

# Software Intelligence Misurare per conoscere

Agenzia delle entrate-Riscossione è un Ente pubblico economico istituito ai sensi dell'articolo 1 del Decreto legge 22 ottobre 2016, n. 193, convertito con modificazioni dalla Legge 1 dicembre 2016 n. 225, e svolge le funzioni relative alla riscossione nazionale.

L'obiettivo dell'Agenzia delle entrate-Riscossione è di migliorare l'attività di riscossione nazionale mediante un approccio che garantisca economicità della gestione, soddisfazione dei contribuenti per i servizi prestati e aumento dei volumi di riscossione, anche mediante azioni di prevenzione e contrasto dell'evasione ed elusione fiscale.

Uno dei modi possibili per realizzare una riscossione efficiente, efficace ed economica è quella di sviluppare la componente informatica (*digital transformation*), in modo da:

1. rendere i sistemi di back end sempre più integrati ed efficienti;
2. ampliare i servizi ai Contribuenti offerti attraverso canali alternativi allo sportello (Sito web, App, ATM, ecc.) mediante una infrastruttura tecnologica ben integrata con i sistemi di back end.

Principale direttrice evolutiva del triennio è rappresentata dalla prosecuzione del programma d'iniziative, denominato Agenda Digitale AdeR, orientato a innovare i rapporti tra Pubblica Amministrazione, cittadini e imprese, attraverso azioni coordinate dirette a favorire lo sviluppo di domanda e offerta di servizi digitali innovativi ed a incentivare cittadini e imprese all'utilizzo dei servizi digitali

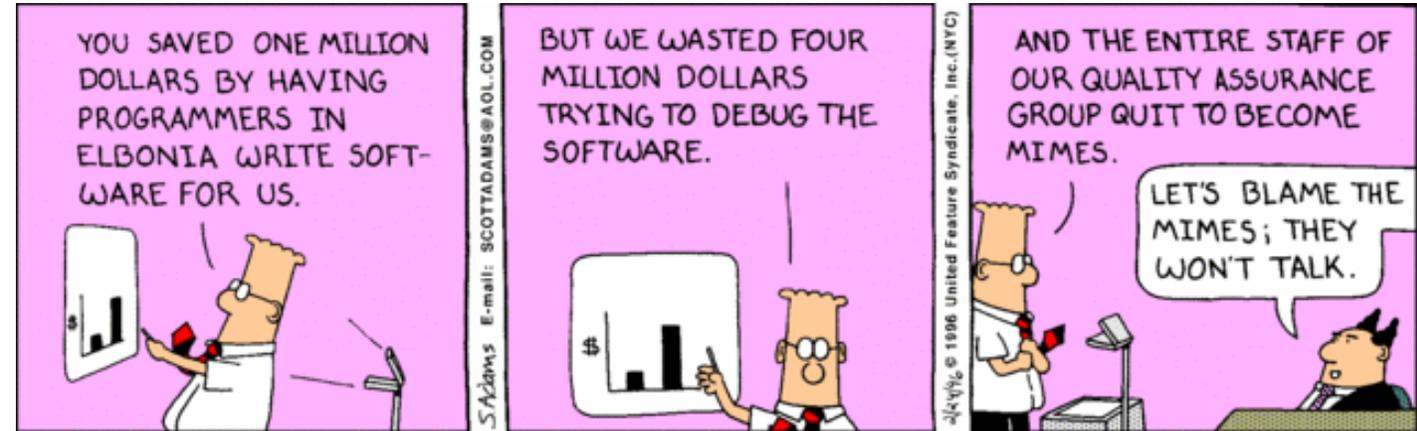
In linea con questo principio di base i progetti, piattaforme e soluzioni tecnologiche devono assicurare la crescita digitale, soddisfacendo le esigenze di:

- **considerare prioritario il principio di “digitale per definizione”** (digital first), progettando e implementando i servizi per il cittadino, a partire dall'utilizzo delle tecnologie digitali;
- **agevolare la modernizzazione dei processi della Pubblica Amministrazione**, superando la logica delle regole tecniche e della rigidità delle linee guida emesse per legge;
- **adottare un approccio architetturale basato sulla separazione dei livelli di back end e front end**, con logiche aperte e standard pubblici che garantiscano ad altri attori, pubblici e privati, accessibilità e massima interoperabilità di dati e servizi;
- **promuovere soluzioni volte a stimolare la riduzione dei costi e a migliorare la qualità dei servizi**, contemplando meccanismi di remunerazione che possano anche incentivare i fornitori ad individuare modalità di erogazione dei servizi sempre più innovative.
- **incrementare l'automazione dei processi di back office e l'interoperabilità con gli enti** appartenenti al medesimo ecosistema per minimizzare i tempi di risposta al cittadino ampliando i canali di contatto;
- **implementare una strategia multicanale** per migliorare la customer experience mediante la realizzazione di ulteriori servizi self service per i cittadini;

- I sistemi ci sono già e vanno «convinti» a fare anche altro
- Le risorse (hardware, software, economiche, ecc.) non sono mai infinite
- Difficile operare con solo risorse interne, occorre coinvolgere necessariamente fornitori o partner tecnologici



- Quanto conosco bene i sistemi su cui intendo intervenire?
- Quali risorse contrattuali ho a disposizione?
- Quale modello di outsourcing posso applicare?
- Come misurare i risultati?



1996, Scott Adams®



1997, Scott Adams®

## I costi

**€185**  
Tariffa giornaliera di uno sviluppatore outsourced

**75%**  
Degli sviluppatori dei fornitori hanno meno di 3 anni di esperienza specifica

**10x**  
La forchetta di produttività tra sviluppatori esperti e inesperti

?

