

# Le architetture SOA: governance & qualità

*Massimiliano Argiolu*

*Technis Scrl*

*massimiliano.argiolu@technis.info*



**TECHNIS**<sup>®</sup>

**McCabe**  
SOFTWARE

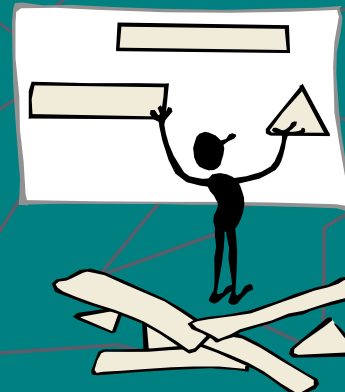
# Scenario

- ▶ Rivisitazione dei BP dei clienti correlati alla situazione economica attuale
- ▶ Utilizzo da parte dei clienti IT di KPI's (Key Performance Indicators) e cruscotti di Balanced Scorecards per misurare i BP (Business Process)
- ▶ Necessità di allineamento dell'IT ai BP dei clienti in termini competitivi rispetto a tempi e costi
- ▶ Necessità di impiantare dei processi IT "supportati" da indicatori oggettivi per la loro gestione in accordo con i BP aziendali e/o del cliente
- ▶ Necessità di incrementare la "qualità" del prodotto software per decrementare i costi dell'IT



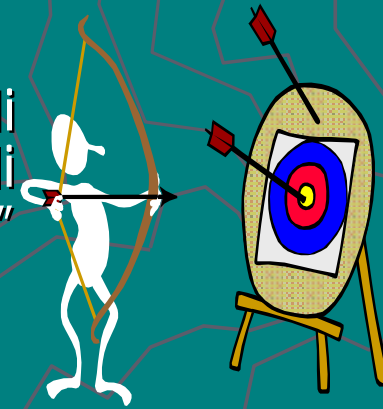
# La "rivoluzione" delle SOA

- ▶ IT modellato sui BP per partecipare al Business e non essere solo un "costo"
- ▶ Possibilità di "mapping" tra applicazioni IT (spesso web sited) e i BP
- ▶ L'owner del BP non deve conoscere l'infrastruttura IT
- ▶ L' IT supporta l'owner del BP nella definizione del Business diventandone "Partner" grazie allo stesso linguaggio (SOA)



# Obiettivi dell' IT

- Utilizzare modelli per stabilire relazioni tra IT e BP's per ottimizzare gli effort monitorando costantemente costi e attività;
- Utilizzare Software LifeCycles "adattivi" e indicatori di processo e prodotto software per raggiungere gli obiettivi funzionali, qualitativi e il "time-to-market" abbattendo i costi;
- Utilizzare prodotti per automatizzare la (ri)modellazione dei processi e per la rilevazione degli indicatori minimizzando l'impatto del cambiamento;
- Tali prodotti per automatizzare il Ciclo di Vita del Software abbattendo i costi e garantendo nel breve il ROI del loro acquisto, devono essere integrati "realmente" tra loro.



# Metriche di Processo...

I processi IT devono essere sempre più adattabili alle esigenze aziendali: per mantenerne il "controllo" è ***fondamentale***:

- *Monitorare ed ottimizzare i processi tramite KPI's garantendo il popolamento di un "dashboard" per il supporto decisionale;*
- *Utilizzare indicatori e soglie nei contratti di outsourcing, per garantire un "controllo" ottimale dell'esecuzione contrattuale;*
- *Utilizzare strumenti automatici per ottimizzare gli effort e minimizzare gli errori.*



# ...e KPI's

Alcune tipologie di KPI possibili nei cicli di vita del software :

- ▶ Qualità rispetto alle aspettative del cliente
- ▶ Qualità del prodotto software
- ▶ Produttività dei Team di sviluppo
- ▶ Stato generale dei progetti



Una volta definiti i KPI se ne deve effettuare il test p.e tramite una tecnica semplicissima detta SMART:

