

# OSS: come misurare il software



Gianfranco Lanza - GUFPI-ISMA

26 Giugno 2015

# AGENDA

- Chi è il GUFPI-ISMA
- Misure funzionali e non funzionali
- Function Point
- SNAP Point
- Come utilizzare i numeri
- OSS: cosa misurare?
- Riuso del software: i worked FP
- Esempio
- Conclusioni

# GUFPI-ISMA

**Gruppo Utenti Function Point Italia Italian Software Metrics Association**  
**Il GUFPI-ISMA è l'associazione italiana per la promozione, la diffusione e lo sviluppo delle tecniche di misurazione funzionale e non funzionale del software.**



# Partnership e Collaborazioni

## In-Sw Measurement



## Out-Sw Measurement



# Misure funzionali e non funzionali

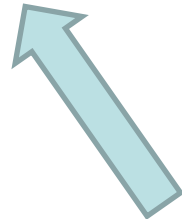
FUR (Functional User Requirements) (norma ISO/IEC 14143):

Quante funzionalità offre il software?



SNAP Point

Function Point

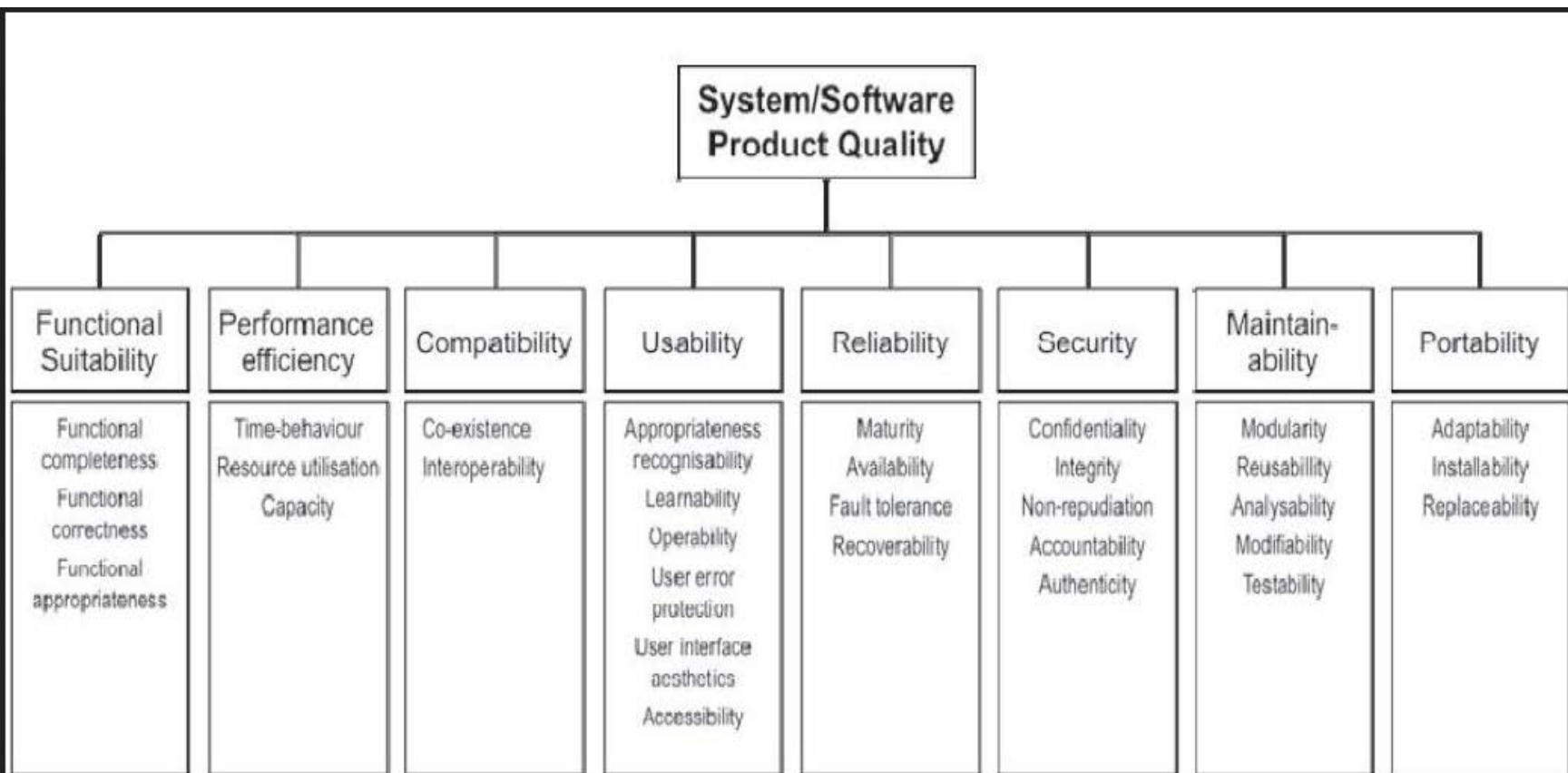


NFR (Non Functional Requirements) (norma ISO/IEC 25010 - ex 9126):

Quali sono le caratteristiche del software?

# Misure funzionali e non funzionali

## Norma ISO/IEC 25000



# Misure funzionali e non funzionali

## Norma ISO/IEC 25000 : strumenti



Home | Quality Models | Squale Application

WELCOME!