Come possiamo trovare le giuste proporzioni tra **costo**, **agilità**, e **rischio** per le nostre organizzazioni in cui il software è critico per il business ?

Fonte: *The Economics of Software Quality*, Jones, Bonsignour.

## Il modello di outsourcing

**Progetto A a Roma**

- 20 sviluppatori
  - 60 k€ per FTE
  - 18 FP per mese persona
  - 360 FP consegnati
- costo 278 €/FP

**Progetto B a Bucarest**

- 20 sviluppatori
  - 45 k€ per FTE
  - 18 FP per mese persona
  - 360 FP consegnati
- costo 208 €/FP

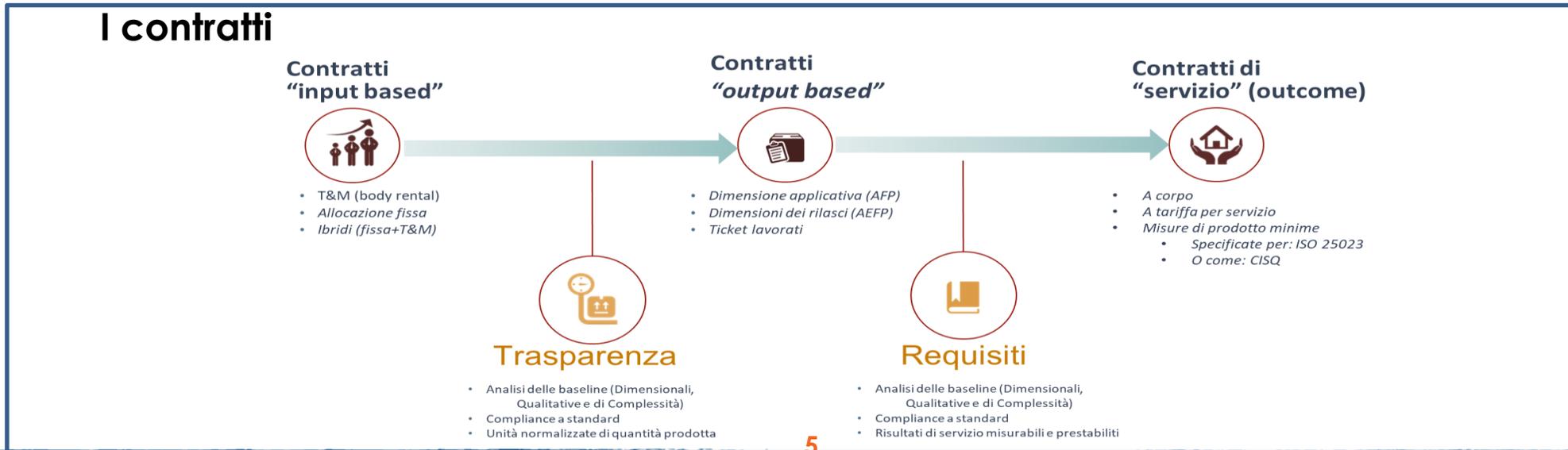
Progetto B sembra avere produttività del 33% superiore

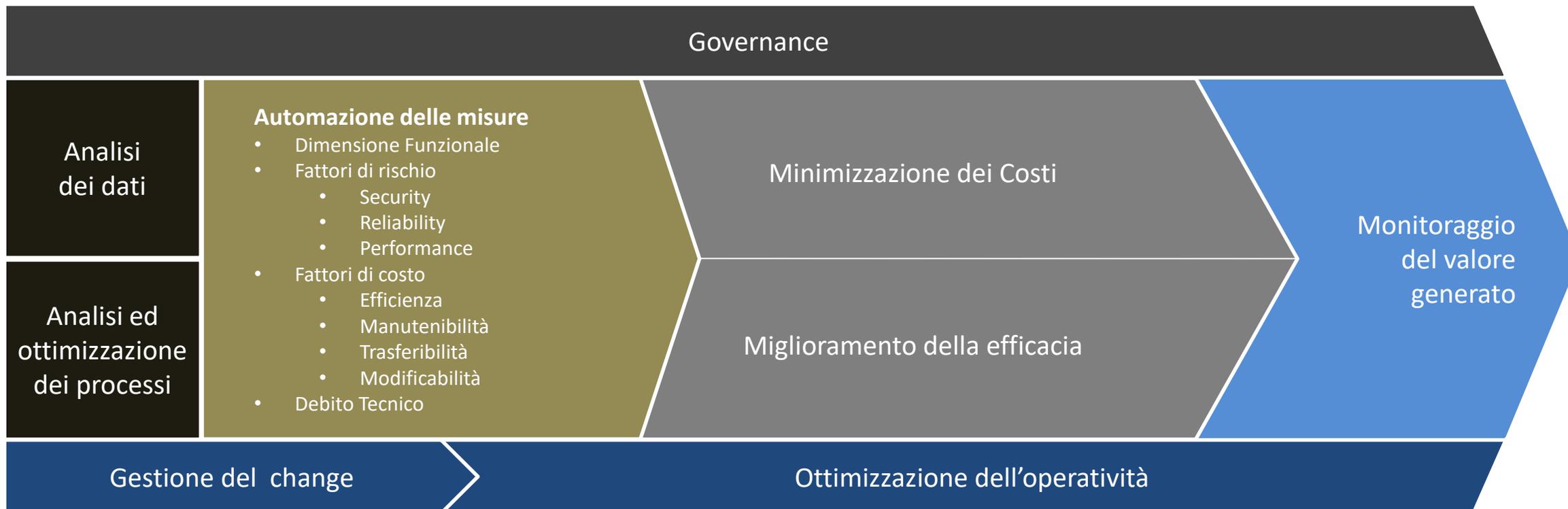
**Un attimo ! Abbiamo considerate il rischio incluso ?**

