestione dei requisiti Pianificazione del progetto Monitoraggio e controllo del progetto Gestione degli accordi con i fornitori Controllo della qualità del prodotto software Gestione della configurazione
Eseguita		

Slide 19



Capability Maturity Model

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Esempio “pianificazione progetto”
 - SG 1 *Definire le stime.*
 - SP 1.1-1 Stimare l'ampiezza del progetto.
 - SP 1.2-1 Stabilire le stime del prodotto del lavoro e degli attributi delle attività.
 - SP 1.3-1 Definire il ciclo di vita del progetto.
 - SP 1.4-1 Stimare l'impegno e i costi.
 - SG 2 *Sviluppare un piano per il progetto.*
 - SP 2.1-1 Definire il budget e la pianificazione dei tempi.
 - SP 2.2-1 Identificare i rischi del progetto.
 - SP 2.3-1 Pianificare la gestione dei dati.
 - SP 2.4-1 Pianificare le risorse del progetto.
 - SP 2.5-1 Pianificare le conoscenze e le abilità richieste.
 - SP 2.6-1 Pianificare il coinvolgimento degli stakeholder
 - SP 2.7-1. Definire il piano del progetto.
 - SG 3 *Ottenere il coinvolgimento nel piano.*
 - SP 3.1-1 Rivedere i piani del progetto.
 - SP 3.2-1 Riconciliare i livelli del lavoro e delle risorse.
 - SP 3.3-1 Ottenere il coinvolgimento nel piano.

Slide 20





Capability Maturity Model

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- GG 1 *Raggiungere gli obiettivi specifici.*
 - GP 1.1 Eseguire le pratiche di base.
- GG 2 *Istituzionalizzare un processo gestito.*
 - GP 2.1 Creare una politica a livello dell'organizzazione.
 - GP 2.2 Pianificare il processo.
 - GP 2.3 Fornire le risorse.
 - GP 2.4 Assegnare le responsabilità.
 - GP 2.5 Addestrare il personale.
 - GP 2.6 Gestire le configurazioni.
 - GP 2.7 Identificare e coinvolgere gli stakeholder.
 - GP 2.8 Monitorare e controllare il processo.
 - GP 2.9 Valutare obiettivamente l'aderenza del processo.
 - GP 2.10 Rivedere lo stato a un livello più elevato.
- GG 3 *Istituzionalizzare un processo definito.*
 - GP 3.1 Definire un processo definito.
 - GP 3.2 Raccogliere le informazioni sui miglioramenti.
- GG 4 *Istituzionalizzare un processo gestito quantitativamente.*
 - GP 4.1 Definire gli obiettivi quantitativi del processo.
 - GP 4.2 Stabilizzare le prestazioni dei sottoprocessi.
- GG 5 *Istituzionalizzare un processo di ottimizzazione.*
 - GP 5.1 Garantire un miglioramento continuo del processo.
 - GP 5.2 Correggere le cause dei problemi.

Slide 21



SEI model problems

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Focuses on project management rather than product development.
 - Ignores use of technologies such as rapid prototyping.
 - Does not incorporate risk analysis as a key process area
 - Does not define its domain of applicability
- ↓
- Revised SEI standard
 - Capability Maturity Model *Integration* (CMMI)

Slide 22





Alcuni risultati ottenuti

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Categoria	Intervallo	Media
Guad. Produttività	9% - 67%	35%
Guad. Rilevamento difetti	6% - 25%	22%
Riduz. Scostam. tempi consegna	15% - 23%	19%
Riduz. Rilevam. difetti post-release	10% - 94%	39%
Rapporto Ricavi/Costi	4.0 - 8.8:1	5.0:1

Azienda	Caratteristica	Da	A
Hewlett Packard	Difetti/KLOC	0.4	0.11
Schlumberger	Scostam. tempi rilascio	50%	5%
Texas Instruments	Risoluzione difetto	8 ore	11 min.