### Main

Home  
Introduction  
Screenshots  
Discovering Squale  
A Competitive Cluster Project  
Sponsors  
Releases  
News

### Downloads

Squale Application  
Quality Models

### Documentation

For Squale users

### Quick links

License  
Team  
Source

## WELCOME TO SOFTWARE QUALITY ENHANCEMENT!



*Assist developers in improving the code of their projects.*

*Help project managers to meet quality requirements for their applications.*

*Give top-managers dashboards to monitor the overall health of their information system.*

**This is what the Squale project is all about.**

**26 mai 2011** : we are pleased to announce

- **Squale 7.1!** Please go to the [release notes](#) to know more about this version and give Squale a try with its **standalone version!**

### Squale in a nutshell

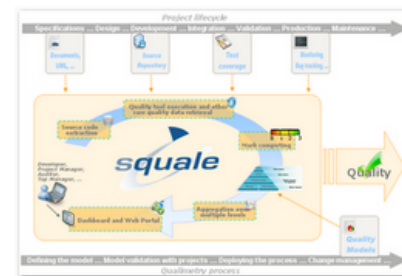
To help you deal with the quality of your software developments, the open source **Software QUALity Enhancement** project (aka Squale) focuses on two main aspects:

- **Works on enhanced quality models**
  - inspired by existing standards (ISO-9126) and approaches (GQM, McCall),
  - validated and improved by famous researchers who are part of the Squale team,
  - dealing of both technical and economical aspects of quality,
- **Development of an open-source application that helps assessing software quality and improving it over time**
  - based on third party technologies (*commercial or open-source*) that produce raw quality information (like metrics for instance),
  - using the quality models to aggregate this raw information into high level quality factors,
  - all this targeting different languages, including Java, C/C++, .NET, PHP, Cobol, ...

### Squale is an open-source project!



Squale Application is developed and distributed under the terms of the [GNU Lesser General Public License](#) version 3.



# Misure funzionali e non funzionali

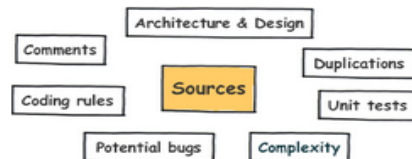
## Norma ISO/IEC 25000 : strumenti

The image shows the top part of the SonarQube website. It features a navigation bar with links: Download, Features, Get Support, Get Involved, Development, Roadmap, Resources, Blog, and Company. Below the navigation bar is a main banner with the text "Put your technical debt under control" and "Productivity is falling? Confess your source code to clean it up!". To the right of the banner is a screenshot of the SonarQube dashboard, which displays various metrics like code quality, bugs, and technical debt.

SonarQube™ software (previously known as "Sonar") is an open source project hosted at [Codehaus](#).  
[Download](#) and [install](#) your own copy. Version: 5.1 (Apr 2, 2015) distributed under [license LGPL v3](#).

### All in one

SonarQube is an [open](#) platform to manage code quality. As such, it covers the 7 axes of code quality:



### Extend with plugins

Covering new languages, adding rules engines, computing advanced metrics can be done through a powerful extension mechanism. More than [50 plugins](#) are already available.

### Languages covered

More than 20 programming languages are covered [through plugins](#), including Java, C#, C/C++, PL/SQL, Cobol, ABAP...

### In 3 clicks

SonarQube has got a very efficient way of navigating, a balance between high-level view, dashboard, TimeMachine and defect hunting tools. This enables to quickly uncover projects and / or components that are in Technical Debt to establish action plans.



### Quality is central

SonarQube is a web-based application. Rules, alerts, thresholds, exclusions, settings... can be configured online. By leveraging its database, SonarQube not only allows to combine metrics altogether but also to mix them with historical measures.

### Get started

1. [Download](#)
2. Unzip and start
3. [Analyze](#) projects
4. Ready to [improve quality](#)

### SonarQube™ in action



To see more, visit [Nemo](#), the online instance of SonarQube dedicated to open source projects.

### Extend SonarQube™

- Install [plugins](#)
- Integrate to [Continuous Integration servers](#)
- Integrate to [Eclipse](#)

### Powered by SonarSource™

SonarSource proposes [commercial extensions](#) to cover additional languages and manage portfolios of projects along with [Professional services](#).



# Misure funzionali e non funzionali

## Norma ISO/IEC 25000 : strumenti



« SpagoWorld | acquista | accesso clienti | download | registrazione comunità | contatti | en.it

spago4Q

HOME PIATTAFORMA SERVIZI COMUNITÀ ABOUT

SUPPORTO CONSULENZA FORMAZIONE

# Spago4Q

the OPEN SOURCE platform to measure, analyze and monitor the QUALITY of products, processes and services

