



# La Misurazione nel Software

Approfondimenti:  
Fenton, "Software Metrics a rigorous and practical approach"  
Capitolo 1,2,3.

# Perchè misurare? ...

⇒ **Quattro motivi principali per misurare processi, prodotti o risorse software:**

- ⇒ **Caratterizzare.** Comprendere i processi, i prodotti, le risorse, gli ambienti e **stabilire baselines** (valori soglia) di confronto per le valutazioni future.
- ⇒ **Valutare.** Determinare lo **stato di avanzamento** rispetto ai piani di progetto; stabilire il **raggiungimento degli obiettivi** di qualità e valutare l'impatto dei miglioramenti di processo e delle tecnologie sui prodotti e sui processi stessi

# ... Perchè misurare?

- Fare previsioni. In modo da poter pianificare opportunamente. Comprendere le relazioni/interdipendenze tra processi e prodotti; e costruire modelli basate su tali relazioni in modo che i valori che osserviamo per alcuni attributi possano essere utilizzati per predire gli altri. È il punto di partenza per estrapolare e identificare tendenze, stime di costo, tempo e qualità; diagnosticare i trade-off basati su dati storici
- Migliorare. raccogliere informazioni quantitative per identificare le inefficienze e altre opportunità per migliorare la qualità dei prodotti e le prestazioni di processo. Le misure aiutano a pianificare e monitorare gli sforzi di miglioramento

# Elementi della misurazione

- ⇒ **Misurazione:** processo attraverso cui numeri e simboli sono assegnati agli attributi di entità del mondo reale in modo da caratterizzare gli attributi attraverso regole chiaramente specificate.
- ⇒ Abbiamo bisogno di:
  - **Entità** (oggetto di interesse)
  - **Attributi** (caratteristiche delle entità)
  - **Regole/Metriche** (e scale) per assegnare i valori agli attributi
- ⇒ Perchè si possa misurare è necessario:
  - Chiaramente **identificare le entità** e gli **attributi** che saranno presi in considerazione e le **regole** usate per assegnare valori a ciascun attributo

# Elementi della misurazione: Entità

- ⇒ Elementi del mondo reale che desideriamo misurare
  - ❑ Products; processes, resources
  - ❑ Artifacts, activities, agents
  - ❑ Organizations, environments, constraints
  
- ⇒ Entità possono anche essere gruppi o insiemi di altre entità
  - ❑ subprocesses, flowpaths, by-products

# Elementi della misurazione: **Attributi**

⇒ **Caratteristiche** o proprietà delle entità

- Person (entity)
  - Characteristics: height, color of eyes, IQ, age, years of experience ...
- Software (entity)
  - Characteristics: size, cost, elapsed time, effort, response time, transaction rates, number of defects ...

⇒ L'arte della misurazione consiste nel **decidere quali attributi usare** in modo da fornire una chiara idea delle entità considerate

Ciascuna entità è legata **agli attributi** che caratterizzano l'entità e **alle misure** che possono esser usate per quantificare gli attributi stessi

Resource Entities	Attributes	Possible Measures
assigned staff	team size	number of people assigned
	experience	years of domain experience years of programming experience
CASE tools	type	name of type
	is_used?	yes/no (a binary classification)
time	start date, due date	calendar dates
	elapsed time	days

Product Entities	Attributes	Possible Measures
system	size	number of modules number of bubbles in a data-flow diagram number of function points number of physical source lines of code number of memory bytes or words required (or allocated)
	defect density	defects per KSLOC defects per function point
module	length	physical source lines of code logical source statements
	percent reused	ratio of unchanged physical lines to total physical lines, comments and blanks excluded
unit	number of linearly independent flowpaths	McCabe complexity
document	length	number of pages
line of code	statement type	type names
	how produced	name of production method
	programming language	language name
defect	type	type names
	origin	name of activity where introduced
	severity	an ordered set of severity classes
	effort to fix	staff-hours
	age (of open defects)	elapsed time (days) since receipt of defect report

Process Entities	Attributes	Possible Measures
development process	elapsed time	calendar days working days
	milestones	calendar dates
	development effort	staff-hours, days, or months
	phase containment	percent of total defects found in phase where introduced
	process compliance	percent of tasks complying with standard procedures or directives
	performance	number of tests passed divided by number of tests executed
detailed designing	elapsed time	calendar days working days
	design quality	defect density: number of design defects found in down-stream activities divided by a measure of product size, such as function points or physical source lines of code.
testing	volume	number of tests scheduled
	progress	number of tests executed number of tests passed
maintenance	cost	dollars per year staff-hours per change request
change request backlog	size	number of change requests awaiting service estimated effort (staff-hours) for pending requests