Slide 23



Qualità di prodotto

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

• Che cos'è la Qualità del Software?

- *La qualità del software può essere definita come:*
 - conformità ai **requisiti funzionali** e prestazionali enunciati esplicitamente,
 - agli **standard di sviluppo** esplicitamente documentati,
 - a **caratteristiche implicite** che è lecito aspettarsi da un prodotto.
- *Tre punti fondamentali:*
 - I requisiti sono il fondamento su cui misurare la qualità;
 - Gli standard di qualità definiscono i criteri da seguire nello sviluppo software;
 - Esistono requisiti impliciti (spesso taciuti) la cui assenza compromette la qualità del software

Slide 24





Qualità di prodotto

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Che cos'è la Qualità del Software? (segue)**

- *Alcuni problemi nella definizione di qualità introdotta...*

- Le specifiche software sono in genere incomplete e spesso inconsistenti;
 - Alcuni requisiti di qualità sono difficili da specificare in maniera non ambigua;
 - Può esistere contrapposizione fra i requisiti di qualità attesi dal cliente (efficienza, affidabilità, etc.) e quelli dello sviluppatore (manutenibilità, riusabilità, etc.);
 - Alcuni requisiti di qualità sono difficili da valutare: non possiamo valutarli *direttamente*, ma soltanto *indirettamente*.

Slide 25



Qualità di prodotto

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Il problema fondamentale della qualità del software è che non possiamo valutare la qualità del software in assoluto, ma solo alcune sue manifestazioni!**

- *Ciò equivale a dire che non possiamo valutare direttamente la qualità del software, ma indirettamente, attraverso la valutazione di attributi che si correlano a questa, supponendo che la relazione tra la qualità e questi attributi sia valida!*
 - *Ad esempio: non posso misurare l'usabilità e la manutenibilità in assoluto, ma debbo riferirmi ad altri attributi misurabili correlati ad esse.*
 - *Abbiamo dunque bisogno di modelli di qualità condivisi!*

Slide 26





Qualità di prodotto

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Come caratterizzare la Qualità del Software?**

- *La qualità di un prodotto software si caratterizza attraverso un insieme finito e definito di attributi:*
- ragionevolmente esaustivi
 - in modo che per una qualsiasi richiesta di caratteristica di qualità sia possibile associarvi un sottoinsieme degli attributi definiti in modo da poterla valutare;
- privi di reciproche sovrapposizioni:
 - per evitare che più attributi riguardino la stessa caratteristica del software

Slide 27



Qualità di prodotto

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Modello di Qualità (MQ) del software**

- *È un insieme di attributi del software che fornisce uno **schema di riferimento** che, con una opportuna distribuzione di pesi per ciascun attributo, va adeguato e tarato per la rappresentazione dei requisiti di qualità **desiderati** dal committente o **posseduti** dal software.*

Slide 28





Struttura dei modelli di qualità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- I modelli di qualità del software pubblicati in letteratura sono tipicamente *gerarchici*, a n livelli.
- Il primo livello descrive un insieme di caratteristiche (**proprietà**), che, nel loro complesso, rappresentano la qualità del prodotto software, eventualmente secondo diversi punti di vista.
- Le proprietà (in genere qualitative, astratte) sono precisate attraverso degli **attributi** misurabili, quantitativi (in genere da una combinazione di attributi).
- Il grado di possesso che il software ha di questi attributi può essere valutato su una scala di riferimento, facendo ricorso ad opportune **metriche** ed a meccanismi di *rating*.

Slide 29



I Modelli di McCall e Boehm

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- I primi modelli di qualità del software sono stati sviluppati negli anni '70 da
 - *McCall*
 - [McCall, J. A., Richards, P. K., and Walters, G. F., "Factors in Software Quality", Nat'l Tech. Information Service, no. Vol. 1, 2 and 3, 1977]
 - *B. Boehm*
 - [Boehm, Barry W., Brown, J. R, and Lipow, M.: Quantitative evaluation of software quality, International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering, 1976].