SPAGO4Q IN PILLOLE OPEN SOURCE

demo on-line download documentazione

### Spago4Q in primo piano

#### Spago4Q a supporto della Productivity Intelligence

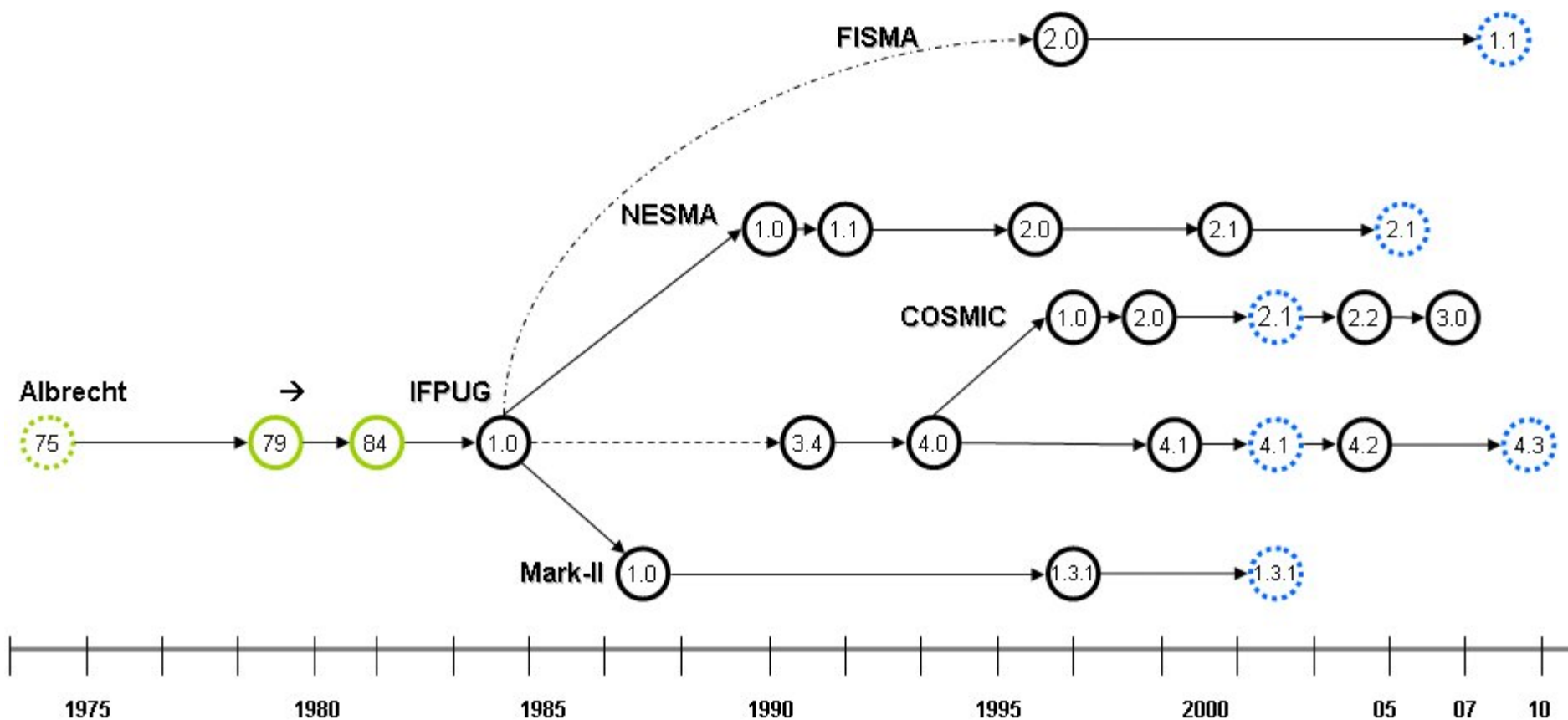
 Spago4Q consente ad Engineering, nell'ambito dei propri Software Labs, di misurare la qualità dei processi di sviluppo del software tramite il modello Productivity Intelligence, il quale fa emergere la Qualità dalla combinazione di 3 dimensioni di analisi: benefici Economici, aspetti Tecnici, impatto Sociale. [Leggi la Success Story](#)

### Esperienze

- Productivity Intelligence: un approccio completo per misurare la qualità del software con Spago4Q
- Spago4Q a supporto del miglioramento della qualità dei servizi per il mercato finance
- Spago4Q in Regione Veneto - Sistema di Governance Operativa
- Spago4Q per Consip - Ministero dell'Economia e delle Finanze
- Spago4Q per la Commissione Europea – Direzione Generale per le Politiche Regionali
- Spago4Q (SpagoBI for Quality) nel Portale di Project Management del Gruppo Engineering

