



MAPS
SHARING KNOWLEDGE

I QUADERNI DI **6MEMES**

I Servizi IT al lavoro: dal concetto di Prodotto a quello di Servizio, passando in primo luogo dalle “persone”.

di Giulio Destri



Introduzione

Abbiamo visto insieme nel precedente [White Paper](#) come l'innovazione tecnologica e il trattamento (sicuro) dei dati siano divenuti ogni giorno di più concetti e funzioni strategiche, sia per le aziende che per le singole persone.

Partendo dal ruolo dell'ICT nella società di oggi – passando attraverso le sue figure professionali – abbiamo approfondito insieme i rischi della materia in termini di privacy e sicurezza dei dati per approdare a un'analisi approfondita sulla messa in atto del GDPR e il suo impatto su servizi esistenti e sulla progettazione di nuovi servizi IT.

In questo nuovo quaderno il focus è invece sulla trasformazione in atto dal concetto di “prodotto” a quello di “servizio”, cambiamento reso possibile dalle attuali tecnologie che veicolano ogni “merce” e più in generale ogni informazione di valore attraverso veri propri flussi di dati.

Il discorso è poi proseguito sull'economia dei servizi – per parlare della sua integrazione con l'IoT (Internet delle Cose), con un caso applicato riguardante la cura della persona – approdando infine alle figure più strategiche dell'IT che rendono possibili le trasformazioni in atto, quali il Business Analyst, il Developer e il Sistemista.

In itinere, è stato affrontato anche il tema, importantissimo e troppo spesso sottovalutato, della comunicazione interpersonale in questo ambito.

Perché le tecnologie sono sicuramente uno dei pilastri del futuro, ma vanno gestite, organizzate, costruite al meglio, e ogni innovazione va accompagnata non solo dalle giuste tecnologie, ma anche (e soprattutto) dalla giusta organizzazione

e attraverso persone qualificate, motivate e competenti.

Il che porta all'ultimo capitolo del nostro White Paper, dedicato ai costi dell'innovazione: da quello della creazione di nuovi sistemi e processi a quelli di "esercizio", gestione e manutenzione. Buona lettura.

Giulio Destri

INDICE

- Introduzione

- 01** - L'economia che si rinnova con la società e la tecnologia: dal concetto di Prodotto a quello di Servizio.
- 02** - Servizi e IoT: società dei servizi e Internet delle Cose.
- 03** - Business Analyst: sinergia tra IT e Business per guidare l'azienda in un mondo che cambia. Donna è meglio?
- 04** - Comunicazione interpersonale nei settori tecnici e specialistici.
- 05** - Il compito fondamentale in ICT: lo sviluppo del codice sorgente.
- 06** - A me le "reti": la figura del buon amministratore (di sistema)!
- 07** - Sviluppo versus funzionamento operativo di servizi IT: coordinare esigenze in apparenza contrapposte.
- 08** - Total cost of ownership: spese di costruzione e di esercizio di un sistema IT.

- Conclusioni

- Approfondimenti

01 L'economia che si rinnova con la società e la tecnologia: dal concetto di Prodotto a quello di Servizio

Iniziamo il nostro percorso ITC con un concetto, quello di servizio IT: una unità funzionale, composta di software, hardware e reti, che eroga un valore per i suoi utenti e, che spesso nasce come soluzione ad una specifica esigenza e poi si evolve nel tempo.

Nel nostro "discorso" verrà inoltre ampliato e generalizzato il **concetto di servizio**, inteso come *"il risultato di attività svolte all'interfaccia tra fornitore e cliente e di attività proprie del fornitore, per soddisfare le esigenze del cliente"*.

Un servizio, infatti, **produce valore per il cliente**, a fronte di un pagamento (ad esempio, di un canone periodico), **senza che il cliente si debba preoccupare di dettagli** realizzativi, di costi diretti e, almeno in linea teorica, dei rischi.

Esempi di servizi sono:

- ✓ *In ambito ICT, la connessione ad Internet.*
- ✓ *In ambito trasporti, il treno o l'autobus.*
- ✓ *La fornitura di acqua, elettricità, gas.*
- ✓ *La raccolta delle immondizie.*
- ✓ *La consegna a domicilio di prodotti.*



In molti manuali di economia il servizio viene definito come *“l'equivalente immateriale di una merce”* e quindi un analogo immateriale del prodotto, distinguendo quindi da un lato prodotti, tangibili, e dall'altro servizi, che procurano valore ma intangibili.

Ma siamo sicuri che questa differenza sia ancora così netta?

Per approfondire la questione, prendiamo ad esempio qualcosa che *“dovrebbe”* essere un prodotto: l'automobile.

Un'automobile è sicuramente un oggetto materiale, ha caratteristiche fisiche ben precise, un valore di mercato iniziale e che poi tende a decrescere nel tempo e con l'uso, come tanti altri *prodotti*. La sua scelta viene fatta per un insieme di motivi, sia razionali sia emotivi.

Ma, **dal punto di vista del proprietario**, al di là dei rari casi (es. automobili di lusso o storiche) in cui può essere un investimento, **cosa è l'automobile? Essenzialmente è lo strumento principale attraverso cui si viaggia, ci si muove**, per lavoro, per andare al lavoro, per diletto, per necessità.