Slide 30





Modello di McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Hanno un'architettura a più livelli.
- McCall prevede:
 - **Fattori**, che descrivono il software da un punto di vista esterno, quello degli utenti; i fattori corrispondono a requisiti specificati dal cliente.
 - **Criteri**, che descrivono gli elementi su cui agiscono gli sviluppatori per soddisfare i requisiti del cliente.
 - **Metriche**, che servono a controllare che i criteri sviluppati corrispondano ai fattori specificati. Vengono utilizzate dagli auditors e/o dagli addetti alle verifiche.

Slide 31



Modello di McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

3 settori e 11 fattori:



Slide 32





Modello di McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Schema dei Fattori di Qualità nel Modello di McCall**
 - **Uso del prodotto**
 - Correttezza
 - Affidabilità
 - Usabilità
 - Integrità
 - Efficienza
 - **Revisione del prodotto**
 - Manutenibilità
 - Flessibilità
 - Testabilità
 - **Transizione del prodotto**
 - Portabilità
 - Riutilizzabilità
 - Interoperabilità

Slide 33



Modello di McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Uso del Prodotto: ciò che emerge quando si tiene il software in esercizio;**
- i fattori di qualità che riguardano questo punto sono:
 - *correctness (does it do what I want?): correttezza, almeno come la vede l'utente;*
 - *reliability (does it do accurately all the time?): affidabilità;*
 - *efficiency (will it run on my hardware as well as it can?): efficienza;*
 - *integrity (is it secure?): integrità, sicurezza; non ha nulla a che fare con l'affidabilità, pensiamo all'integrità dei dati, delle business rules, etc.;*
 - *usability (can I run it?): usabilità.*

Slide 34





Modello di McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Revisione del Prodotto:** ciò che emerge quando si modifica il software;
- i fattori di qualità che riguardano questo punto sono:
 - *maintainability (can I fix it?):* manutenibilità;
 - *flexibility (can I change it?):* flessibilità;
 - *testability (can I test it?):* testabilità;
- **Transizione del Prodotto:** ciò che emerge quando si cambia piattaforma tecnologica; i fattori di qualità che riguardano questo punto sono:
 - *portability (is it possible to use it on another machine?):* portabilità;
 - *reusability (is it possible to reuse some of the software?):* riusabilità;
 - *interoperability (is it possible to interface it with another system?):* interoperabilità.

Slide 35



Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Fattori di Qualità come funzioni di Criteri**
- I fattori del Modello sono espressi in funzione di ulteriori caratteristiche di più basso livello (dette Criteri).
- Ad esempio:
 - *Correttezza = f (tracciabilità, coerenza, completezza)*
 - *Affidabilità = f (tolleranza all'errore, coerenza, accuratezza, semplicità)*
 - ...
 - *Usabilità = f (operabilità, addestramento, comunicabilità)*

Slide 36





Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **CORRETTEZZA**

La correttezza del software è definibile come il grado di adesione alle sue specifiche e agli standard definiti sia nel processo produttivo che nel suo dominio di applicazione.

- Correttezza = f (tracciabilità, coerenza, completezza)
 - **Tracciabilità**: il grado di reperibilità delle varie specifiche richieste dal software all'interno del codice, sia nell'ambiente di sviluppo che in fase di esercizio;
 - **Coerenza**: quegli attributi che si riferiscono all'uniformità delle tecniche utilizzate e delle notazioni adottate nei diversi componenti software, dai requisiti al codice (pensiamo a grandi sistemi, sviluppati da più gruppi oppure alle successive evoluzioni o operazioni di modifica);
 - **Completezza**: è la capacità del software di soddisfare tutti i requisiti per cui è stato sviluppato.

N.B. per ogni fattore vengono descritti solo i criteri non precedentemente descritti

Slide 37



Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **AFFIDABILITA'**

L'affidabilità è definibile come la capacità del software di eseguire le sue funzioni senza insuccessi in uno specificato periodo di tempo (per insuccesso si intende anche il venir meno a livelli di precisione desiderati).