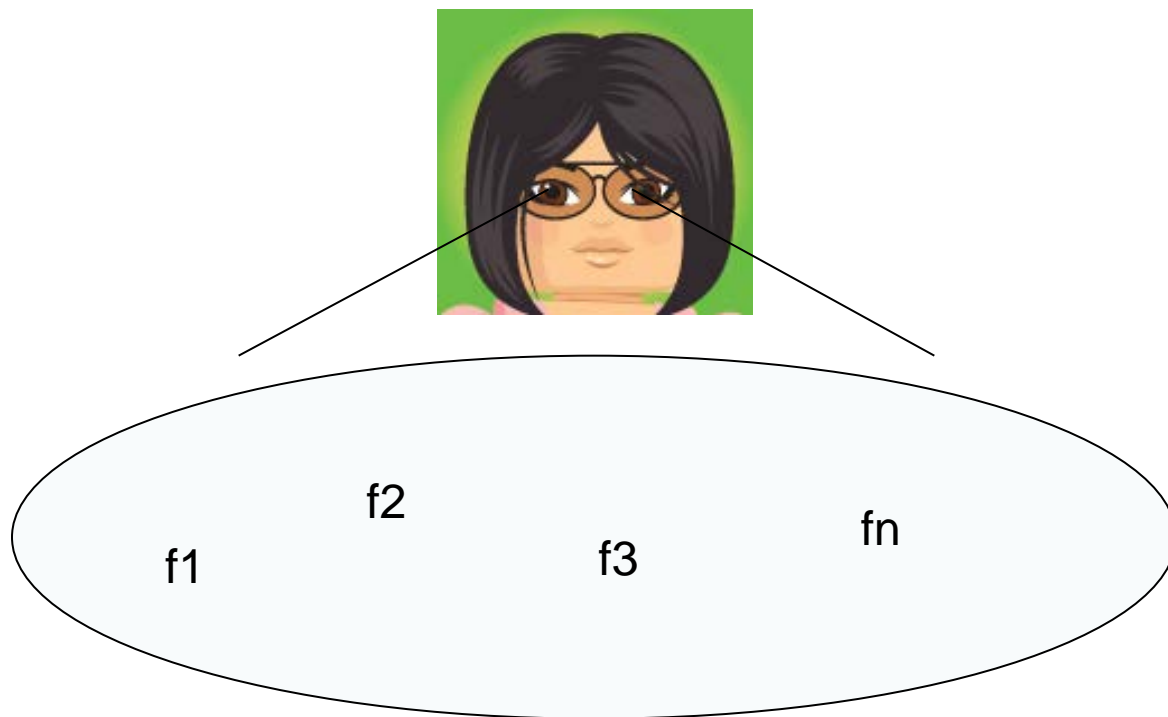
[Continua](#)

# Function Point



# Function Point

Ogni Prodotto software è misurato analizzando i requisiti funzionali del software secondo il punto di vista dell'utente.



# IFPUG Function Point

Secondo il metodo IFPUG sono identificate 5 funzioni e ognuna di esse porta un numero di fp in base ad una complessità

ILF: entità logiche gestite (inserimento, modifica, cancellazione..)

EIF: Entità logiche referenziate ma gestite da un altro prodotto

EI: Input dell'applicazione (videate di input, file, etc.)

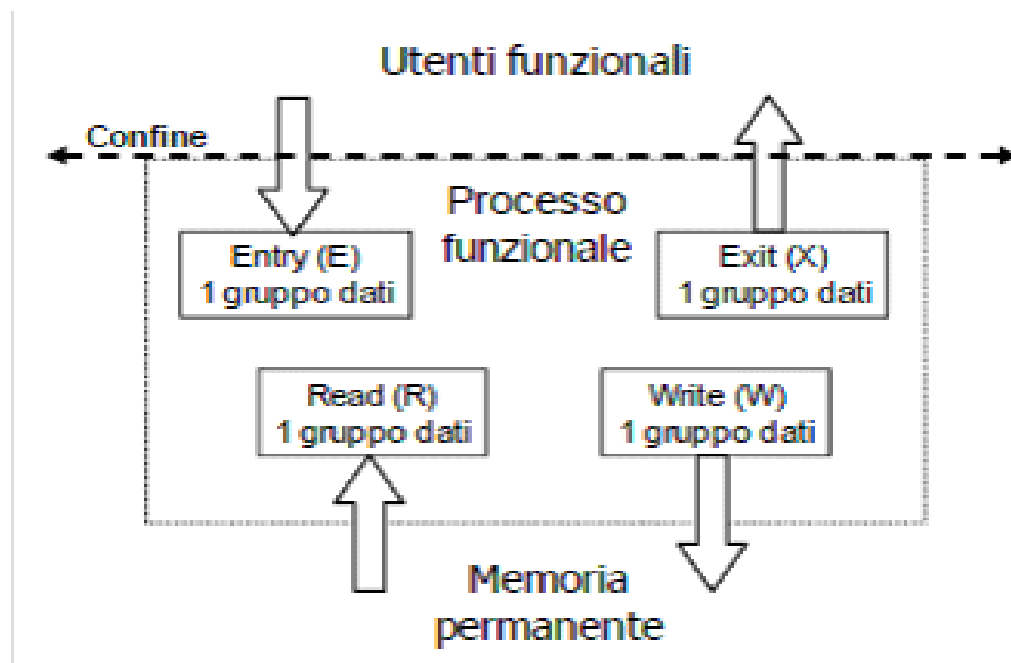
EQ: Presentazione di dati (Ricerche, liste di informazioni, etc.)

EO: Output di dati (report, statistiche ,invio dati derivati, etc.)

|       | ILF | EIF | EI | EO | EQ |
|-------|-----|-----|----|----|----|
| Bassa | 7   | 5   | 3  | 4  | 3  |
| Media | 10  | 7   | 4  | 5  | 4  |
| Alta  | 15  | 10  | 6  | 7  | 6  |

# COSMIC Function Point

Secondo il metodo COSMIC si identificano 4 movimenti dati, ciascuno di peso 1 FP. Metrica che si adatta meglio ad ambiti quali real-time, batch, software a strati.