# Scale di Misurazione

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](http://serlab.di.uniba.it)

# Scale di Misurazione...

- ⇒ La scala di misurazione fornisce i valori e le unità per descrivere gli attributi
  - Person
    - height: 68 inches
    - weight: 163 pounds
    - Eyes: brown
    - Disposition: aggressive
  - Software Project:
    - Produce 30 000 LOC
    - Planned completion date: August 30
    - Use: 11 345 staff hrs of effort
    - Application type: real time command and control
- ⇒ Ovunque si effettui una misurazione e qualunque sia la sua forma, richiede sempre che vi siano delle **scale ben definite** per rilevare e registrare i valori misurati

# ... Scale di Misurazione ...

- ⇒ **Nominale:** ogni valore della scala rappresenta **una classe o una categoria** di appartenenza
  - Non c'è **ordinamento** tra i valori
  - Non c'è **significato di grandezza** associato ad ogni classe
- ⇒ Le scale nominali pongono gli elementi in **uno schema di classificazione (categorie di appartenenza)**. Le classi non sono ordinate anche se sono numerate (solo per scopi di identificazione) da 1-n.

Phase in which a fault is injected	
$m_1$	$m_2$
1 if specification fault	A if specification fault
2 if design fault	D if design fault
3 if code fault	C if code fault

# ... Scale di Misurazione ...

- ⇒ **Ordinale:** Oltre ad essere nominale esprime un ordine (ascendente o discendente) tra tutti i possibili valori dell'osservabile. La distanza tra un elemento e l'altro della scala non ha significato.
- Le classi di appartenenza dei valori sono ordinate rispetto all'attributo
  - I valori esprimono solo un punteggio (rank), pertanto non ha senso applicare su essi gli operatori aritmetici

Complexity of a Software Module		
$m_1$	$m_2$	$M_3$ not a valid measure!
1 if simple	1 simple	1 simple
2 if moderate	100 moderate	3 moderate
3 if high	1000 high	2 high
4 if incomprehensible	11000 incomprehensible	4 incomprehensible

# ... Scale di Misurazione ...

- ⇒ **Intervallo:** L'intervallo **esprime sia l'ordinamento, sia la misura dell'intervallo che separa le classi della scala**
  - Preserva l'ordine come per una scala ordinale
  - Preserva differenze ma non i rapporti (divisioni). Pertanto, Ai valori di questa scala sono applicabili somma e sottrazione; non hanno senso la moltiplicazioni e la divisione
  - Non c'è un origine nella scala (valore di zero assoluto)
    - If the temperature reaches 40°C today and 20°C yesterday, we *cannot say that it was twice as warm today as it was yesterday.*
    - The interval from moving from one degree to another is the same, and each degree is considered a class related to heat: Moving from 20-> 21 °C increases the heat the same way that moving from 40 ->41°C.
    - Since there is **no true-zero value** on the scale, however we cannot say that it is 50% hotter today than yesterday
- ⇒ Eg.: clock times, calendar dates, temperature in °C or °F

# ... Scale di Misurazione ...

- ⇒ **Ratio:** aggiunge un origine alla scala (un elemento zero - null)
  - Preserva l'ordinamento, la dimensione degli intervalli, il rapporto tra entità (come nelle scale precedenti)
  - (in più) Esiste un **elemento zero significativo**, che indica l'assenza di quell'attributo
  - Il valore della misura, a partire da zero, deve aumentare di intervalli uguali, detti **unità della misura**
  - **Tutte le operazioni aritmetiche sono significative**
  - Due scale ratio M ed M' sullo stesso attributo (es. dimensione) sono **convertibili** con la relazione  $M=a*M'$  essendo "a" una costante
    - From Feet to Inches:  $I=12*F$
    - From KLOC to LOC:  $KLOC=1000*LOC$
    - $M = \text{LOC}$ ;  $M' = \text{length as number of characters}$  ->  $M'=a*M$  where  $a=\text{average number of characters per LOC}$
- ⇒ Eg. Productivity, defect density, integration cost, development cost, effort, etc.
- ⇒ Esempio: la **dimensione** di un programma misurata **in LOC** è una scala Ratio ( $m_1$ ); la **dimensione in KLOC** è un'altra scala ratio ( $m_2$ );  $m_2=1000*m_1$

# ... Scale di Misurazione ...

- ⇒ **Assoluta:** caso particolare della scala ratio dove l'unico moltiplicatore ammissibile è 1.
  - La misurazione nella scala assoluta è fatta semplicemente "conteggiando" gli elementi di una entità (es. Nr. Persone in un aula)
  - L'attributo ha sempre forma del tipo "numero di occorrenze di x in una entità"
  - Esiste solo un modo per mappare la misurazione: **contando**
  - Tutte le operazioni aritmetiche sono significative
- ⇒ Eg. Number of failures observed during integration testing; number of people working on a software project; number of design defects; ...