- Affidabilità = f (tolleranza all'errore, coerenza, accuratezza, semplicità)
 - **Tolleranza all'errore**: il grado di affidabilità dei risultati ottenuti dal software in presenza di condizioni non normali;
 - **Accuratezza**: attributi che si riferiscono alla precisione nei calcoli e nei risultati in output;
 - **Semplicità**: se il software implementa le sue funzioni in maniera chiara e comprensibile.

Slide 38





Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **EFFICIENZA**

È il livello di utilizzo di risorse da parte del software (es. tempi di elaborazione e di comunicazione, spazio di memoria, ecc.);

- Efficienza = f (efficienza di esecuzione, efficienza di memorizzazione).
 - **Efficienza di esecuzione:** il tempo impiegato per svolgere il compito richiesto;
 - **Efficienza di memorizzazione:** la quantità di spazio occupato in memoria dai dati.

Slide 39



Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **INTEGRITA'**

è il livello di capacità del software di operare senza insuccessi dovuti ad accessi non autorizzati a codice e/o dati in uno specificato periodo di tempo.

- Integrità = f (controllo accessi, revisione accessi)
 - **Controllo Accessi:** quegli attributi del software che si riferiscono al controllo degli accessi ai dati ed al software;
 - **Revisione Accessi:** quegli attributi del software che consentono la revisione (riorganizzazione) degli accessi ai dati ed al software

Slide 40





Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **USABILITA'**
è lo sforzo necessario per usare il software (addestramento ed esecuzione) (ad esempio familiarizzazione, preparazione degli input, esecuzione, interpretazione degli output).
- **Usabilità** = $' = f$ (operabilità, addestramento, comunicabilità)
 - **Operabilità**: quegli attributi del software che si riferiscono alla semplicità delle operazioni e delle procedure necessarie al controllo dell'attivazione ed esecuzione del software;
 - **Addestramento**: quegli attributi del software che ne consentono la familiarizzazione e ne supportano la transizione dal vecchio ambiente al nuovo sistema;
 - **Comunicabilità**: quegli attributi del software che supportano una agevole assimilazione degli input e degli output utili (interazione uomo macchina).

Slide 41



Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **MANUTENIBILITA'**
È definita come lo sforzo necessario per trovare e correggere un errore all'interno del codice dopo il rilascio in esercizio al cliente;
- **Manutenibilità** = f (coerenza, semplicità, concisione, modularità, auto documentazione).
 - **Concisione**: la quantità di codice necessaria per adempiere ad una certa funzione;
 - **Modularità**: il grado di indipendenza dei vari moduli operanti all'interno del software;
 - **Auto documentazione**: la capacità del codice di spiegare le funzioni implementate.

Slide 42





Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **PORTABILITA'**

È la capacità di adattamento del software ad operare su nuovi ambienti di lavoro (nuovo hardware, software, configurazioni).

- Portabilità = f (modularità, auto documentazione, indipendenza dalla macchina, indipendenza dal software).
 - **Indipendenza dalla macchina:** gli attributi che definiscono l'indipendenza del software dalla piattaforma hardware.
 - **Indipendenza dal software:** gli attributi che definiscono il grado di indipendenza del software dall'ambiente software in cui opera, quindi indipendenza dal sistema operativo, dai drivers, utility, ecc..

Slide 43



Modello di McCall - Criteri

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **TESTABILITA'**

È la facilità con cui è possibile effettuare il testing sull'applicazione;

- Testabilità = f (semplicità, modularità, strumentazione, auto documentazione).
 - **Strumentazione:** È la facilità con cui è possibile monitorare il funzionamento del software e quindi verificarne possibili errori.