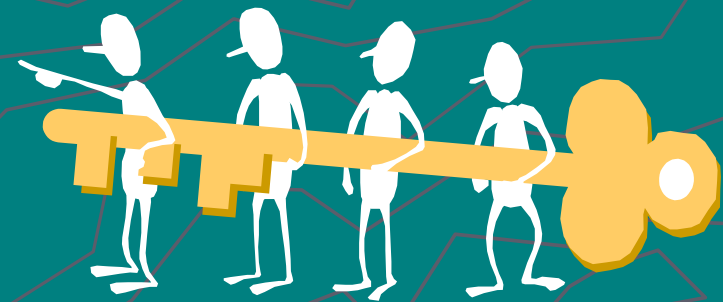
- S = Specifico:** chiaramente focalizzato per impedire errori di interpretazione
- M = Misurabile:** può essere quantificato e comparato con altri dati/indicatori
- A = Raggiungibile:** raggiungibile e credibile nell'ambito di utilizzo
- R = Realistico:** applicabile nella struttura organizzativa (è un costo stimabile)
- T = Opportuno:** realizzabile nel tempo a disposizione

# Un esempio pratico

<b>Obiettivo strategico : <i>Produrre software in architetture SOA incrementando i profitti</i></b>			
<i>Qualità rispetto alle aspettative del Cliente</i>	<i>Qualità del prodotto software dal punto di vista tecnologico</i>	<i>Produttività dei team di sviluppo</i>	<i>Stato generale del progetto</i>
<p><b>Obiettivo:</b> avere un prodotto affidabile e che possa essere adattato ai cambiamenti del suo business rapidamente e a costi contenuti utilizzando outsourcing di servizi parte dei BP aziendali</p>	<p><b>Obiettivo:</b> ottenere un prodotto molto affidabile di grande riuso adattabile a diverse tecnologie o architetture</p>	<p><b>Obiettivo:</b> utilizzare dei cicli di vita del software di tipo adaptive o Extreme Programming per minimizzare gli effort dovuti alle modifiche</p>	<p><b>Obiettivo:</b> ridurre al massimo i costi operativi e di gestione del rischio rispettando il Time to Market del Progetto</p>
<p><b>KPI Strategici</b></p> <ul style="list-style-type: none"> <li>•Disponibilità di interfacce “open” nei propri BP erogati tramite SOA</li> <li>•Bassi costi di esercizio</li> <li>•Costi minimali sulle variazioni dei BP</li> </ul>	<p><b>KPI Strategici</b></p> <ul style="list-style-type: none"> <li>•Progettazione OO con pattern riusabili (p.e. J2EE)</li> <li>•Progettazione accurata del Test</li> <li>•Validazione della progettazione del Test</li> </ul>	<p><b>KPI Strategici</b></p> <ul style="list-style-type: none"> <li>•rispetto dei Timebox assegnati</li> <li>•assorbimento delle metodiche di sviluppo adaptive o XP</li> </ul>	<p><b>KPI Strategici</b></p> <ul style="list-style-type: none"> <li>•rispetto dei costi preventivati</li> <li>•gestione dei rischi derivati dalla scarsa conoscenza dei BP del Cliente</li> <li>•rispetto dei tempi di consegna delle componenti</li> </ul>
<p><b>KPI Operativi</b></p> <ul style="list-style-type: none"> <li>•N° di servizi erogati tramite software</li> <li>•N° di servizi SOA utilizzati dal Sistema Informatico interno</li> <li>•Costo annuo dei servizi utilizzati</li> <li>•Costo dei singoli interventi al Sistema Informatico per modifiche ad uno o più BP</li> </ul>	<p><b>KPI Operativi</b></p> <ul style="list-style-type: none"> <li>•N° di componenti riusate da progetti precedenti</li> <li>•N° di componenti specifiche di interfaccia con altri sistemi sviluppate “ad hoc”</li> <li>•Copertura dei Casi di Test progettati rispetto ai requisiti</li> <li>•Copertura topologico strutturale del software</li> </ul>	<p><b>KPI Operativi</b></p> <ul style="list-style-type: none"> <li>•Scostamento in ore/uomo rispetto al Timebox assegnato</li> <li>•N° di persone skillate su nuovi metodi/linguaggi di programmazione</li> <li>•Ore/uomo spese per consulenti esterni</li> </ul>	<p><b>KPI Operativi</b></p> <ul style="list-style-type: none"> <li>•Scostamento dei costi tra preventivo e consuntivo e del del Time to Market</li> <li>•Costi sostenuti per il risk management</li> <li>•Costi sostenuti per acquisizione di consulenze esterne</li> <li>•N° di persone skillate su nuovi metodi/linguaggi di programmazione</li> </ul>

# Altri aspetti dell' IT Governance

- ▶ *BPM e BPR*
- ▶ *Change Management*
- ▶ *Quality Assurance*
- ▶ *Test e Performance Management*
- ▶ *SW Reengineering*
- ▶ *Delivery Management*



# Metriche di Prodotto: la suite McCabe

Per quanto concerne metriche di analisi statica è necessario definire quelle che sono adeguate alle tecnologie utilizzate: metriche tipiche di ambienti procedurali (p.e. McCabe Cyclomatic Complexity) non consentono la valutazione corretta del software Object Oriented (Java,.NET) rispetto alle caratteristiche indicate dalla norma ISO/IEC 9126:2001.