- 2 violazioni critiche per FP
- Costo for 720 interventi = 400 k€
- 370 €/FP di TCO

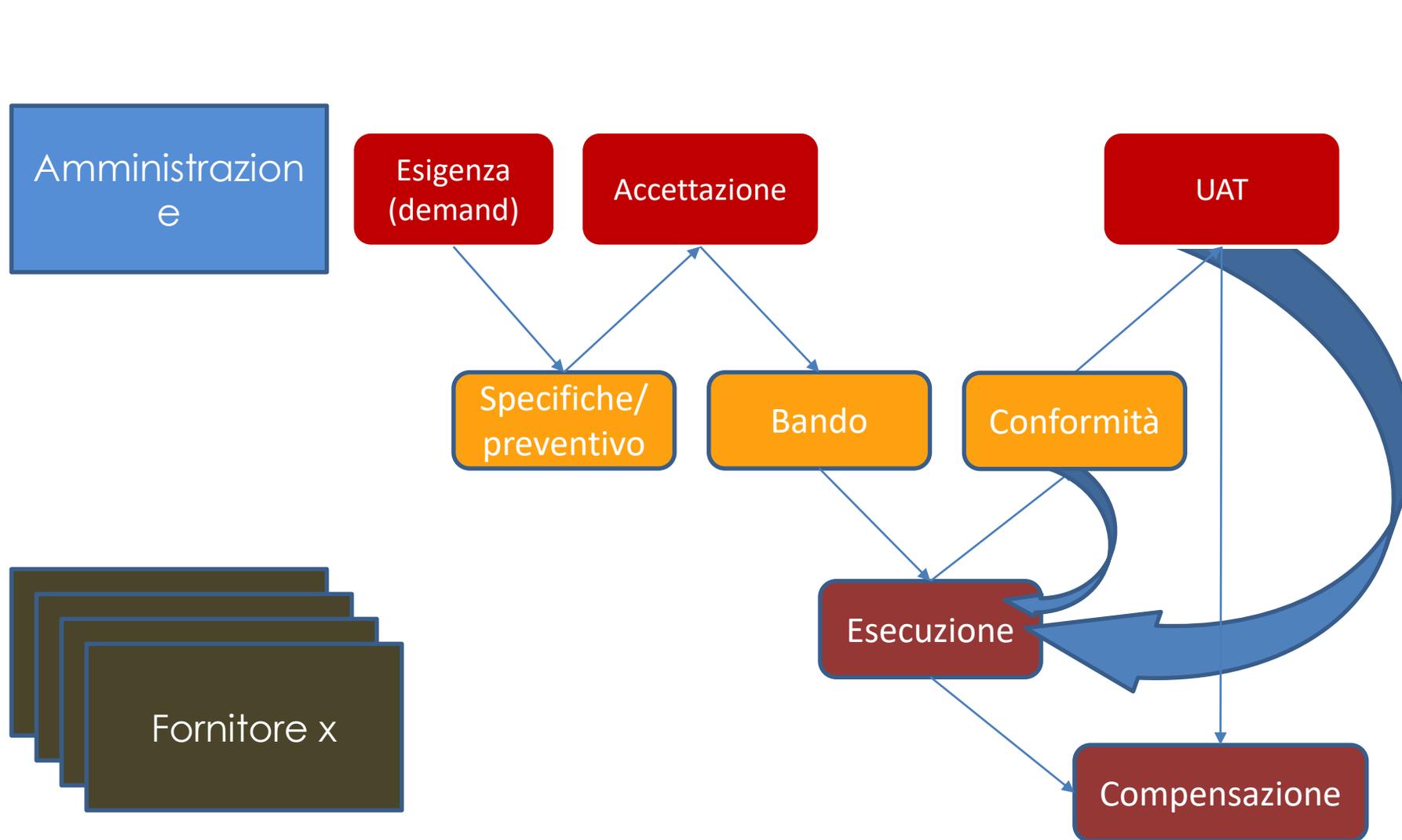
- 5 violazioni critiche per FP
- Costo per 1800 interventi = 750 k€ !!!
- 382 €/FP di TCO

Progetto A contiene meno rischio: Ha una produttività maggiore!!!





La **software intelligence** lavora sui diversi piani di gestione del ciclo di vita degli software asset, in modo da potere **condurre la governance**, supportati da **misurazioni oggettive** delle attività e dei rischi. A livello operativo, tramite le indicazioni sulla rimozione di criticità insite nelle applicazioni complesse, permette di migliorare i propri risultati a contenendo il fattore costo.



GAP

L'efficacia dei tests di accettazione viene diluita dalla presenza di carenze qualitative

Prevedere ricicli è oneroso a bando e comunque offre poca leva: la mancata adozione di standards quali-quantitativi rende ampio il margine di questionabilità. Inoltre l'intervento a termine del processo eleva il costo delle modifiche, rendendo il fornitore poco propenso a collaborare.