# SNAP Point

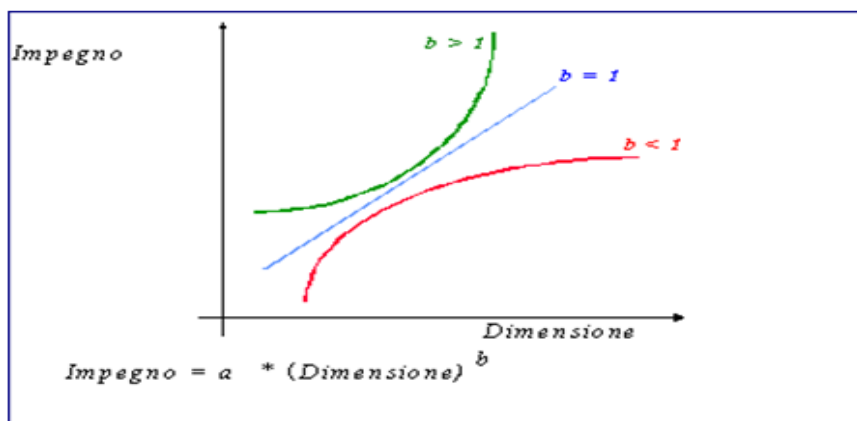
Metodo che permette di dare un peso non funzionale al prodotto secondo differenti categorie

| Data Operations   | Interface Design   | Technical Environment  | Architecture  |
|---|--|--|---|
| <ul style="list-style-type: none"><li>• 1.1 Data Entry Validation</li><li>• 1.2 Logical and Mathematical Operations</li><li>• 1.3 Data Formatting</li><li>• 1.4 Internal Data Movements</li><li>• 1.5 Delivering Added Value to Users by Data Configuration</li></ul> | <ul style="list-style-type: none"><li>• 2.1 UI Changes</li><li>• 2.2 Help Methods</li><li>• 2.3 Multiple Input Methods</li><li>• 2.4 Multiple Output Methods</li></ul> | <ul style="list-style-type: none"><li>• 3.1 Multiple Platforms</li><li>• 3.2 Database Technology</li><li>• 3.3 Batch Processes</li></ul> | <ul style="list-style-type: none"><li>• 4.1 Component based software</li><li>• 4.2 Multiple Input / Output Interfaces</li></ul> |

# Come utilizzare i numeri

*Non si può governare ciò che non si controlla,  
non si può controllare ciò che non si misura  
(Tom De Marco)*

## Equazione dell'impegno



Esempio:  
 $Impegno = 2,23 * (FP)^{0,869}$

# Come utilizzare i numeri

Concetto di produttività: Quanto lavoro faccio nell'unità di tempo FP/gg



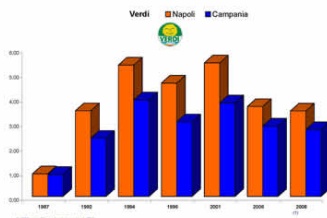


# Come utilizzare i numeri

Come in una tappa ciclistica la velocità media non indica la velocità nei vari tratti (salita, discesa, pianura), così anche nello sviluppo del software la velocità di sviluppo può cambiare sensibilmente.



# Come utilizzare i numeri



- Previsione dell'effort di sviluppo del software (produttività: FP/GG)
- Qualità del prodotto: Difetti/FP
- Gestione dei change request su un prodotto (Function Point lavorati)
- Dimensionamento di un asset applicativo
- Differenziazione tipologie di prodotti in funzione della loro complessità.
- Gestione contratti cliente/fornitore in base a dati oggettivi

# OSS: cosa misurare?

Modello DeLone-McLean: dare valore a un OSS

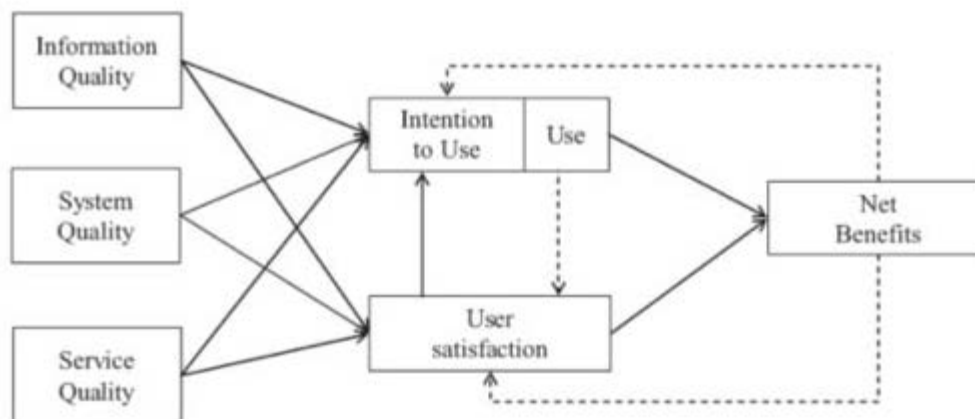


Figure 1: DeLone and McLean (2003) information system success model.

Dal punto di vista prettamente del software cosa può aver senso misurare e conoscere?

# OSS: cosa misurare?

Quale è la misura funzionale del prodotto?

Quale è la sua difettosità?