Slide 44





Modello di McCall - Metriche

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

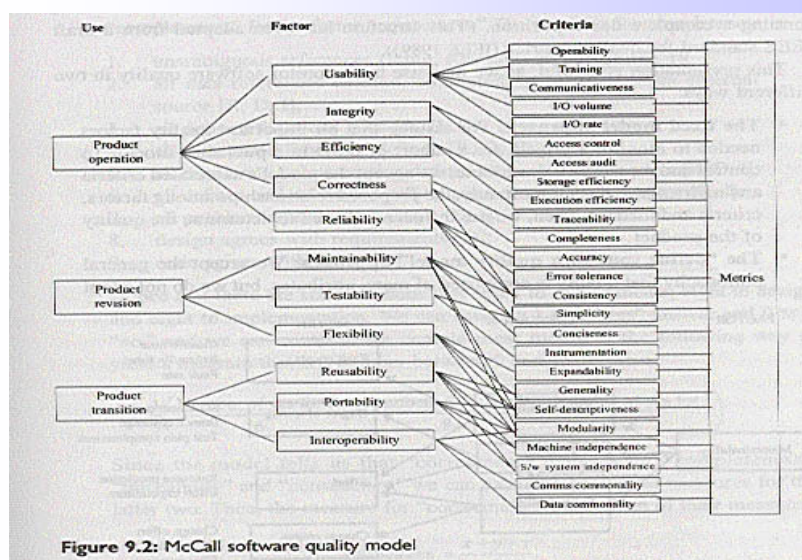
- **Le metriche nel Modello di McCall**
- McCall completò il quadro con un numero notevole di metriche:
 - 25 per le specifiche,
 - 108 per la progettazione,
 - 157 per la codifica.
- ed associando ad ogni fattore di qualità una valutazione data da una funzione lineare delle metriche interessate:
 - $F_i = a_0 + a_1 m_1 + \dots + a_k m_k$
 - gli a_i esprimono la rilevanza (peso) di ciascuna metrica ai fini della valutazione del fattore (sono ottenuti per analogia con altri progetti di caratteristiche simili);
 - ogni m_i è normalizzato sull'intervallo $[0, 1]$.

Slide 45



Modello McCall - Schema completo

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin



6





Modello McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- **Sinergie e Conflitti fra Fattori di Qualità**
 - *Esistono sinergie e conflitti fra i diversi fattori di qualità, da tenere in considerazione in sede di definizione dei requisiti di qualità e di sviluppo di un sistema software.*

Slide 47



Modello McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

	C	A	E	I	U	M	T	F	P	R	IO
Correttezza (C)	C										
Affidabilità (A)	S	A									
Efficienza (E)			E								
Integrità (I)			C	I							
Usabilità (U)	S	S	C	S	U						
Manutenibilità (M)	S	S	C		S	M					
Testabilità (T)	S	S	C		S	S	T				
Flessibilità (F)	S	C	C	C	S	S	S	F			
Portabilità (P)			C			S	S		P		
Riusabilità (R)		C	C	C		S	S	S	S	R	
Interoperabilità (IO)			C	C					S		IO