La remunerazione non è sempre strettamente correlata alla qualità e quantità effettiva della esecuzione

## 1. Identificazione di risultati che sono critici per il business / l'utenza / per i processi



## 2. Riconduzione ai KPI o indicatori di rischio



## 3. Misurazione per stabilire una baseline e fare benchmark interno o di settore

## 4. Monitoraggio nel tempo : comunque prima del passaggio in produzione

## Stima a priori

Prima dell'esecuzione del progetto



*Quanto costerà questo progetto?  
Quanti sviluppatori saranno  
necessari?*

## Consuntivo ex post

Misurazione del risultato dello sviluppo



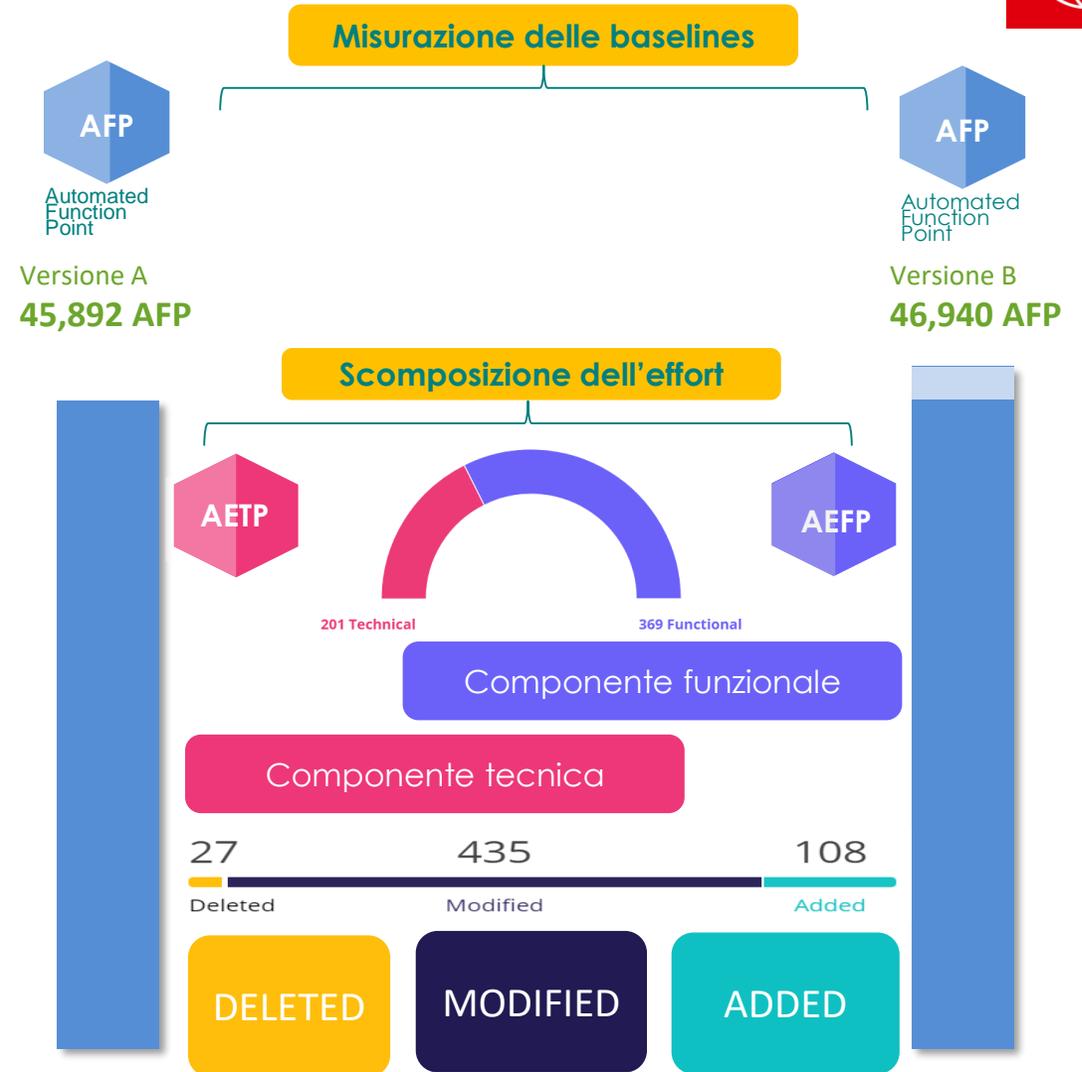
*Stiamo pagando il giusto?  
Il software è a «regola d'arte»?*

La stima a preventivo permette l'inizio dei lavori, la misurazione a consuntivo ne assicura l'adeguatezza

- **Misurazione delle baseline**, resa possibile dalla misurazione in punti funzione automatici (AFP): standard ISO 19515:2019 e standard OMG.
- **Misurazione delle attività**, utilizzando l'approccio IFPUG (ovvero il concetto di «added», «changed», «deleted») e quello OMG (AEP).



Usare unità di misura standard per monitorare la dimensione funzionale e non funzionale delle attività di sviluppo e manutenzione





Numero di PF **Aggiunti o Modificati**

- 1) Misura non legata ad una specifica tecnologia della dimensione e valore del "consegnato"
- 2) Confronto oggettivo tra applicazioni diverse, team,

- fornitori, release e metodologie di sviluppo.
- 3) Monitoraggio temporale per controllare i costi e i rilasci nel tempo