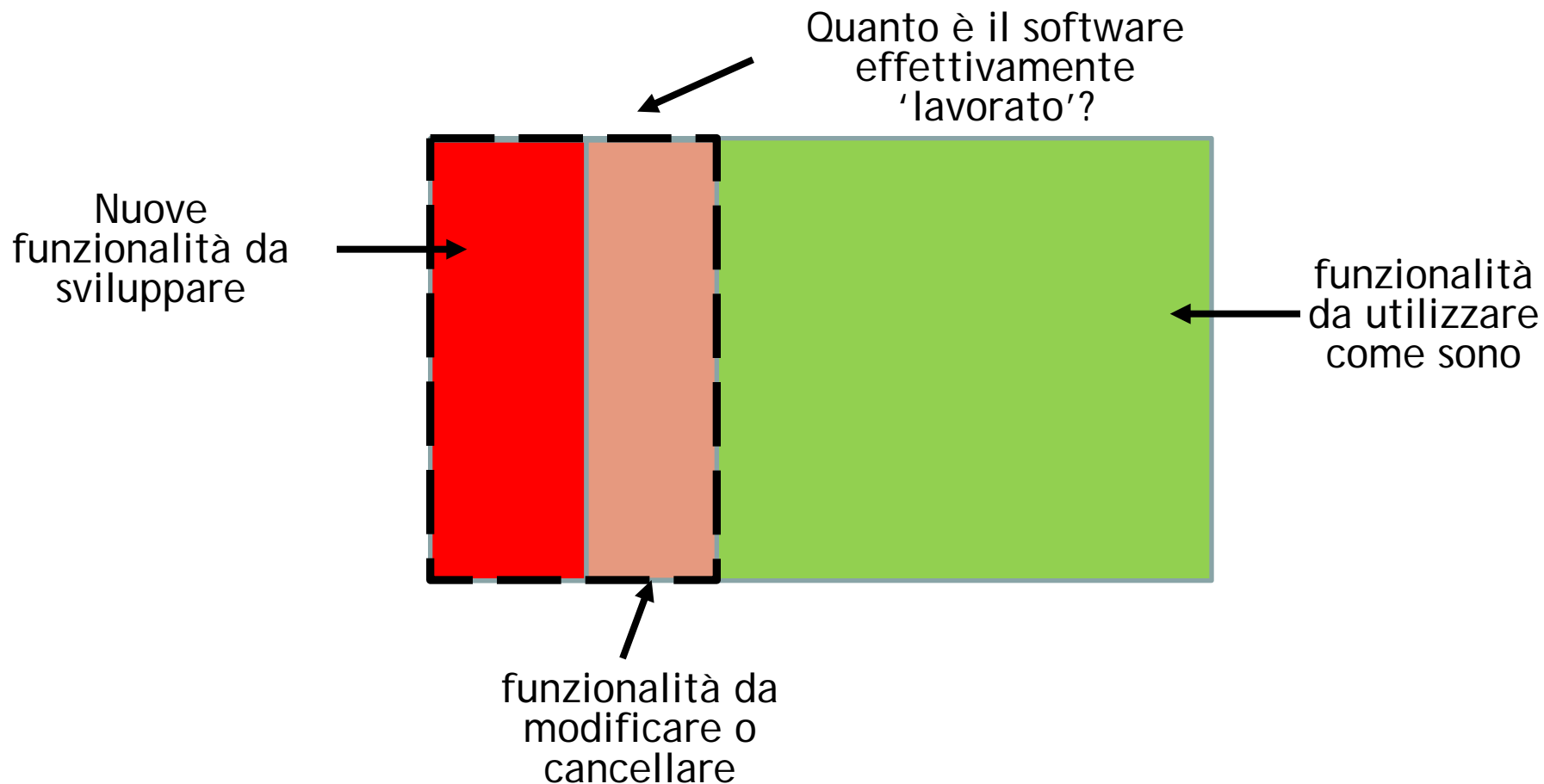
Quale possibile l'effort di sviluppo?

Quale è il peso della eventuale personalizzazione?

Quale quello dei change request?



# Riuso del software: i Worked FP



# Esempio

## SISTEMA INFORMATIVO PER LE CERTIFICAZIONI ENERGETICHE DEGLI EDIFICI (SICEE)

Sistemapiemonte

Profilo personale 

nuovo utente? registrati »

📍 SERVIZI PER PRIVATI » AMBIENTE E ENERGIA » ENERGIA


### Sistema Informativo per le Certificazioni Energetiche degli Edifici (SICEE)


Presentazione


Novità 

Servizi software-house

FAQ per l'utilizzo di SICEE

FAQ sulla normativa 

Guida all'iscrizione 

Guida all'utilizzo di SICEE 



Si avvisa di possibili disservizi per gli applicativi del settore Energia (SICEE) nella giornata del 23 Maggio dalle ore 17:00 alle ore 24:00 per manutenzione ordinaria dell'infrastruttura.



Si segnala che sulla pagina <http://www.docet.itc.cnr.it/> aggiornata al 30/10/2014 è stata confermata la non adeguatezza del software Docet alle attuali metodologie di calcolo. Tale inadeguatezza non ne consente l'uso per la redazione degli attestati e quindi, al fine di evitare contestazioni e sanzioni, si prega di prendere nota che, a far data dal 2 ottobre 2014, effettuare valutazioni della prestazione energetica mediante il Docet non è più ammesso. Per maggiori approfondimenti visitare il [sito dei produttori del software](#).

Utenti del servizio

ACCEDI AL SERVIZIO >

Servizio ad accesso libero per cittadini e notai.