# ... Scale di Misurazione

Scale type	Admissible transformations (how measures $M$ and $M'$ must be related)	Examples
Nominal	1–1 mapping from $M$ to $M'$	Labeling, classifying entities
Ordinal	Monotonic increasing function from $M$ to $M'$ , that is, $M(x) \geq M(y)$ implies $M'(x) \geq M'(y)$	Preference, hardness, air quality, intelligence tests (raw scores)
Interval	$M' = aM + b$ ( $a > 0$ )	Relative time, temperature (Fahrenheit, Celsius), intelligence tests (standardized scores)
Ratio	$M' = aM$ ( $a > 0$ )	Time interval, length, temperature (Kelvin)
Absolute	$M' = M$	Counting entities



# Scale delle Misure e Significatività delle Misurazioni

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](http://serlab.di.uniba.it)

# Significatività delle Funzioni Statistiche

- ⇒ Ha significato applicare una funzione statistica ad un tipo di scala se le **affermazioni** dedotte dall'applicazioni delle funzioni sono **invarianti rispetto alla scala utilizzata**.

Tipo scala	Funzione statistica significativa
Nominale	Moda
Ordinal	Moda Mediana
Intervallo	Moda Mediana Media
Ratio	Tutte le funzioni

# Funzioni Statistiche

⇒ Data una lista di valori:

- **Media** : la somma dei valori diviso per il numero di valori nella lista
- **Mediana** : il valore centrale nella lista ordinata
- **Moda** : il valore che si presenta più frequentemente nella lista

⇒ Esempio: 2,2,4,5,5,8,8,10,11,11,11,15,16

- Media : 9.1
- Mediana : 8
- Moda : 11

## Esempio : Classificazione dei problemi

⇒ La **moda** è sempre **valida**; sia che le due scale fossero **nominali** sia che fossero **ordinali**

Osservabile	Classe (lettera)	Classe (numeri)
$X_1$	A	1
$X_2$	B	5
$X_3$	B	5
$X_4$	C	3
$X_5$	B	5
Moda	$B(X_2, X_3, X_5)$	$5(X_2, X_3, X_5)$

# Esempio: Comprensibilità dei programmi

⇒ La mediana sarebbe **valida** se le due scale fossero **ordinali**, non lo sarebbe se fossero **nominali**

Osservazione	Comprensibilità	Comprensibilità
$X_1$	Banale	1
$X_2$	Moderato	4
$X_3$	Semplice	3
$X_4$	Moderato	4
$X_5$	Incomprensibile	10
Moda	Moderato( $X_2, X_4$ )	4( $X_2, X_4$ )
Mediana	Semplice( $X_3$ )/Indeterm.	3( $X_3$ )/Indeterminata

## Le scale di misurazione possono cambiare man mano che si matura esperienza

- ⇒ Measurements **can progress from lower to higher scales** as societies, organizations and practices mature
- ⇒ Look for opportunities that allow to evolve measurement practices toward scales that provide greater information

*We can imagine, for example, that certain Eskimos might speak of temperature only as freezing or not freezing and, thereby, place it on a nominal scale. Others might try to express degrees of warmer and colder, perhaps in terms of some series of natural events, and thereby achieve an ordinal scale. As we all know, temperature became an interval scale with the development of thermometry, and, after thermodynamics had used the expansion ratio of gases to extrapolate to zero, it became a ratio scale."*

— S. S. Stevens, 1956

# Misure Soggettive e Oggettive ...

- ⇒ Quando si misurano gli attributi di entità ci sforziamo di manterere le misurazioni il più possibile oggettive
  - Durante il processo di misurazione, assicuriamo che persone diverse producano le stesse misure, indipendentemente dal fatto che stanno misurando prodotti, processi o risorse.
  - Alcune misure sono più soggettive di altre
- ⇒ **Misure Soggettive:** dipendono dal misuratore, sono strettamente legate al contesto in cui sono rilevate
  - Possono variare al variare del misuratore e dipendere dalla sua opinione
    - Programmer experience (high, medium, low)
    - Quality of requirements (1-5 scale)
- ⇒ **Misure Oggettive:**
  - Sono indipendenti dal misuratore. Rilevate oggettivamente, indipendentemente sulla entità, e dunque il valore non è influenzato
    - Development time
    - Number of LOC of a program / module
    - Number of defects
    - Productivity