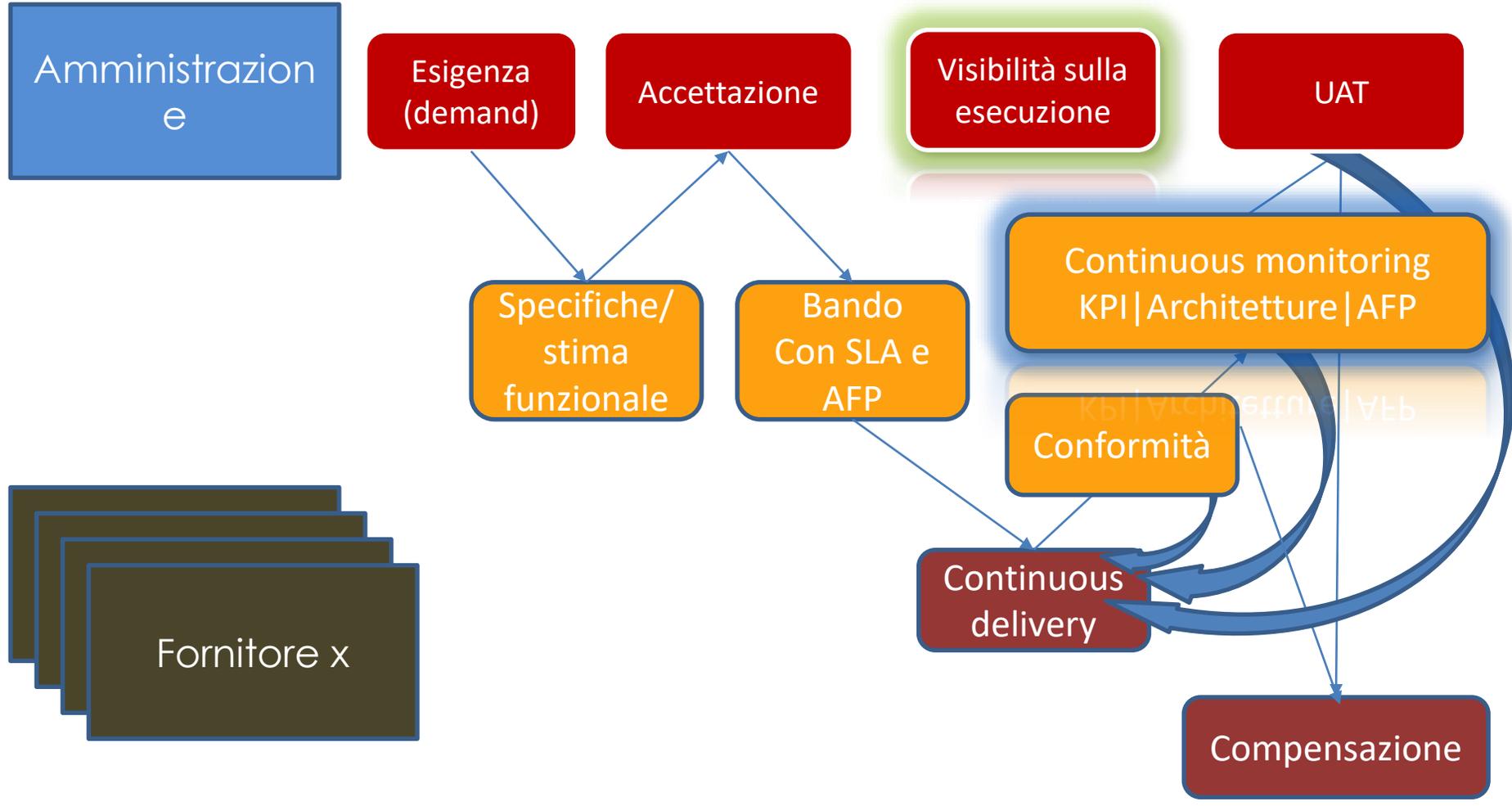
Numero di **Violazioni Critiche** aggiunte

Numero di **Punti Funzione Aggiunti o Modificati**

- 1) Valutazione complessiva dell'ultima release
- 2) Confronto oggettivo tra release
- 3) Trend temporali per visualizzare i progressi
- 4) Può essere usato come release gate



**⚠ Violazioni Critiche:** sono il livello più grave di non-compliance con le migliori pratiche software.



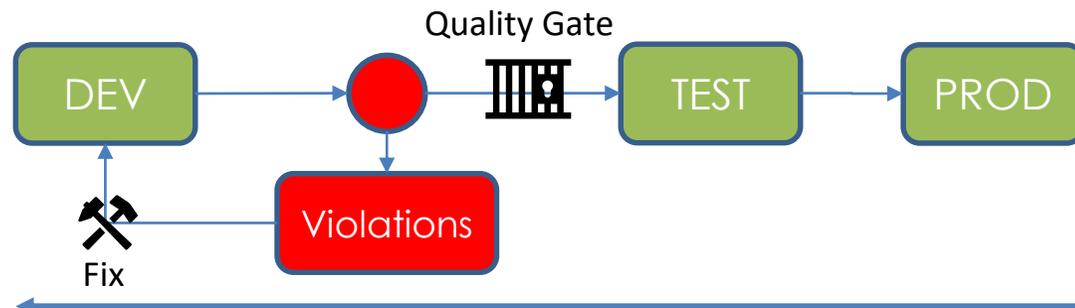
I tests di accettazione sono focalizzati sugli aspetti funzionali, i rischi non funzionali sono già mitigati

Molte problematiche vengono intercettate dalla software intelligence grazie agli SLA sui KPI per i rischi. La visibilità sull'esecuzione permette alle attività di governance (tecniche e di business) di rimanere allineate con l'esecuzione.