ACCEDI AL SERVIZIO >

Il servizio è riservato ai certificatori ed è soggetto ad autenticazione tramite certificato digitale rilasciato da una Certification Authority (CA) accreditata presso l'Agenzia per l'identità digitale (AGid, ex DigitPa)

# Esempio

## SISTEMA INFORMATIVO PER LE CERTIFICAZIONI ENERGETICHE DEGLI EDIFICI (SICEE)

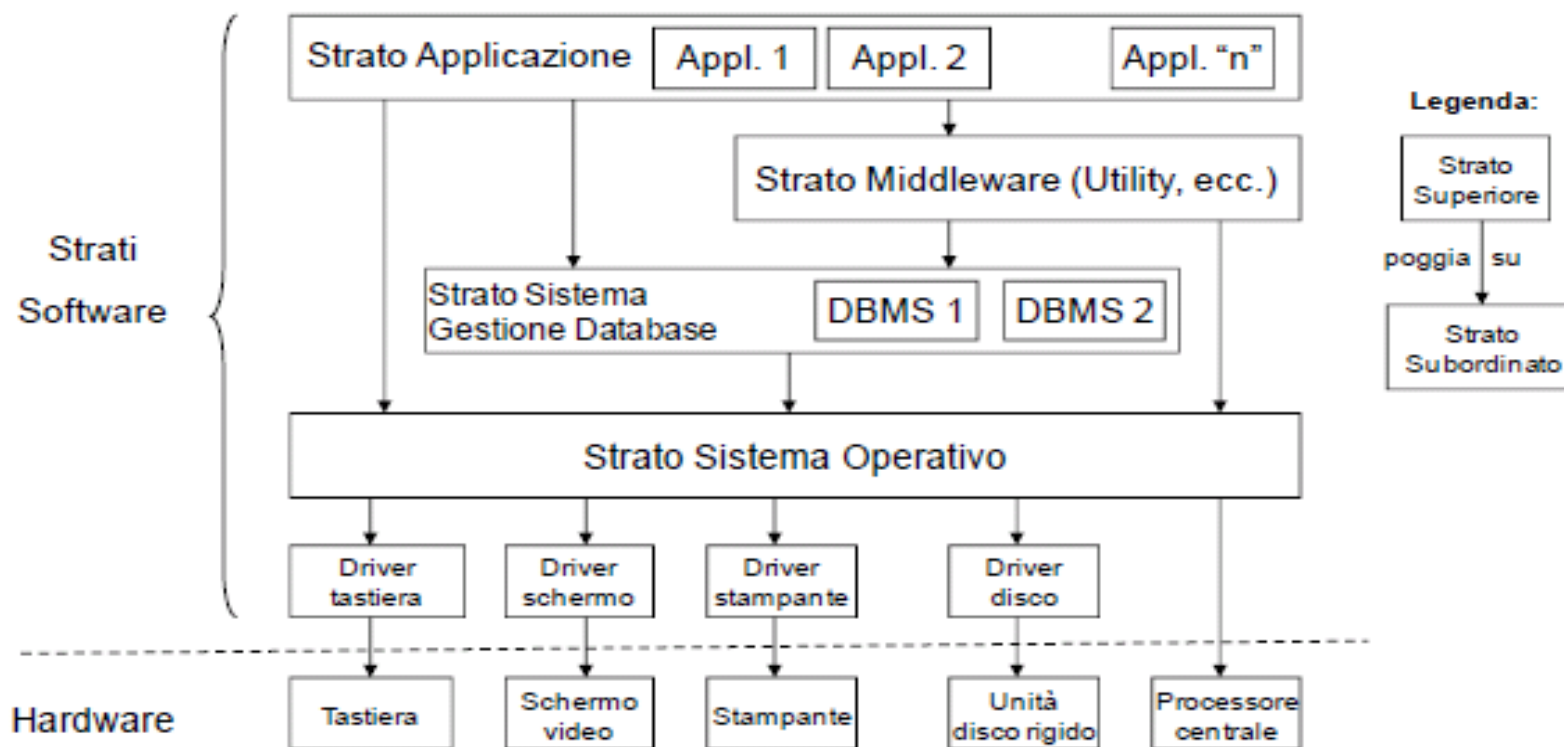


Figura 2.2.3.1 – Tipica architettura software a strati per un sistema informatico aziendale/MIS.

# Esempio

## SISTEMA INFORMATIVO PER LE CERTIFICAZIONI ENERGETICHE DEGLI EDIFICI (SICEE)

sistemapiemonte.it

utente: DEMO 30 CSI PIEMONTE [esci >](#)

[home](#) [indice servizi](#) [informazioni](#) [cerca](#) [posta](#) [profilo personale](#)

sei in: [sistema piemonte](#) > [ambiente](#) > [SICEE Sistema Informativo Certificazione Energetica Edifici](#)



### SICEE Sistema Informativo Certificazione Energetica Edifici

[home del servizio](#)

[dati personali](#)

> [visualizzazione](#)

[gestione a.c.e](#)

[prenotazione e acquisto  
codici a.c.e](#)

[compilazione e  
trasmissione a.c.e](#)

[documenti utili](#)

[contatti](#)

Utente: DEMO 30 CSI PIEMONTE | L'utente non è ancora registrato all'albo certificatori

[Gestione dati personali](#)

Registrazione o visualizzazione dati del certificatore

1. Dati anagrafici | **2. Residenza e contatti** | 3. Studio professionale | 4. Ordine e competenze | 5. Riepilogo |

I campi contrassegnati con asterisco (\*) sono obbligatori

Regione \*

Provincia \*

Comune \*

Indirizzo \*

Numero civico

CAP \*

Telefono \*

Cell.