C = conflitto, S = sinergia

Slide 48





Modello McCall

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

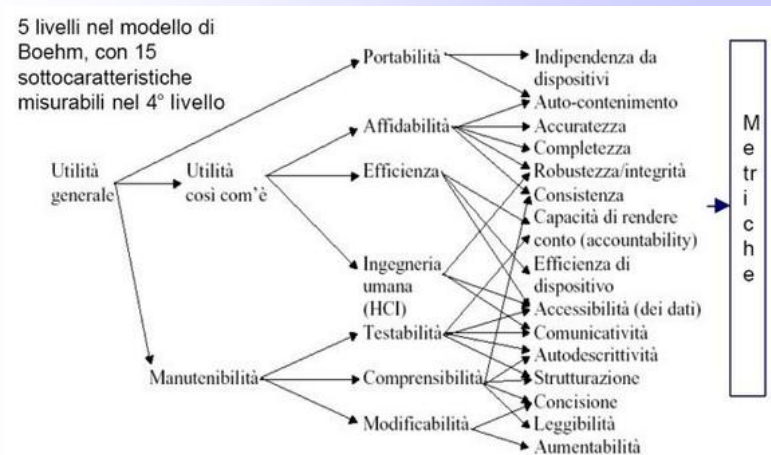


Slide 49



Modello Bohem

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin



Slide 50





Limiti dei modelli di McCall e Boehm

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Le caratteristiche sono in genere proprietà astratte misurabili solo attraverso indicatori e metriche. Non sempre l'andamento di queste grandezze è in correlazione perfettamente lineare con le caratteristiche che devono stimare;
- È difficile che le caratteristiche e sottocaratteristiche siano sempre prive di sovrapposizioni;
- Manca in ogni caso il legame esplicito tra il modello qualitativo e "come" fare poi del buon software.

Slide 51



ISO 9126

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- The objective of this standard is to provide a framework for the evaluation of software quality.
- ISO/IEC 9126 does not provide requirements for software, but it defines a quality model which is applicable to every kind of software.
- It defines six product quality characteristics and in an annex provides a suggestion of quality subcharacteristics

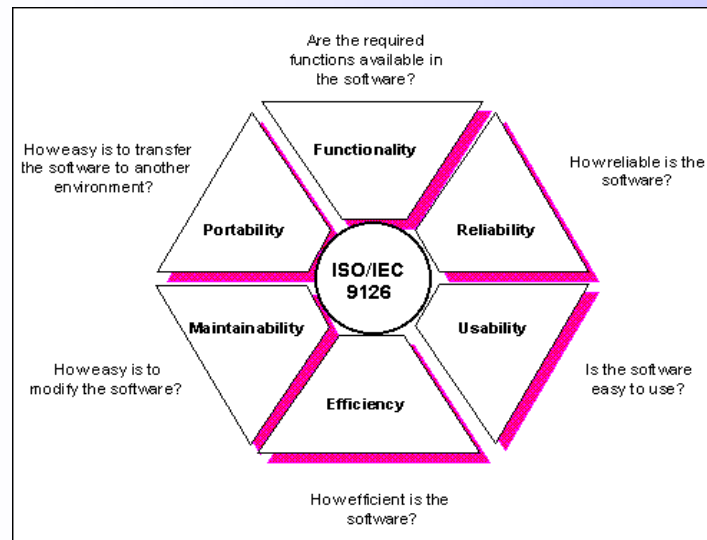
Slide 52





ISO 9126

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin



Slide 53



ISO 9126 - Functionality

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
	Suitability	Attributes of software that bear on the presence and appropriateness of a set of functions for specified tasks.
	Accurateness	Attributes of software that bear on the provision of right or agreed results or effects.
Functionality	Interoperability	Attributes of software that bear on its ability to interact with specified systems.
	Compliance	Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.
	Security	Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs or data.

Slide 54





ISO 9126 - Reliability

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
	Maturity	Attributes of software that bear on the frequency of failure by faults in the software.
Reliability	Fault tolerance	Attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface.
	Recoverability	Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.

Slide 55



ISO 9126 - Usability

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
	Understandability	Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.
Usability	Learnability	Attributes of software that bear on the users' effort for learning its application.
	Operability	Attributes of software that bear on the users' effort for operation and operation control.

Slide 56





ISO 9126 - Efficiency

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
Efficiency	Time behaviour	Attributes of software that bear on response and processing times and on throughput rates in performances its function.
	Resource behavior	Attributes of software that bear on the amount of resource used and the duration of such use in performing its function.

Slide 57



ISO 9126 - Maintainability

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
Maintainability	Analyzability	Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.
	Changeability	Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.
	Stability	Attributes of software that bear on the risk of unexpected effect of modifications.
	Testability	Attributes of software that bear on the effort needed for validating the modified software.