In molte parti del nostro paese disporre di un'automobile è una necessità, anche solo per poter svolgere le normali attività come recarsi al lavoro o fare la spesa. **Per svolgere tale compito l'automobile deve essere in efficienza** ogni volta che se ne ha bisogno. **Quindi, accanto alle spese di acquisto, il proprietario deve prevedere le spese di carburante, olio e manutenzione periodica o straordinaria.**

In caso di incidente l'auto deve essere riparata e si vorrebbe poter disporre di un'auto di cortesia in caso di non disponibilità prolungata a più giorni. Quindi in quest'ottica per il proprietario **esiste un *servizio auto* che garantisce gli spostamenti**, a fronte di



un *canone* composto dalla somma delle spese di acquisto iniziale (o rateizzato), delle spese del carburante, olio, manutenzione, bollo e assicurazione ecc... suddivisa per il periodo di uso dell'automobile stessa.

E per l'azienda che produce l'automobile questa è un prodotto? Non solo, nel senso che **accanto alla costruzione delle singole auto l'azienda** deve predisporre **servizi di manutenzione ed assistenza**, sia rivolti direttamente ai clienti come, per esempio, il richiamo e la riparazione gratuita in caso di difetti di produzione in un lotto, sia rivolti alla propria rete di concessionari e autofficine autorizzate, come la distribuzione dei pezzi di ricambio "originali", la formazione dei meccanici sui nuovi modelli ecc...

Non solo: la qualità di tali servizi influenzerà enormemente la scelta del cliente di acquistare di nuovo un modello di quella particolare casa automobilistica.

A maggior ragione quando il cliente ha maggiori possibilità di spesa. Inoltre le auto sono sempre più dotate di software, software che deve essere aggiornato periodicamente, come quello di altri dispositivi. Quindi sia per l'azienda produttrice, sia per il cliente finale, il **prodotto auto** è, in realtà, la parte materiale del **servizio auto** sopra definito.

Negli ultimi anni si è così assistito ad una transizione verso la gestione completa dell'auto come servizio. Il cosiddetto noleggio di lungo termine, in cui si paga un canone nel tempo in cambio del disporre di un'auto completa di servizi di assistenza, manutenzione ed assicurazione per un periodo di tempo o per un chilometraggio massimo è un esempio di *servizio auto* completo.

Un altro esempio sono i servizi di car sharing come Car2Go e EniJoy, disponibili in alcune città italiane, **con cui si paga per il noleggio di un'auto per un periodo di tempo limitato.**

E' stato stimato che la presenza di servizi di car sharing potrebbe disincentivare le famiglie che vivono nelle città dall'acquisto della seconda macchina. Questo ha impatto diretto



sulle case automobilistiche, che stanno iniziando a prendere contromisure: il servizio Car2Go è di una società del gruppo Mercedes, che in tal modo estende il proprio business usando le auto di propria produzione.

Ora spostiamo l'attenzione verso un secondo prodotto, più "tecnologico": lo smartphone.

E' molto facile riconoscere qui i concetti già visti in precedenza. Consideriamo inoltre tutti i casi in cui lo smartphone viene acquistato insieme ad un contratto di servizio telefonico, pagando la rata di acquisto con il canone dello stesso fornitore, oppure ai casi in cui lo si acquista con rateizzazioni.

Ed ecco che il *prodotto fisico smartphone* (senza dimenticare che il suo software viene comunque aggiornato di frequente tramite un apposito servizio online...) **diventa in realtà il servizio smartphone, terminale personale del servizio telefonia e connettività dati mobile** fornito da un operatore telefonico.

E ora **prendiamo in esame servizi nuovi, che hanno trasformato prodotti fisici, come un disco, un CD o un libro, in servizi** che, in cambio di un canone mensile o annuale, consentono l'accesso a raccolte di brani musicali in formato digitale, o a biblioteche virtuali. **Anche il download a pagamento di singoli brani o di libri elettronici è da considerarsi un servizio e non una vendita di prodotti digitali:** il file del brano o del libro rimangono, de iure o de facto, di proprietà del fornitore, che ne può revocare l'uso in qualsiasi momento...

La trasformazione in atto...

Gli esempi precedenti ci dimostrano che è **in corso una trasformazione, resa possibile dai nuovi sistemi IT e, in prospettiva, dai Big Data.** Le aziende stanno progressivamente **trasformando**

molti dei prodotti di valore (e prezzo) più elevati in servizi, servizi che continuano durante la vita operativa dell'oggetto e che tendono a fidelizzare il cliente e a fargli rinnovare il servizio con un nuovo prodotto al termine della vita operativa del primo esemplare.

Questo è esattamente il processo con cui una compagnia telefonica ci vende lo smartphone, oggetto terminale del servizio di telefonia mobile. E il servizio prevede l'analisi dei dati di uso dell'oggetto (e del servizio stesso) da parte del cliente.

Se pensiamo ai sistemi di analisi per la previsione degli interventi di manutenzione, ormai in uso presso i modelli di auto di fascia alta della maggior parte dei produttori, l'evoluzione è proprio questa.

L'auto raccoglie dati sull'uso e sullo status dei propri componenti attraverso un sistema di sensori connesso con la centralina. La centralina comunica periodicamente i dati al costruttore tramite una scheda dati di telefonia mobile. E il costruttore usa questi dati sia per invitare il cliente ad una manutenzione poco prima che qualche componente si possa guastare, sia per offrirgli altri servizi, magari in partnership con altre aziende.