E-Mail \*

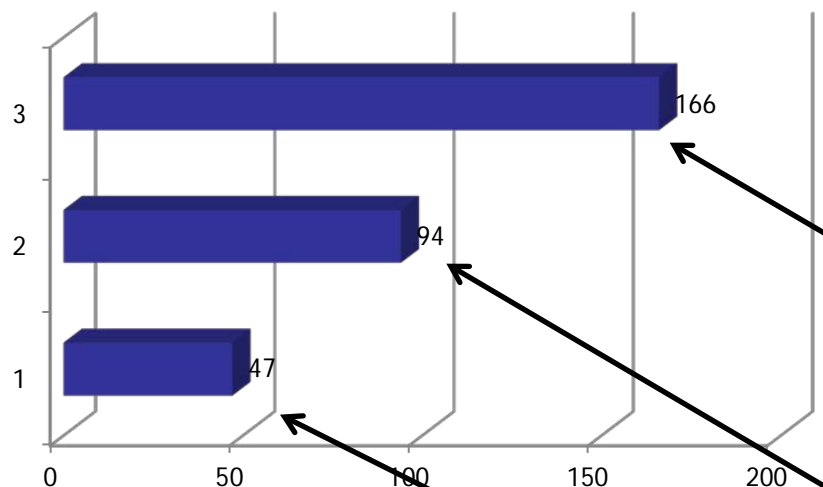
Fax



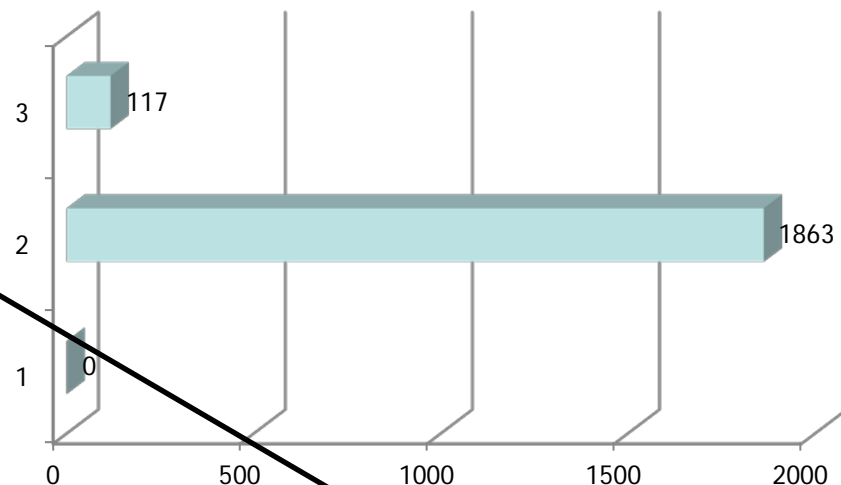
# Esempio

## SISTEMA INFORMATIVO PER LE CERTIFICAZIONI ENERGETICHE DEGLI EDIFICI (SICEE)

Dimensione Funzionale (FP)  
per Partizione



Dimensione Non Funzionale  
(SP) per Partizione



Funzionalità dati

Funzionalità di calcolo e  
computazionali

Funzionalità di data entry  
semplici

# Esempio

## Sviluppo applicazione SICEE

Prodotto SICEE, Numero di function point totale: 307  
Numero Function Point parte dati: 47 IFPUG FP  
Numero di Function Point Partizione 1: 94 IFPUG FP  
Numero di Snap Point Partizione 1: 1863  
Numero di Function Point Partizione 2: 166 IFPUG FP  
Numero di Snap Point Partizione 2: 117  
Ipotesi Effort di sviluppo previsto totale : 433 gg

# Esempio

## MEV applicazione SICEE

Numero Function Point parte dati change: 8,8 IFPUG FP  
Numero di Function Point Partizione 1 change : 9,5 IFPUG FP  
Numero di Snap Point Partizione 1 change : 98  
Numero di Function Point Partizione 2 change: 12 IFPUG FP  
Numero di Snap Point Partizione 2: 0  
Numero di Function Point Partizione 3 Add: 20 IFPUG FP  
Numero di Snap Point Partizione 3 Add: 16  
Numero di Function Point non toccati : 277

Prodotto SICEE, Numero di function point totale: 50  
Ipotesi Effort di MEV previsto totale : 70 gg

# Conclusioni

- Definizione del volume di software da sviluppare in base ai FUR e NFR
- Ipotesi di effort
- Miglior governo del progetto sui vari lotti/partizioni
- Gestione dei Change Request (pesati in FP e SP) e del riuso
- Accettazione deliverable con soglie di difettosità (difetti/FP)



# Conclusioni



Misurare è un fatto di cultura e va estesa a tutti i livelli aziendali (programmatori, analisti, PM, manager, etc.)

Le metriche sono un aiuto; non debbono e possono sostituire tutte le tecniche tradizionali di stima (Delphi, Bottom up, etc.)

# GRAZIE !

Domande? Perplexità?



Gianfranco.Lanza@gufpi-isma.org