Slide 58





ISO 9126 - Portability

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Characteristics	Subcharacteristics	Definitions
Portability	Adaptability	Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.
	Installability	Attributes of software that bear on the effort needed to install the software in a specified environment.
	Conformance	Attributes of software that make the software adhere to standards or conventions relating to portability.
	Replaceability	Attributes of software that bear on opportunity and effort using it in the place of specified other software in the environment of that software.

Slide 59



ISO 9126 – Metriche

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

- Ad ogni sottocaratteristica è associato un insieme di metriche che la qualificano
- Presentiamo un esempio tratto da
 - “**CNIPA – Modelli qualità nelle forniture ICT**”
- **CNIPA = Centro Nazionale per l'Informatica nella Pubblica Amministrazione**
che è stato trasformato in
DigitPA - Ente nazionale per la digitalizzazione della pubblica amministrazione

Slide 60





Metriche per Funzionalità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
funzionalità/ adeguatezza	adeguatezza funzionale	misura il rapporto tra funzioni implementate e funzioni richieste	$X = 1 - A/B$ A = numero delle funzioni mancanti. B = Numero di funzioni previste nei requisiti
funzionalità/ accuratezza	accuratezza delle funzioni implementate	misura il rapporto tra funzioni implementate con il livello di accuratezza richiesto e le funzioni i cui requisiti prevedono specifici livelli di accuratezza	$X = A/B$ A = numero di funzioni implementate con il livello di accuratezza richiesto B = numero di funzioni i cui requisiti prevedono specifici livelli di accuratezza
funzionalità/ interoperabilità	Difetti nello scambio dei dati tra applicazioni	misura il numero di casi in cui la funzione di interfaccia provoca errori	$X = 1 - A/B$ A = numero di casi in cui è usata la funzione di interfaccia e sono rilevati errori. B = numero di casi in cui è usata la funzione di interfaccia
funzionalità/ conformità	conformità funzionale	Conta il numero di item per i quali è assicurata la conformità rispetto al numero di item per i quali è richiesta la conformità	$X = A/B$ A = numero di item per i quali è assicurata la conformità B = numero di item per i quali è richiesta la conformità
funzionalità/ sicurezza	controllo degli accessi	Conta il numero dei requisiti di controllo degli accessi implementati rispetto al numero dei requisiti di controllo degli accessi	$X = A/B$ A = numero dei requisiti di controllo degli accessi implementati B = numero dei requisiti di controllo degli accessi

Slide 61



Metriche per Affidabilità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
affidabilità/ maturità	tempo intercorrente tra due errori (MTBF)	Conta il numero di errori che si presentano durante un periodo definito e calcola l'intervallo medio tra due errori	$Y = T / A$ T = somma degli intervalli di tempo tra due errori A = numero totale di errori rilevati
affidabilità/ tolleranza	Malfunzioni che impediscono l'utilizzo del sistema	misura il rapporto tra le malfunzioni che producono la sospensione dell'uso del sistema ed il totale delle malfunzioni rilevate	$X = 1 - A/B$ A = numero di malfunzioni che producono la sospensione dell'uso del sistema B = numero di malfunzioni rilevate
affidabilità/ ripristinabilità	Disponibilità	misura la disponibilità del servizio in un determinato periodo di tempo	$X = \{ T_o / (T_o + T_r) \}$ T _o = tempo in cui il servizio è operativo T _r = tempo per le riparazioni

Slide 62





Metriche per Usabilità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
usabilità/ comprensibilità	completezza della descrizione	Misura la percentuale di funzioni che risultano comprensibili dopo la lettura della descrizione del prodotto/servizio	$X = A / B$ A = Numero di funzioni comprensibili B = Numero totale di funzioni
usabilità/ apprendibilità	facilità d'uso delle funzioni	Misura il tempo necessario per apprendere ad usare una funzione	T = tempo medio necessario per imparare ad utilizzare una funzione correttamente
usabilità/ operabilità	comprensibilità dei messaggi/ comunicazioni	misura il grado di comprensione dei messaggi/comunicazioni trasmessi in fase di utilizzo del prodotto/servizio	$X = A / UOT$ A = numero di volte che l'utente deve ripetere la stessa operazione in quanto non è in grado di operare correttamente a fronte di un messaggio UOT = periodo di osservazione