Le Google Car collegate in permanenza con Google Maps che segnalano all'automobilista in giro per turismo luoghi turistici, alberghi, ristoranti o altro in base alle preferenze da lui espresse sono ormai dietro l'angolo.

Tutto questo fa parte dell'ormai prossimo mondo dell'Internet delle Cose, che sarà il tema del prossimo capitolo.



Nel capitolo precedente abbiamo parlato dell'evoluzione progressiva che sta avvenendo nella nostra società e nei nostri mercati, con una transizione progressiva dal concetto di vendita di prodotti "materiali" a quello della vendita di servizi, di cui i prodotti rappresentano la parte "materiale", attraverso cui il cliente usufruisce del servizio.

Le grandi infrastrutture di calcolo disponibili in cloud, i sistemi Big Data che vi si appoggiano, le connessioni in rete disponibili sempre più universalmente e i dispositivi terminali come smartphone, smart device e smart meter (ad esempio, sensori di temperatura e contatori elettronici) hanno un ruolo determinante in questa trasformazione.

In questo capitolo approfondiamo il concetto di Internet delle Cose (nota anche con l'acronimo IoT dalle iniziali della sigla inglese Internet of Thing), ossia l'insieme di dispositivi, più o meno autonomi, che dialogano fra di loro e con le grandi infrastrutture di calcolo (data center) attraverso la rete, già accennato nei capitoli precedenti.

Per comprendere meglio tutto questo partiamo con esempi tratti dallo scenario della vita quotidiana. **Nella nostra stessa casa, ad esempio, gli elettrodomestici di ultimissima generazione sono già in grado di "dialogare" in rete fra di loro per coordinare meglio le funzioni, per esempio per stabilire il momento migliore perché la lavatrice avvii il proprio ciclo di funzionamento in relazione al carico**



elettrico ed alla tariffa.

Sensori di temperatura, umidità ed altri parametri ambientali, posti in varie parti di edifici, possono dialogare con i sistemi di climatizzazione per ottimizzare il loro funzionamento e garantire un ambiente ottimale. **Sensori di luminosità possono interagire con le lampade a led** per garantire il livello di luminosità ottimale durante il giorno, **minimizzando nel contempo i consumi.**

Giocattoli “intelligenti” possono interagire con i bambini per aiutarli nell’apprendimento di nozioni e concetti. **La smart TV può diventare il “pannello di controllo della casa” e noi possiamo usare lo smartphone come telecomando.**

Nel capitolo precedente abbiamo esaminato quanto accade per l’auto, che sta evolvendo sempre più verso la frontiera della guida autonoma, permettendo al passeggero di usare il tempo del viaggio per attività più produttive.

Esperti di economia affermano che questo avrà un impatto notevole sulla produttività del lavoro e sul PIL... **Similmente può avvenire in ambito industriale:** già esistono, in varie regioni italiane, impianti di trattamento e depurazione delle acque che non richiedono presenza in loco di personale umano, se non in caso di guasti, e possono essere telecontrollati da centrali distanti anche decine di km.

Alcuni autori, per indicare questo scenario di umani, dispositivi e data center che dialogano fra loro attraverso la rete, usano il termine Internet di Ogni Cosa (IoE o IoX, derivato dalle parole inglesi Internet Of Everything).

Basandosi su questa infrastruttura diventano possibili tanti servizi per l’utente finale, dal calcolo di percorsi di viaggio che passano “automaticamente” attraverso i ristoranti e locali preferiti, al tele-monitoraggio della casa delle vacanze, con possibilità di accendere il riscaldamento mentre si sta andando in montagna e trovare la casa calda al proprio arrivo.

Le aziende del settore stanno facendo a gara per inventare servizi sempre nuovi che possano aprire nuove opportunità di mercato. **In particolare, nella restante parte del capitolo vedremo le applicazioni dell'Internet delle Cose a una nuova frontiera della medicina, la diagnosi preventiva di massa.**

IoT e Salute: monitoraggio e diagnosi preventiva di massa

Negli ultimi anni sono apparsi strumenti di wearable computing (ossia, essenzialmente, dispositivi elettronici contenuti in capi di abbigliamento od ornamento) **legati al mondo del fitness.**

App dedicate, come per esempio la tedesca Cardiofit, installate sui nostri smartphone, possono interagire con attrezzi sportivi "smart", come per esempio ergometri, tapis roulant, biciclette... o anche con magliette con sensori, cinture o semplicemente il nostro smart watch.

La combinazione dei dati raccolti nel tempo da questi sensori ed immagazzinati in servizi cloud ci permette di controllare il nostro stato di forma fisica e anche, se vogliamo, di condividere percorsi di trekking o ciclismo con gli amici, **magari corredati di foto e selfie scattati durante la nostra impresa sportiva.**

Qual è il passaggio successivo? Pensiamo a cosa è l'ECG dinamico secondo Holter: una metodica diagnostica utilizzata per monitorare l'attività elettrica del cuore durante un intervallo di tempo più o meno lungo, solitamente corrispondente a 24-48 ore, e con i registratori di ultima generazione sino ad un massimo di sette giorni. **La stessa metodica si può applicare anche alla pressione sanguigna o ad altri parametri vitali.**

Tramite uno strumento integrato nello smart watch o in un

braccialetto si potrebbero ottenere combinazioni di questi parametri. Per esempio pulsazioni cardiache, pressione sanguigna, composizione chimica del sudore durante una prova sportiva, conducibilità elettrica della pelle in condizioni normali... e per periodi molto lunghi, praticamente con continuità.