# ... Misure Soggettive e Oggettive ...

- ⇒ Al fine di “**limitare e contenere**” la soggettività delle misure soggettive si definisce e associa un **Modello Descrittivo**, alla metrica.
- Ogni **valore della scala** è definito con una **descrizione testuale** per meglio chiarire e **spiegare i range dei valori** e **ridurre la interpretazione soggettiva delle informazioni**.
  - Criticality of requirements (1-5 point scale)
    - 1. **not important** (it does not impact on any user)
    - 2. **low importance** (impacts on few users)
    - 3. **average importance** (impacts on several users but does not determine consequences if it is not respected)
    - 4. **important** (impacts on several users and determines consequences if it is not respected)
    - 5. **critical** (if it is not respected, the system cannot be used)

# ... Misure Soggettive e Oggettive

- ⇒ In caso di misure Oggettive è necessario fornire un **Modello Quantitativo** (equazioni, modello di calcolo, formula) associato alla metrica
  - ❑ Non conformance = (nr of violations / nr of times the techniques have been applied) \* 100
  - ❑ Temperature Range
    - Low: in summer between 8-12 °C; in winter from -X to 11°C
    - High: in summer above 13°C; in winter from 12°C and above

# Misure Dirette (osservate) e Indirette (calcolate)

- ⇒ Una volta ottenuto il **modello** delle entità e degli attributi implicati (i.e. come gli attributi sono mappati e usati per misurare una specifica entità) è possibile definire le misure sulla base del modello
  - ❑ Measure height in cm
  - ❑ Measure dimension of a room in square metres
  - ❑ Measure dimension of a program in LOC
- ⇒ **Misurazione diretta** di un attributo di una entità non implica altri attributi o caratteristiche in quanto è rilevato/osservato direttamente
  - ❑ Length of source code (measured by lines of code)
  - ❑ Duration of testing phase (measured by elapsed time in hrs)
  - ❑ Number of defects discovered during the testing process (measured by counting the defects)
  - ❑ Time a programmer spends on a project (measured by months worked)
- ⇒ **Misurazione indiretta** implica che un attributo sia misurato combinando misure in relazione tra loro. Necessità di un modello di calcolo (Modello Quantitativo)
  - ❑ Programmer productivity: LOC produced / person months of effort
  - ❑ Module defect density: number of defects / module size
  - ❑ System spoilage: effort spent for fixing defects / total project effort

# Riepilogando: Modelli Descrittivi

- ⇒ Ogni valore del **Range** della misura è **definito con una descrizione testuale**.
- ⇒ Esempi
  - **Tipo di Test** può assumere i seguenti valori:
    - **Test strutturale**, in caso di componente ad alta complessità procedurale, usa la struttura interna del codice
    - **Test per valori limite**, in caso di componenti che hanno bassa complessità procedurale, usa le specifiche della componente che tratta come una black - box
  - **Criticità della Richiesta** può assumere uno dei valori del seguente punteggio(soggettivo)
    1. **Non Importante**, anche se non è rispettata non danneggia alcun utente
    2. **Scarsamente Importante**, impatta su pochi utenti
    3. **Mediamente Importante**, impatta su molti utenti, ma non è dannosa se non è rispettata.
    4. **Importante**, impatta su molti utenti ed è dannosa se non è rispettata.
    5. **Critica**, se non è rispettata il sistema non si può utilizzare

# Modelli Quantitativi

- ⇒ Il valore di una misura è calcolato da altre attraverso uno dei seguenti modelli quantitativi
  - ❑ Calcolo con equazione
  - ❑ Trasformazione con corrispondenze tabulari
  - ❑ Calcolo Statistico
- ⇒ Esempi:
  - ❑ Non Conformità = (Numero di violazioni/numero di applicazioni delle tecniche)\*100
  - ❑ Media dei voti degli studenti di ITPS iscritti al terzo anno