Diventa quindi necessario definire (anche contrattualmente) metriche e relative soglie in base all'ambiente e/o al linguaggio col quale è sviluppato il prodotto software, anche se si utilizzano metodologie "agili". Si consiglia l'uso delle metriche di Chidamber & Kemerer per la misurazione della qualità del software Object Oriented.



# Come concorrere al Business

- ▶ Incremento della qualità del prodotto software
- ▶ Certificazione del livello di Qualità del SW sviluppato o acquisito (ISO/IEC 9126:2001, ISO/IEC 25000)
- ▶ Definizione di processi di miglioramento della Qualità del SW (ISO/IEC 9004:2000)
- ▶ Riduzione dei costi indiretti di collaudo e manutenzione



# Mezzi...

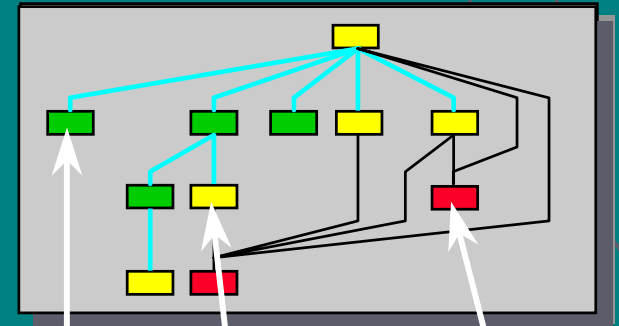
Monitoraggio della qualità SW tramite metriche statiche

Validazione del SW individuando:

- Codice inerte
- Test duplicati
- SW non testato
- Copertura topologica del Test