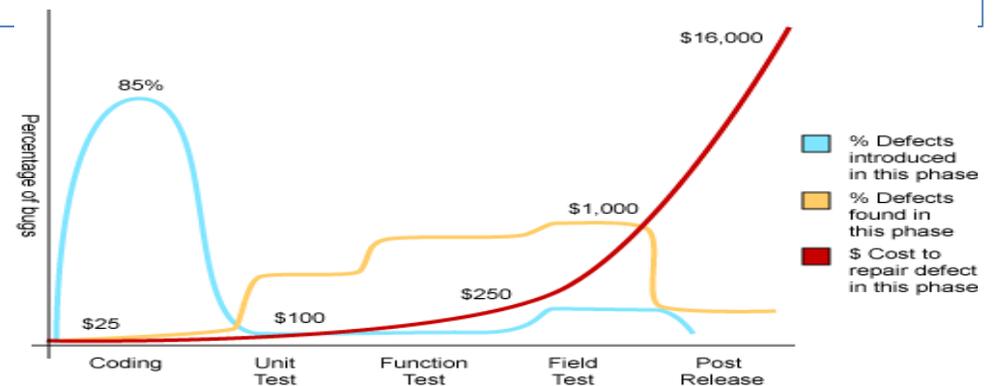
La compensazione avviene al netto di importi sospesi per rimedi in corso o carenze quantitative

Con l'utilizzo ottimale di questo approccio nella toolchain di automazione si avvia il processo di mitigazione dei rischi connesso alla produzione di software. **Questo include (in progressione):**

- **Misurazione** degli aspetti di rischio del software per mitigarli prima che siano troppo vicini alla produzione  
→ Automazione della **identificazione anomalie**, priorità e notifica agli sviluppatori senza condizionare tempistica dei rilasci
- **Correlazione misurazioni & anomalie** riscontrate lungo gli step dell'SDLC (test falliti, commit ripetute..etc.)  
→ Impostazione di un **quality gate** per impedire che le anomalie passino agli stadi successivi con i conseguenti maggiori costi di rimedio o addirittura i costi di ripristino a causa incidents causati
- **Revisione e affinamento delle caratteristiche del/i gate** (hard/soft) in base al processo (agile, wf)  
→ Quantificazione del **valore delle attività di rimozione** delle anomalie comparandole a dati storici, confrontandole con applicazioni in cui non vengono effettuate attività preventive.



Spostamento della identificazione e rimedio a «sinistra»

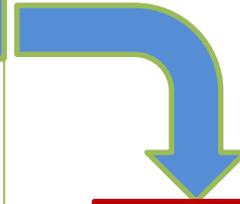


Source: Applied Software Measurement, Capers Jones, 1996



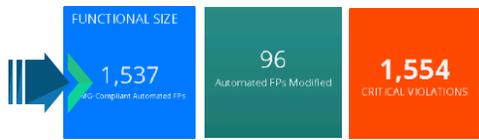
## Osservazioni fattuali

- I fornitori **non sono committati** nel rispettare gli obiettivi di qualità che riguardano il codice consegnato
- **Non sono chiari gli incentivi o i disincentivi (penali)** per i fornitori per cui devono rimediare ad anomalie nel codice entro dei tempi certi
- I **budget** per gli sviluppi SW **non sono adeguati al dimensionamento** AFP/EFP
- La formalizzazione dei contratti **non permette di valorizzare correttamente la produttività** dei fornitori e di confrontarla con un riferimento oggettivo standard normalizzato.

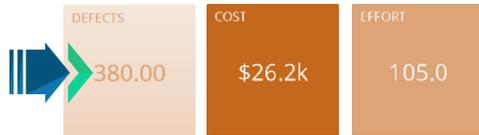


## Direzione strategica

- **introdurre una misura di produttività** condivisa ed accettata dai fornitori, basata su standard ISO AFP/EFP, rendendola parte delle condizioni contrattuali
- I fornitori dovranno **permettere accesso o fornire trasparenza secondo KPI** standard al processo di sviluppo, in modo da potere monitorare ed rilevare qualità e rischi anticipatamente ai rilasci
- Una **revisione degli schemi contrattuali** introducendo la valutazione dei fornitori in base alle misure oggettive di qualità, rischio ed efficacia del prodotto fornito



Metriche del software



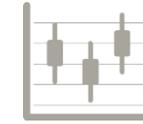
Dati di progetto e costi

10001010  
010101  
0101

## Dati integrati



## Benchmark di baseline

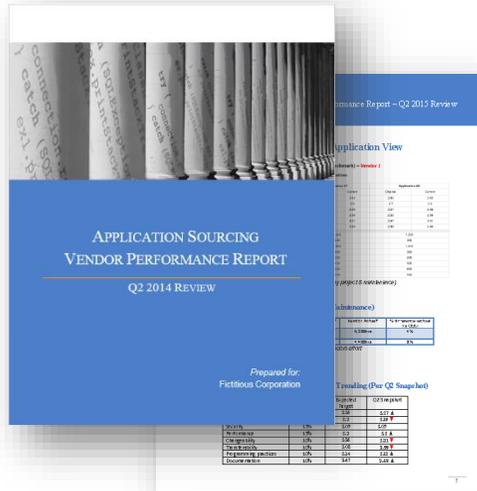


## Benchmark Interno



Grazia anche al supporto di CAST, AdeR ha avviato un programma di adozione delle metriche e degli standards, basati su una progressiva “educazione” dell’ecosistema dei fornitori e degli stakeholders:

**1** **Trasparenza**                      **2** **Indicazioni**                      **3** **SLA**                      **4** **Accettazione**