Da alcuni anni grandi attori del mercato ICT come SAP, IBM e Google, stanno realizzando, usando le tecnologie dei Big Data, enormi banche dati di cartelle cliniche, con la storia di milioni di pazienti. Analizzando l'insieme di questi dati con le metodologie di intelligenza artificiale oggi disponibili diventa possibile stabilire rapporti statisticamente validi che conducono a relazioni causa-effetto (o, meglio, sintomo-malattia) fra parametri rilevabili tramite sensori, come quelli sopra descritti, e stati di malattia, prima del loro insorgere.

In sostanza, come funzionerà il servizio di diagnosi preventiva? Sensori contenuti in dispositivi indossabili monitoreranno un insieme di parametri fisiologici sul nostro corpo e, combinando questi dati con quelli delle attività principali che staremo svolgendo, rilevabili dal nostro smartphone, sarà possibile associare nel corso della giornata stato fisiologico ed attività, come si fa nell'ECG dinamico sopra descritto.

Questi dati potranno essere raccolti più volte nella giornata e immagazzinati in un archivio dati. Confrontando, a scadenze periodiche opportune, questi dati con i profili statistici a noi vicini come età, sesso, abitudini e profilo genetico sarà possibile scoprire discrepanze rispetto ai valori ottimali, che potrebbero essere indice di qualcosa. Segnalando tempestivamente tali discrepanze e suggerendo gli esami specialistici necessari per indagare nel merito i sistemi automatici, eventualmente corredati dal parere di medici esperti umani, **realizzeranno la medicina preventiva di massa, per tutti gli utenti di questi servizi.**

E al secondo livello, **i risultati degli esami specialistici, confrontati con i modelli tratti dalle banche dati, consentiranno di arrivare in**

tempi rapidi ad una diagnosi precisa, prima che la malattia si manifesti con sintomi più evidenti. In sostanza si tratterà di un raffinamento progressivo della diagnosi, sempre più preciso. La nostra storia personale precedente, raccolta entro le banche dati nel corso degli anni, porterà a prestazioni ancora migliori per il sistema.

Dati statistici sul mercato USA, estrapolati, indicano che le diagnosi compiute dai soli medici umani sono corrette solo tra il 70 e l'80% dei casi. Sebbene le diagnosi puramente automatiche, con i sistemi odierni, abbiano un tasso di errore ancora superiore, la combinazione di medico umano e sistema di intelligenza artificiale consente di arrivare ad una diagnosi corretta in più del 90% dei casi già oggi. **Sistemi basati sulle banche dati oggi in via di realizzazione potrebbero condurre a diagnosi corrette nel 99% dei casi.**

Lo scenario che si prospetta...

Chiediamoci ora: lo scenario che abbiamo innanzi è quindi ottimale? Non del tutto: si intravede già anche la possibilità di situazioni piuttosto preoccupanti.

Il monitoraggio continuo di quello che facciamo e delle associate reazioni del nostro corpo, infatti, se i dati non saranno adeguatamente protetti, potrebbe portare a situazioni quanto meno imbarazzanti. Per esempio, se in presenza non del nostro coniuge, ma di un'altra persona (fattore rilevabile dalla vicinanza dei due smartphone) il nostro battito cardiaco aumenta sempre e sono presenti altri segnali di forte emozione, questo cosa significa?

In sostanza la nascita di sistemi di monitoraggio di massa di persone come quelli sopra descritti deve prevedere anche sistemi



di sicurezza, ossia di garanzia di riservatezza, esattezza ed integrità e disponibilità dei dati adeguati. **Il che probabilmente potrebbe essere realizzato riducendo al minimo o eliminando del tutto il fattore umano dal monitoraggio ed acquisizione dei dati stessi. Ed evitando le correlazioni dei dati fra persone diverse.**

Tutto ciò premesso, si arriverà a una messa a sistema dei sistemi descritti? La risposta è quasi sicuramente sì, almeno nelle società occidentali, e basata su motivi essenzialmente economici:

- ✓ **Il costo della sanità attuale è sempre più elevato**, la riduzione della disponibilità finanziaria dello stato sta portando a tutte quelle conseguenze di peggioramento del servizio che siamo osservando.
- ✓ **Molte delle malattie** gravi oggi esistenti, **se scoperte e trattate nelle primissime fasi del loro insorgere, sono curabili** e le cure sono molto meno costose; alcuni check-up di massa già oggi realizzati sono convenienti economicamente anche per questo motivo.
- ✓ **Il tasso di diagnosi sbagliate per un singolo medico umano è, come abbiamo visto, elevato**: la disponibilità di basi di conoscenza basate sulla storia clinica di centinaia di milioni di persone ridurrà tali errori.
- ✓ **L'invecchiamento progressivo della popolazione condurrà le aziende** che offrono le assicurazioni sanitarie **a proporre ai propri assistiti servizi come quelli descritti**, per guadagnare poco su tanti sani, riducendo le spese di cure con la prevenzione di massa.