Slide 63



Metriche per Efficienza

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
efficienza/ efficienza temporale	tempo di risposta	misura il tempo medio di attesa tra la formulazione di una richiesta e la sua completa evasione	$X = T_{medio} / TX_{medio}$ $T_{medio} = \sum(T_i) / N$, (per $i=1$ a N) TX_{medio} = tempo medio di risposta richiesto
efficienza/ Utilizzazione delle risorse	per i prodotti sw si misura il grado di utilizzo di dispositivi di I/O, memoria, risorse di trasmissione. per i servizi si misura il grado di competenza e di efficienza del personale impiegato.	misura il numero di errori che si verificano in una unità di tempo simulando una condizione di massimo carico misura quanto l'utente valuta competente il personale preposto all'erogazione del servizio	$X = A / T$ A = numero di messaggi di errore T = tempo di osservazione Uso di questionari e metodi assegnazione del punteggio per la valutazione della competenza delle risorse

Slide 64





Metriche per Manutenibilità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
manutenibilità/ analizzabilità	supporto alla diagnosi	misura il grado di difficoltà nel diagnosticare malfunzionamenti	$X = A / B$ A = Numero di malfunzionamenti che possono essere diagnosticate con il supporto delle funzioni di diagnosi B = numero totale di malfunzionamenti rilevati
manutenibilità/ modificabilità	efficienza del ciclo di modifica	misura il tempo medio necessario a rimuovere un malfunzionamento	Tempo medio: $T_{av} = \text{Somma}(T_u) / N$ T_u = tempo intercorrente tra la richiesta di intervento ed il suo completamento N = numero di interventi richiesti
manutenibilità/ stabilità	stabilità a seguito di interventi	misura la percentuale di interventi di modifica che producono malfunzioni	$X = N_a / T_a$ N_a = Numero di interventi che hanno prodotto malfunzioni T_a = Tempo di erogazione durante il periodo di osservazione successivo all'intervento
manutenibilità/ testabilità	efficienza per i test	misura il tempo necessario per verificare l'effettiva rimozione di un malfunzionamento	$X = \text{Somma}(T) / N$ T = Tempo impiegate per assicurarsi l'effettiva risoluzione di un malfunzionamento N = numero di malfunzionamenti risolti

Slide 65



Metriche per Portabilità

Ingegneria del Software
Progettazione e Laboratorio
Qualità
Maurizio Pighin

Metriche di misura della qualità – 4/4			
caratteristica/ sottocaratt.	nome della metrica	descrizione della metrica	misura, formula e dati elementari
portabilità/ adattabilità	facilità di porting del software	misura la facilità di adattare il software all'ambiente	T = Tempo speso dall'utente per adattare il software all'ambiente
portabilità/ installabilità	facilità di installazione	misura la facilità di installazione del software all'ambiente operativo	$X = A / B$ A = Numero di casi in cui l'utente riesce a modificare le operazioni di installazione per esigenze specifiche B = Numero totale di casi in cui l'utente tenta di modificare le operazioni di installazione
portabilità/ coesistenza	Utilizzo concorrente con altri sistemi	misura il numero di malfunzioni nell'unità di tempo dovuti ad utilizzo concorrente con altri sistemi	$X = A/T$ A = numero di malfunzioni nel periodo T T = periodo di utilizzo in concorrenza con altri sistemi
portabilità/ sostituibilità	continuità nell'uso dei dati	misura la quantità di dati che è possibile usare passando dal precedente software senza effettuare interventi	$X = A / B$ A = numero di dati usati con il software da sostituire che possono essere usati senza interventi B = numero di dati usati con il software da sostituire che devono essere usati

Slide 66