**4.6 CISQ-OMG**  
 Il CISQ è un organismo fondato dall'OMG. Il suo scopo è sviluppare, in collaborazione con le maggiori organizzazioni ICT mondiali, standard internazionali per la misurazione automatica, a partire dai sorgenti, della dimensione funzionale e non-funzionale del software, nonché dei fattori di rischio derivanti dalla struttura degli applicativi.  
 Le caratteristiche non funzionali di un software che CISQ propone di misurare sono<sup>23</sup>:  
 - reliability;  
 - performance efficiency;  
 - security;  
 - maintainability<sup>24</sup>.  
 In estrema sintesi, CISQ-OMG propone metriche e metodologie di misura basate sulla verifica del rispetto di alcune regole (buone pratiche architettonici, di codifica e disegno), adatte a essere rilevate tramite analisi statica, definita anche analisi strutturale, del software applicativo in tutte le sue componenti (codice sorgente, database, ecc.). In termini ISO 25010, si riferiscono alla qualità interna del software in esame.  
 Le metriche proposte da CISQ-OMG sono coerenti con la norma ISO 25010 e costituiscono una estensione della ISO 25023, secondo quanto previsto dal paragrafo 2 della norma stessa.

Nome	Descrizione	Tipo	Periodo	Baseline	Peso	Min	Max	Bonus migliorativo
Security	La plausibilità che l'applicazione si presti ad essere soggetta ad incidenti di sicurezza	Componente	Mensile	0.02	35%	0.018	0.022	5%
		Strutturale	Mensile	0.02	35%	0.00	0.019	5%
Reliability	Il rischio di incidente applicativo risultante da operazioni di manutenzione.	Componente	Mensile	0.1	25%	0.09	0.11	5%
		Strutturale	Mensile	0.1	35%	0.07	0.10	5%
Performance Efficiency	Quanto velocemente il sistema possa assorbire eventi non attesi e ristabilire le performances	Componente	Mensile	1	25%	0.9	1.1	2%
		Strutturale	Mensile	1	25%	0.8	1.0	2%
Maintainability	La facilità con cui un'applicazione possa essere evoluta e/o manutenuta	Componente	Mensile	3	25%	2.7	3.3	5%
		Strutturale	Mensile	3	25%	2.3	3.9	5%
Totale					230%			

Acceptance Criteria

Client will indicate its acceptance of each Deliverable in writing and forward such written acceptance to Vendor Company. Once the Deliverable is prepared for acceptance to Client, Client shall have ten (10) business days after then to either (i) accept the Deliverable in writing or (ii) reject the Deliverable by notifying Vendor Company in writing of Client's reasons why the Deliverable is not acceptable.

In the event Client rejects a Deliverable, Vendor Company will use commercially reasonable (meaning multiple, as required) earnest, conscientious attempts and/or services to resolve the issues presented, who demonstrate good judgment intended to complete, acceptably, solve or provide Client with successful results, conclusions, and/or problem and/or complete deliverables related to address Client's reasons for such rejection and resolution. The Deliverable to Client for acceptance at no additional charge. Notwithstanding the foregoing, a Deliverable shall be deemed accepted upon the occurrence of Client failing to accept or reject the Deliverable within the 10 business days of receipt. If, after a commercially reasonable number of attempts to modify the non-conforming Deliverable by Vendor Company, Client still, by written notice to Vendor Company, rejects the Deliverable, Client may terminate this SOW pursuant to Section 2.2.1 (Termination for Cause) of the Agreement, effective upon notice to Vendor Company. Except as set forth in Section 2.2 (Termination) of the Agreement, upon termination as set forth in this section and subject to payment by Client of all fees for services rendered heretofore through the effective date of termination, Client and Vendor Company shall have no further obligations to Vendor Company pursuant to this SOW.

Client will evaluate the Deliverable quality using CAST Application Intelligence Platform (AIP) against the following measures:

Application Quality Measure	Measurement Criteria	Minimum Required Score (based on industry benchmark - top 25 percentile)	Critical Rules (As described in appendix)
Programming practices	Delivered codebase will follow the standard coding guidelines as prescribed by CAST AIP guidelines and a minimum score	100	Zero Violations - Please refer to the appendix for the list of critical programming practice rules.
Architecture and Design	Delivered codebase will adhere to the design and architecture approved by Client in the design phase of the program and as directed by CAST AIP guidelines. Best Practices Deviation from design will be acceptable, however a report approach is evolved during the coding phase and monthly report given with the Client.	100	Zero Violations - Please refer to the appendix for the list of critical architecture rules.
Documentation	Delivered codebase will have source documentation as per the standard coding guidelines and as directed by CAST AIP documentation guidelines. Delivered codebase will not	100	Zero Violations - Please