La nostra salute, dunque, sarà monitorata da sistemi automatici e sparirà progressivamente, o meglio cambierà ruolo, la figura del medico di famiglia attuale? **Molto probabilmente sì, e in un futuro non lontano.**



Perché questo sia possibile occorre però che i sistemi a cui affideremo la nostra salute futura siano definiti nelle specifiche funzionali, progettati, realizzati, collaudati e mantenuti in esercizio in modo adeguato. **Non ci potrà essere alcuno spazio per l'improvvisazione, ne andrà della nostra salute e della nostra vita. E i sistemi informatici su cui questi servizi si baseranno sono progettati da professionisti dell'IT.**

Nei prossimi capitoli riprenderemo in esame le professionalità dell'IT, con particolare riguardo ad alcuni ruoli fondamentali.



Business Analyst: sinergia tra IT e Business per guidare l'azienda in un mondo che cambia. Donna è meglio?

Abbiamo già visto insieme quanto l'IT sia diventata e stia tuttora diventando sempre più pervasiva nella nostra vita e nel nostro lavoro, sottolineando anche come questo possa rivelarsi utile per la nostra salute.

Per tutti questi motivi occorre un **allineamento** ed una **integrazione** sempre maggiore fra l'IT con le sue potenziali soluzioni ed il Business. L'IT è infatti ormai il **fattore abilitante principale per qualsiasi business**. Ad esempio, se una azienda di moda vuole aprire nuovi negozi in una parte del mondo, deve provvedere risorse ed azioni per far sì che questi negozi siano correttamente connessi con la sua rete aziendale.

Quindi l'IT deve poter provvedere proprio a quello che le viene chiesto e, dall'altra parte, le richieste devono essere estremamente chiare e precise. Cosa non sempre facile.

Questo significa, di conseguenza:

Definire con precisione i bisogni per il business in funzione delle attività strategiche, tattiche e/o operative che l'azienda sta per intraprendere.

Valutare la situazione di partenza (il cosiddetto AS-IS).

Definire l'obiettivo, in cui una soluzione tecnica ed organizzativa risolve i bisogni, da realizzarsi entro un tempo accettabile (il cosiddetto TO-BE).



Chi ha in azienda tale ruolo, deve dunque poter parlare con persone diverse (dalla direzione al personale operativo), deve poter raccogliere le informazioni da tutti e trasformarle (con una visione integrata di insieme) nelle specifiche precise per costruire una soluzione, e – come in parte abbiamo spoilerato nel titolo:) – deve avere una pluralità non comune di competenze, sia personali che professionali.

Ma chi è la figura quasi “magica” in grado di essere – e dunque fare – tutto questo questo?

In realtà non c'è niente né di magico né di impossibile da raggiungersi: l'inquadramento di tale figura si basa su di una scienza e una metodologia ormai ampiamente codificata chiamata **Business Analysis** (in italiano tradotto spesso come analisi business o analisi aziendale) e il **magico (o la maga) in questione è un professionista *normato e codificato* a livello di Unione Europea, ovvero il Business Analyst.**

In questo capitolo prenderemo quindi in esame l'Azienda e il ruolo del Business Analyst al suo interno, per mettere a fuoco le capacità di cui deve disporre questa figura essenziale per il successo del business, il cui ruolo e la cui influenza vanno ormai ben oltre l'IT.

Come funziona l'azienda?

Ogni azienda opera per fornire prodotti e servizi ai propri clienti e da questo trae il proprio guadagno. Per ottenere questo entro l'azienda vengono compiute dai dipendenti successioni di attività collegate logicamente, i cosiddetti processi aziendali o processi business (in inglese Business Process).

Ogni processo partendo da un input realizza un output, che ha valore per il cliente. E il cliente può essere esterno all'azienda – nel

qual caso il valore è monetizzabile e costituisce un ricavo per l'azienda – oppure interno, come ad esempio avviene nei processi amministrativi o di gestione del personale, che hanno come clienti altre parti dell'azienda.

I processi possono essere produttivi e costruire materialmente prodotti, come nel caso delle aziende manifatturiere. O gestionali, come i processi di gestione delle fatture, del personale ecc... e trattare informazioni.

I processi, inoltre, possono essere espliciti e codificati in procedure documentate, come avviene, ad esempio, nelle aziende certificate ISO9001, **oppure impliciti ed esistere nella mente dei dipendenti**, che sanno quale è il loro compito e cosa devono fare per fare sì che l'azienda nel suo insieme realizzi il proprio obiettivo. In taluni casi manca la visione di insieme e magari l'azienda non è dappertutto efficace ed efficiente.

Cosa rappresenta la conoscenza dei processi? Semplicemente il modo con cui l'azienda funziona ed opera e comprende in sé il know-how delle specifiche operazioni. Addestrare un neoassunto a svolgere le sequenze di azioni che un veterano compie in modo quasi automatico richiede tempo. Se alcuni processi non sono codificati e documentati, all'uscita di una persona occorre effettuare un passaggio di consegne, trasferendo la conoscenza tacita che forma i processi taciti ad altre persone, altrimenti l'azienda perde una parte del suo funzionamento. **Ecco perché i processi sono così importanti economicamente.**

La analisi, formalizzazione e documentazione dei processi, che viene compiuta ad esempio durante le certificazioni ISO9001, rende quindi i processi patrimonio aziendale, trasferibile da una persona all'altra attraverso la lettura della documentazione, "estraendoli" dalla mente delle persone dove si trovano quando sono ancora taciti. La codifica di procedure ed istruzioni di lavoro o work instruction in modo semplice consente anche di ottimizzare i processi stessi, diminuendone tempi e/o costi e anche di



addestrare rapidamente i neoassunti a svolgere le operazioni.
Questo è uno dei compiti svolti da un Business Analyst.