Uso di "set" di parametri di soglia differenziati in funzione del contesto

Inserimento di requisiti di qualità nelle specifiche tecniche di Outsourcing



Tested

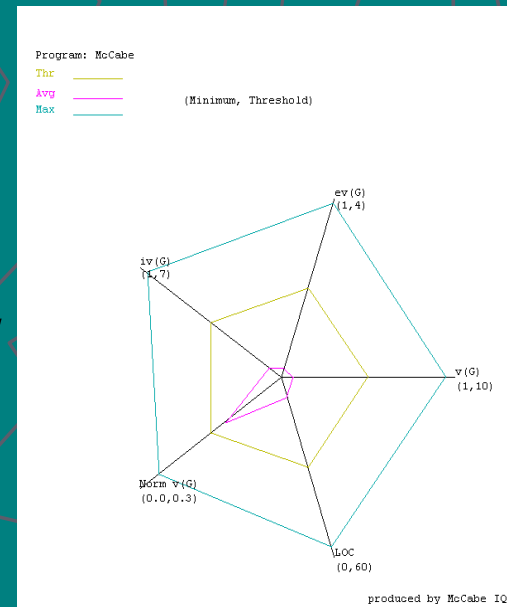
Untested

Partially tested

**McCabe consente il raggiungimento degli obiettivi di Qualità ad "effort zero"...**

# Metodologia

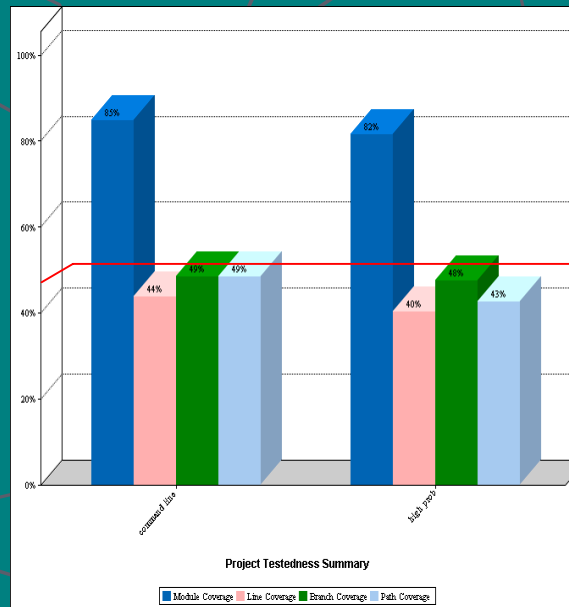
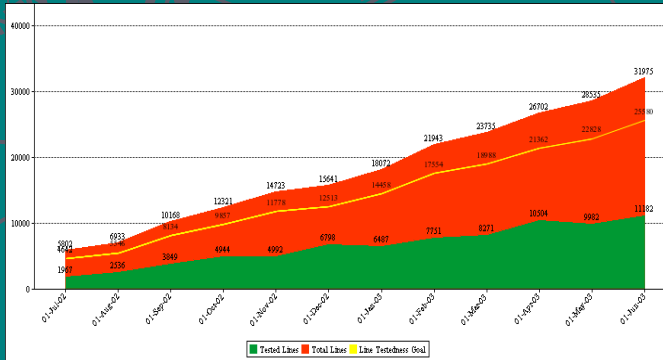
- Individuare parametri e soglie per la valutazione automatica delle metriche rilevate
- Sviluppare SW utilizzando metriche:
  - incrementa la qualità con "effort" molto bassi
  - garantisce il committente
  - migliora la manutenibilità
  - diminuisce i costi nell'ambito del Ciclo di Vita del SW
- Rilevare la copertura topologica del Test:
  - Valida il SW e il processo di Test
  - Diminuisce i "ticket" aperti ed il tempo di chiusura
  - Fornisce dati oggettivi sul rischio di rilascio
- Monitorare le metriche tramite un'apposito "cruscotto" profilato per i responsabili



# I valori di soglia

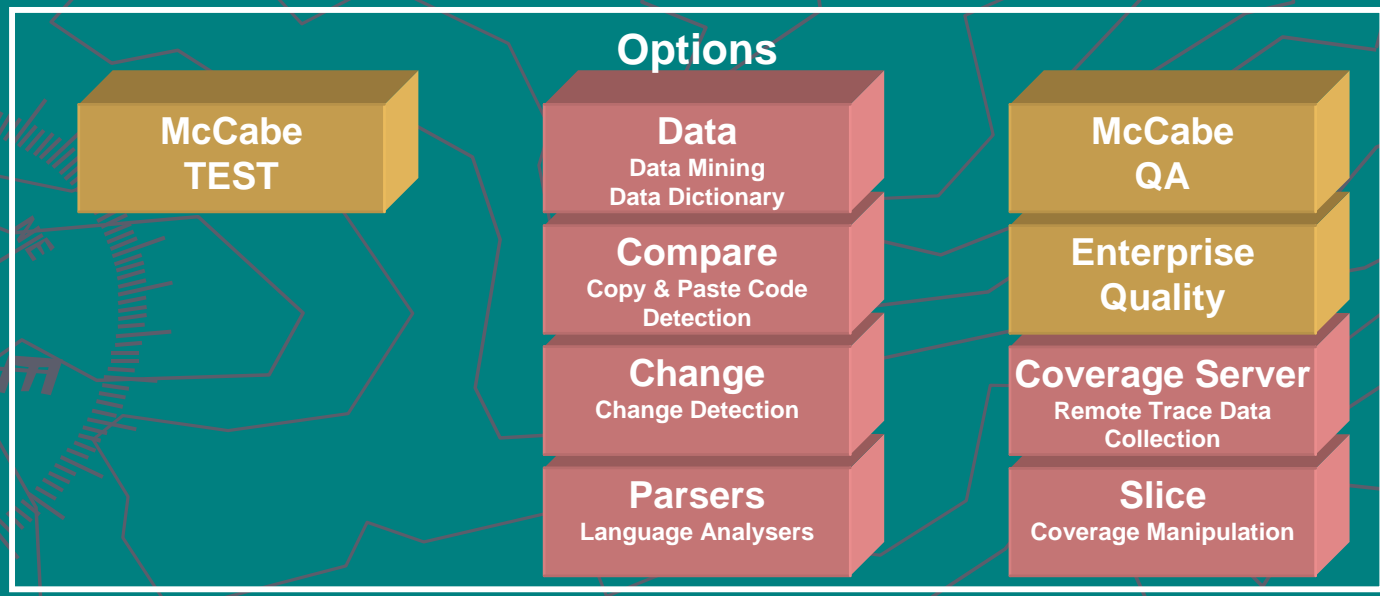
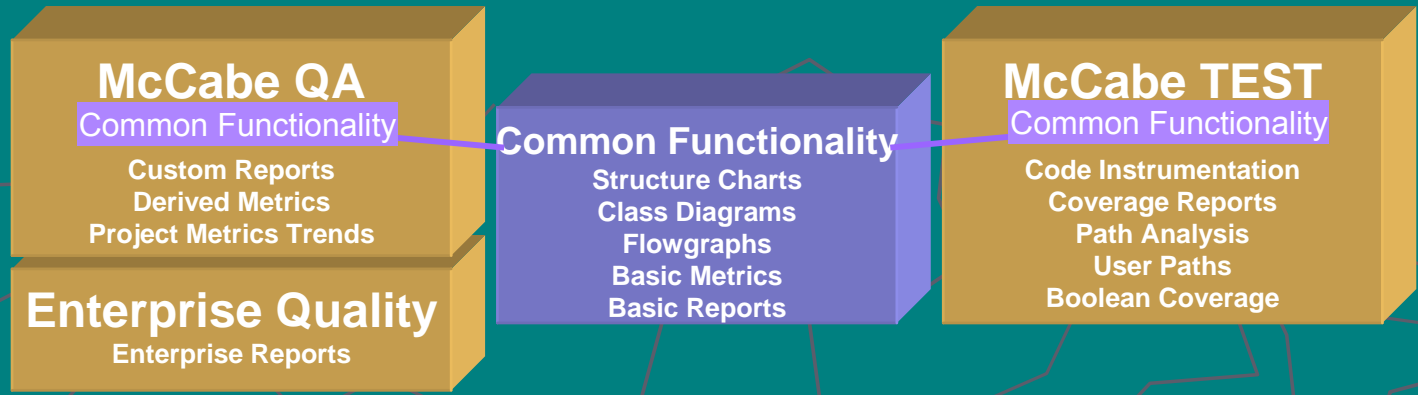
Utilizzo di differenti metriche e soglie in funzione di:

- Contesto contrattuale
- Tecnologie utilizzate (linguaggio, architettura, ecc.)
- Strategicità dell'applicazione



Size

# Configurazione del prodotto



# Ambienti e linguaggi

## Linguaggi:

- Java, JSP
- C, C.Net
- C++, C++.Net, C#
- VB, VB.Net
- Perl
- Fortran
- Ada
- COBOL
- Others

## Piattaforme:

- Windows
  - NT/2000/XP
- Unix
  - Solaris
  - AIX
  - HPUX

# Alcuni clienti europei

## Telecom

- Symbian
- Nokia
- British Telecom
- NCR
- Telefonica Spain
- Cogent
- Nortel
- Syntegra
- Marconi Comms
- NEC
- Orange
- Telecom Italia

## Finanziari

- Barclays
- Axa Sun Life
- Northern Rock
- Bank of America
- Royal Bank of Scotland
- Europay
- Mastercard
- Cap Gemini Ernst & Young
- DST
- Prudential
- Deutsche Boerse
- Sogei

## Industria

- Siemens
- Domino
- Ford
- Scomag
- Philips
- Delft Hydraulics
- Zytek Systems

## Software

- IBM
- Consort
- Systems Union
- 3Com
- ADC Metrica
- Finsiel
- Engineering

## Spazio e Difesa

- BAe
- DERA
- GCHQ
- Westinghouse Systems
- LVNL
- Meggitt Avionics
- Aermacchi - Logic
- Società Italiana Avionica

## Altro

- Philips Research
- Societe des Satellites
- Compaq
- Met. Office
- Dell
- ABB
- Consip

# Ritorno di investimento

I fattori chiave del ROI in questo approccio sono:

- ▶ Diminuzione dei costi di alcune attività del Ciclo di Vita del Software (p.e. Manutenzione Correttiva e Test) anche se si incrementano i costi di progetto qualora si debbano acquisire prodotti o skills peculiari
- ▶ Diminuzione dei rischi di rilascio del Software in esercizio e di altri fattori di rischio presenti in alcune fasi del Ciclo di Vita del Software
- ▶ Pianificazione puntuale dell'effort delle fasi del Ciclo di Vita del Software a valle del suo rilascio
- ▶ Incremento della qualità del prodotto offerto e quindi della "customer satisfaction" spendibile, ovviamente, a livello marketing



# Domande e Risposte

**GRAZIE**



*Massimiliano Argiolu*

*Technis Srl*

*massimiliano.argiolu@technis.info*



**TECHNIS®**

**McCabe**  
SOFTWARE