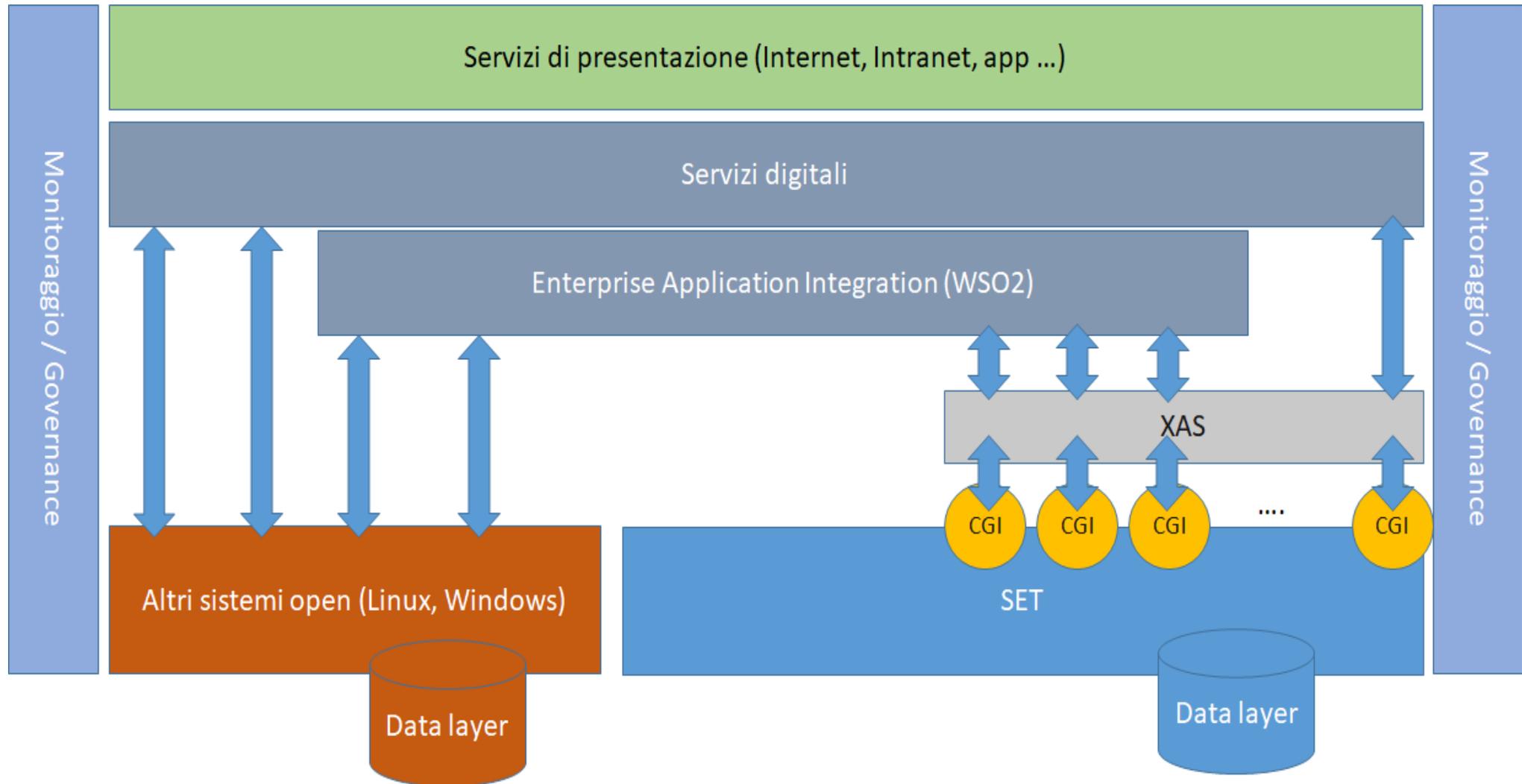
**Percorso di maturazione dell’ecosistema di forniture ADM**

Publicare KPI di misurazione presso gli stakeholders di rilievo (interni e dei fornitori) genera *per se* ritorni positivi

Richiedere immediatamente ai fornitori delle dichiarazioni riguardo alle misure secondo determinati standards (CISQ, ISO 25023) setta il tono della relazione.

Considerare le attività di sviluppo e manutentive come servizi dove i cui livelli di accettabilità sono riscontrabili dalla misurazione del prodotto e non (solo) del processo.

Ogni nuovo sviluppo dovrà essere misurato ed accettato (o rifiutato) sulla base di criteri oggettivi di misurazione.



Sono stati calcolati un totale di:

**77.610 AFP**



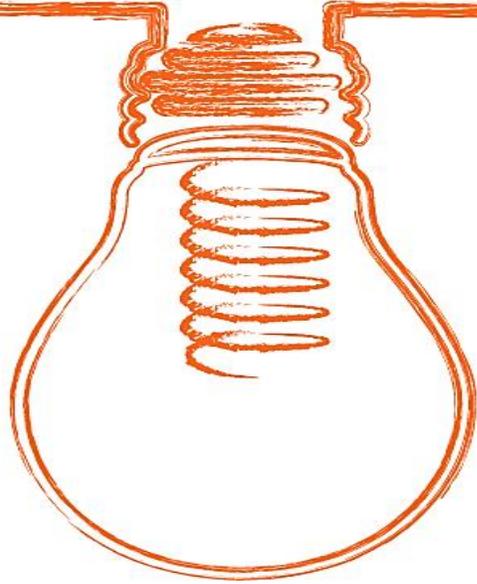
Ricostruita la baseline di SET e definite le «health board» dei singoli macro moduli, sono emersi parametri su cui è necessario intervenire soprattutto se si vuole continuare a offrire servizi digitali sempre più evoluti.

A questo scopo si sta operando su due linee:

- continuare a portare avanti gli sviluppi, soprattutto quelli imposti dalla normativa vigente,
- ricondurre, laddove necessario, i moduli software a valori di salute accettabili.

L'adozione di una metodologia e di strumenti atti alla misura del software ha un ruolo chiave per il successo della digital transformation, che si traduce nelle seguenti azioni:

1. mantenimento della baseline e relativo aggiornamento ad ogni rilascio software
2. Implementazione di una tool chain che preveda per ogni rilascio:
  - a. misurazione degli aspetti di rischio del software per mitigarli prima che siano troppo vicini alla produzione e gestione dei «quality gate»
  - b. conciliazione tra stima, consuntivo e rispetto dei KPI e delle metriche relative
3. estensione di questo approccio a tutti gli applicativi di AdeR



***Backup***

## Livelli dei KPI raggiunti rispetto agli obiettivi



Allarmistica legata a SLA di rilascio applicativi



Dimensione  
Funzionale

Robustness