Congiungere IT e Business

L'analisi dei processi è il primo passo per la loro ottimizzazione. Tale ottimizzazione può avvenire progressivamente, con il miglioramento di tutte o di alcune delle attività la cui successione forma il processo stesso (procedimento chiamato Business Process Improvement).

Oppure con la ridefinizione del processo, con la sostituzione o la modifica di alcune o tutte le attività che lo formano in modo molto più "invasivo" (procedimento chiamato Business Process Reengineering).

Oggi si tende ad usare un approccio evoluzione del primo, ma che combina le caratteristiche positive di entrambi, chiamato Continuous Process Improvement (miglioramento continuo dei processi), recepito anche nelle normative della qualità. Anche le nuove leggi, come il GDPR, richiedono in alcuni casi l'organizzazione di processi espliciti, **come è stato spiegato in [questo articolo](#).**

E oggi, per i processi amministrativi e gestionali l'IT, in qualsiasi azienda, interviene massivamente, automatizzando insiemi di operazioni sull'informazione trattata entro i processi. E anche nei processi produttivi l'OT (Operation Technology), **introdotta in [questo articolo](#)**, è ormai fondamentale.

Quindi il compito complessivo del Business Analyst è studiare i processi e come funziona l'azienda, individuare i punti di miglioramento (ad esempio, riduzione di tempi e/o costi nello svolgere le attività) **e risolvere problemi riscontrati o definire le risposte possibili ad opportunità** che si presentano sul mercato,

come la nascita di nuovi mercati o settori di business. Questo rende l'azienda agile e veloce nel cambiare, adattandosi ad un ambiente che ormai può mutare con velocità semplicemente impensabili fino a pochi anni or sono.

Il cambiamento può avvenire con l'introduzione di mutamenti organizzativi (nuovi processi, nuove suddivisioni di reparti...), **di nuove soluzioni tecnologiche** (nuovi applicativi, nuovi strumenti...), o, nella maggior parte dei casi, **con una combinazione di entrambi**, che porta al risultato migliore. Il semplice inserimento di un nuovo strumento tecnico, senza che i processi si adattino alla sua presenza, per poterne sfruttare al meglio le potenzialità, non riesce ad ottenere tutti i vantaggi possibili dall'investimento economico necessario.

Il Business Analyst quindi deve:

- ✓ *Rilevare i bisogni, espliciti o latenti, del cambiamento.*
- ✓ *Definire la situazione attuale con precisione.*
- ✓ *Definire l'obiettivo per il cambiamento.*
- ✓ *Definire i passi da compiere, compatibili con le risorse esistenti, per ottenere gli effetti positivi del cambiamento.*
- ✓ *Documentare tali passaggi ed, eventualmente, verificare il loro andamento per l'ottenimento del successo.*

Spesso, inoltre, il Business Analyst fa anche il progetto di **dettaglio della soluzione IT**, definendo con precisione cosa essa dovrà fare entro il contesto dell'azienda (la cosiddetta analisi funzionale di un prodotto IT) e magari procede anche, dopo la



realizzazione, all'addestramento delle persone.

Quindi il Business Analyst interviene dal momento in cui il bisogno (risoluzione di problema o sfruttamento di opportunità) inizia ad esistere, lungo tutta la costruzione di una soluzione al bisogno, fino alla sua entrata in servizio, curando spesso l'addestramento necessario per le persone.

Processi, dati ed informazioni

I processi hanno dei flussi di informazioni al loro interno, attraverso le quali si coordinano le attività collegate logicamente e/o cronologicamente che li formano. Ed esistono flussi di informazioni anche fra i vari processi che operano entro l'azienda.

I flussi possono essere legati alle attività operative come la vendita, la produzione, la gestione magazzino, **oppure collegare le attività produttive con quelle tattiche e strategiche svolte al livello della dirigenza aziendale. E possono infine contribuire ad estrarre dalla mole di dati grezzi l'insieme delle informazioni utili ad aumentare la conoscenza sullo stato dell'azienda e dei mercati, il che rende possibile alla direzione aziendale il prendere con saggezza le decisioni vitali per il futuro dell'azienda stessa.**

Compito del Business Analyst è anche definire come trattare questi dati, attraverso l'uso di quella branca dell'IT chiamata Business Intelligence, che oggi, attraverso i Big Data e l'intelligenza artificiale, sta evolvendo in qualcosa di completamente nuovo, in grado quasi di prendere decisioni in modo autonomo.

In questo compito, il Business Analyst è affiancato da due figure specialistiche che in qualche modo ne sono una evoluzione: il Business Information Manager, che possiamo definire anche il gestore aziendale della conoscenza, e il Data Scientist, che "disegna" i trattamenti automatici per passare dai dati alla

conoscenza.