# La Qualità è Relativa

Dipartimento di Informatica - Università degli Studi di Bari

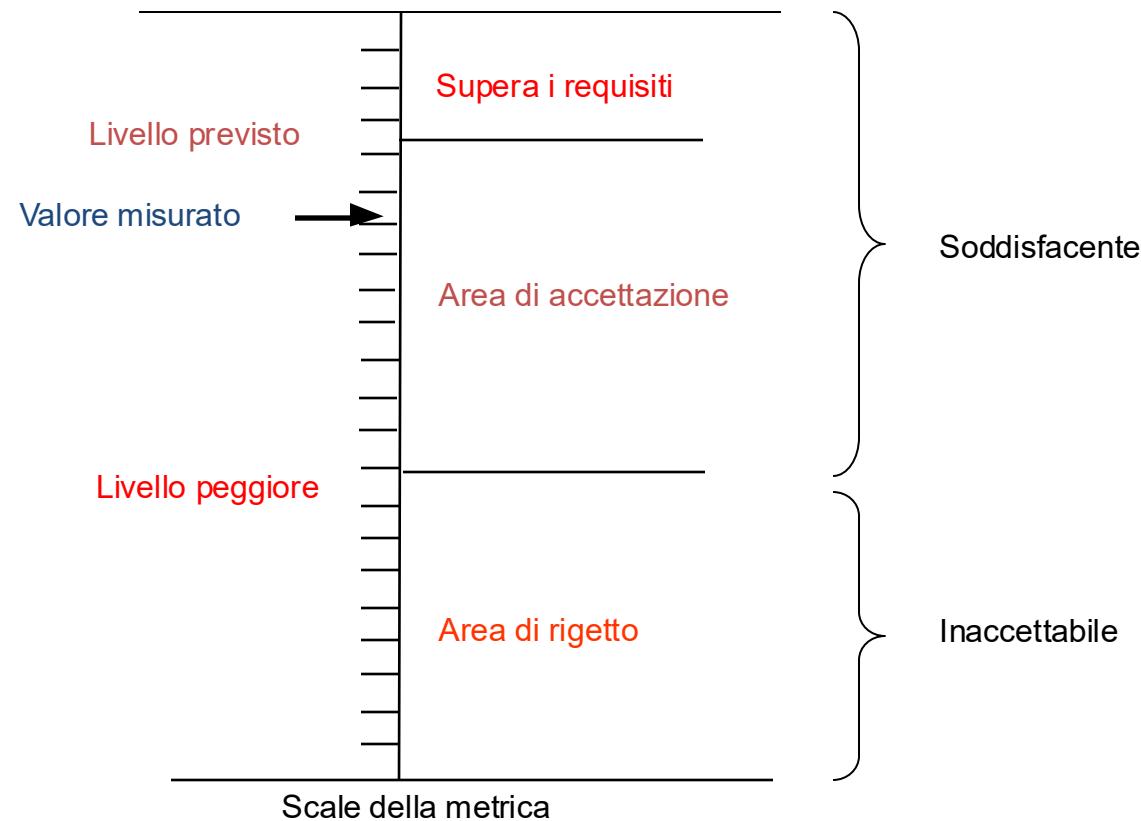
Via Orabona, 4 - 70125 - Bari

Tel: +39.080.5443270 | Fax: +39.080.5442536

[serlab.di.uniba.it](http://serlab.di.uniba.it)

# Modello per la Valutazione secondo la ISO IEC 14598

La qualità è relativa



# Soglie dei Valori

- ⇒ Per soddisfare gli obiettivi di qualità richiesti le misure rilevate sui prodotti e sui processi devono raggiungere predefinite soglie
- ⇒ Le soglie da raggiungere sono dipendenti dal contesto di esecuzione del processo e dal committente del prodotto
- ⇒ Normalmente si definisce
  - un intervallo di successo entro i quali il processo od il prodotto si considera abbia raggiunto l'obiettivo:  $S_{\min}$ ;  $S_{\max}$
  - Una soglia di insuccesso al di sotto della quale il processo od il prodotto è rigettato:  $S_{\text{rigetto}}$

# Miglioramento dei Processi e dei Prodotti

- ⇒ Se si è raggiunti la **soglia di insuccesso** è **indispensabile individuare ed eseguire iniziative di miglioramento**
- ⇒ Se si è **nell'intervallo di successo** il management può decidere:
  - di **migliorare il processo** per far salire le soglie verso  $S_{max}$
  - di **stringere l'intervallo di successo** per aumentare la **capacità del processo**
- ⇒ Se si sono **raggiunte** per tutte le misure **le soglie massime** ed i **processi sono stabili** il management può decidere di far salire il loro valore per indurre **maggior maturità del contesto di esecuzione**