L'analisi dati per estrarne conoscenza è una necessità per ogni azienda, in quanto da ad essa una consapevolezza con cui agire con forza e rapidità nel mercato in continuo mutamento, agendo anche d'anticipo rispetto ai mutamenti. Dalle grandi aziende, via via, il suo uso sta espandendosi a cascata anche verso le aziende più piccole.

Il profilo codificato del Business Analyst? Donna è meglio!

Esistono diverse certificazioni professionali per il Business Analyst. In particolare le normative UNI 11506:2017 e UNI 11621-2, recepite a livello di tutta l'Unione Europea, definiscono con precisione il profilo professionale "Business Analyst" come colui/colei che:

- ✓ *Identifica aree dove sono necessari cambiamenti del sistema informativo per supportare i piani di business e ne controlla l'impatto in termini di gestione del cambiamento.*
- ✓ *Contribuisce ai requisiti funzionali generali dell'azienda per quanto riguarda l'area delle soluzioni basate sull'IT.*
- ✓ *Analizza le esigenze di mercato e le traduce in soluzioni IT.*

Per fare questo il Business Analyst deve comprendere come funziona l'azienda (o almeno una sua parte), i suoi processi, il mercato. Deve interagire proficuamente con le persone, ai livelli manageriali e a quelli operativi. Dovendo guidare anche il cambiamento, il ruolo del Business Analyst è, in taluni casi, quello



di interagire con le persone coinvolte, motivandole nell'accettare ed anzi avvantaggiarsi del cambiamento, svolgendo quasi un'azione di Business Coaching.

Deve quindi avere grandi capacità di analisi, conoscere il business, identificare modelli e situazioni note in cui poter applicare, ad esempio, soluzioni dimostrate di comprovata efficacia (come avviene, ad esempio, per i framework organizzativi legati alle certificazioni ISO, ITIL, COBIT ecc...).

E deve conoscere, almeno ad alto livello, la tecnologia e le soluzioni. **E, soprattutto, deve avere grandi doti di comunicazione interpersonale, deve saper parlare diversi linguaggi, entrando quasi in empatia con le persone con cui interagisce.**

Per questo, in genere – e come suggerito dal titolo dell'approfondimento – **si tratta di uno di quei ruoli in cui, a parità di altre capacità e conoscenze, una donna può riuscire mediamente meglio di un uomo**, essendo in genere sia più empatica che dotata di maggiore pazienza.

Nel prossimo capitolo estenderemo il concetto di comunicazione fra professionalità diverse, dotate di diversi linguaggi specialistici, diversi modi di pensare e diversi approcci ai problemi nel contesto lavorativo. Necessità fondamentale per creare quei team interdisciplinari necessari, ad esempio, **per svolgere funzioni di trasformazione che, come ad esempio nel caso dell'adozione del GDPR, richiede interventi legali, organizzativi e tecnologici.**



Abbiamo già visto insieme i nuovi scenari di servizio resi possibili dalle nuove tecnologie, il ruolo delle persone entro i settori tecnici e specialistici, in primis l'ICT, e analizzato con precisione la figura del Business Analyst, che deve essere dotato di grandissime doti comunicative.

Vediamo ora più nel dettaglio questo focus, ovvero analizziamo il ruolo del professionista tecnico-specialistico nei suoi aspetti di vero e proprio "comunicatore".

Oggi, infatti, sia nel lavoro che nella vita quotidiana, operiamo in un mondo sempre più complesso e interdipendente. Le aziende e le organizzazioni tendono ad essere divise in compartimenti funzionali (ufficio vendite, ufficio acquisti, controllo qualità...) dotati a loro volta di interfacce di comunicazione attraverso cui interagiscono.

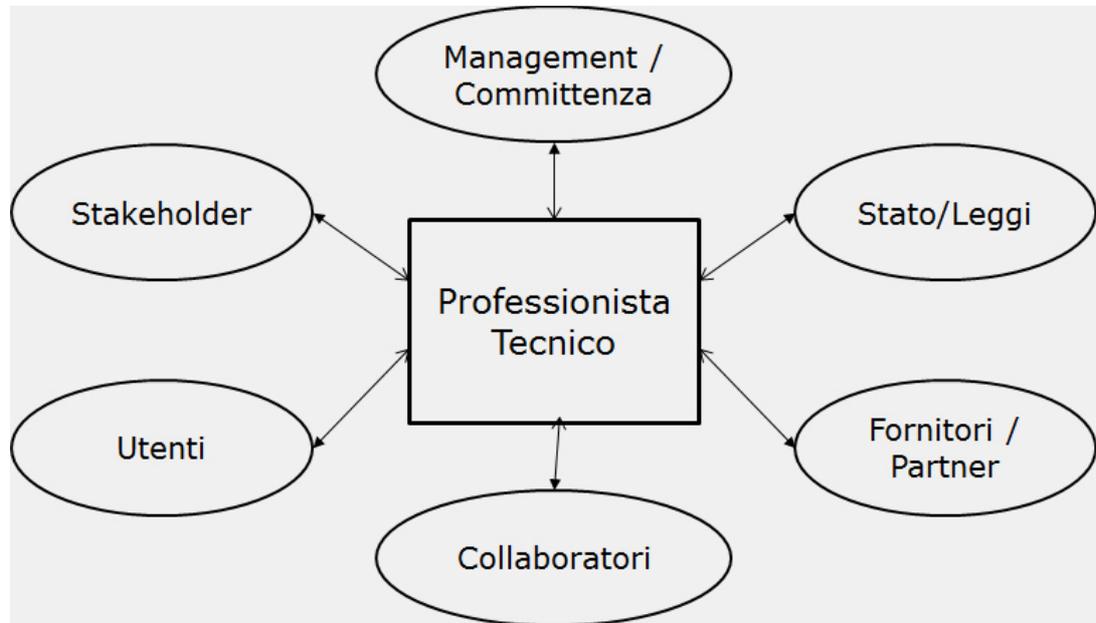
L'iper-specializzazione degli individui, inoltre, tende a rendere la comunicazione più difficoltosa: se un esperto di marketing deve comunicare con un esperto di infrastrutture informatiche, ad esempio, ciascuno dei due si cimenterà (quasi) con un'altra "lingua".

E tuttavia, mai come oggi, è necessario comunicare con tutti gli interlocutori coinvolti nei processi aziendali, per fare sì che l'impresa stessa svolga per intero il suo ruolo nel mercato.

Chi si trova a gestire ruoli di analisi e/o coordinamento (come ad esempio vale per i Manager, i Project Manager, i Business



Analyst o i Direttori di Cantiere nel contesto di Ingegneria Civile e Industriale), diventa molto spesso l'interfaccia privilegiata di comunicazione fra entità organizzative differenti, a loro volta composte da individui che hanno conoscenze, storia professionale e linguaggi specialistici diversi, come viene schematizzato nella figura seguente:



Gli stessi team di lavoro – in molti casi – sono eterogenei in partenza, e riuniscono al proprio interno professionalità diverse. Un gruppo di lavoro sulla sicurezza delle informazioni e/o sulla privacy, per esempio, potrebbe comprendere informatici, avvocati e ingegneri gestionali, oltre che Etical Hacker.

In una prima approssimazione si può pensare che, per realizzare questa comunicazione in modo efficace, sia sufficiente uno dei metodi classici di Business Analysis degli anni '80 e '90, che presuppone che basti imparare il lessico dei vari settori e usarlo con precisione nella comunicazione.

In fin dei conti – si ritiene il più delle volte – si tratta di comunicazione lavorativa, e dunque per sua definizione puramente razionale, a basso contenuto emotivo e destinata alla mente



logico-razionale dei destinatari.

Comunicazione intrinsecamente diversa, quindi, da quella orientata alla vendita e/o al marketing, a maggior ragione in un settore all'apparenza puramente logico come l'ICT... **Ma è veramente così? Non proprio, come vedremo tra poco.**

Le norme stesse lo dicono: bisogna saper comunicare!

Nella scheda del profilo del Business Analyst definita dalle norme UNI 11506, UNI 11621-2 e EAN 16348 (normativa dell'Unione Europea che estende a tutta Europa le due precedenti) **si individuano come competenze assolutamente necessarie le seguenti, definite a loro volte nell'European E-Competence Framework:**

- ✓ *Allineamento strategie tra Sistemi Informativi e Business.*
- ✓ *Sviluppo del Business Plan.*
- ✓ *Miglioramento di Processo.*
- ✓ *Identificazione dei fabbisogni.*

E nella definizione di queste competenze, a sua volta l'E-Competence Framework definisce come basilari, tra le altre competenze e *skill*, le:

- ✓ *tecniche di comunicazione;*
- ✓ *tecniche di narrazione ("Storytelling");*
- ✓ *capacità di proporre i cambiamenti di processo*



(che richiede enormi capacità comunicative e di convincimento).

Analogamente, nella normativa ISO/IEC 27021, che definisce il ruolo del professionista dei sistemi di gestione della sicurezza delle informazioni, sono nominate esplicitamente come conoscenze fondamentali:

- ✓ *Leadership.*
- ✓ *Teoria e metodi di comunicazione.*
- ✓ *Tecniche di comunicazione.*

Quindi, già a livello normativo, viene riconosciuta la necessità di saper comunicare anche per figure in apparenza tecniche o di congiunzione fra il settore tecnico e il business. I migliori manager sanno già che saper comunicare è fondamentale in qualsiasi ambito, anche “tecnico puro”.

Se analizziamo i dettagli delle richieste delle norme sopra elencate troviamo che saper comunicare significa:

- ✓ *Apparire come leader – e quindi essere ascoltati. Guidare nel cambiamento – per motivare e convincere.*
- ✓ *Usare tecniche complesse con contenuto emotivo come prevede lo storytelling.*

La comunicazione tecnica, pertanto, non è soltanto una traduzione di contenuti, ma ha anche un forte impatto emotivo. Ecco quindi che, per realizzare una comunicazione tecnica efficace, presupposto fondamentale per il successo di progetti ed aziende, bisogna costruire una metodologia molto più ampia.

Un modello per la comunicazione tecnico-specialistica efficace

Nei miei anni di esperienza sul campo – esercitata in vari ruoli nei settori sia organizzativi che e tecnologici – **ho potuto vedere, sentire e toccare con mano quanto, durante una comunicazione tecnica, sia importante l'aspetto emotivo e la vicinanza al "modello mentale" dell'interlocutore.**

Allo stesso modo – studiando Coaching, PNL per la comunicazione, intelligenza emotiva e neuro-marketing – **ho potuto apprendere e comprendere le basi culturali, psicologiche e percettive che stanno alla base di tali punti di vista.**

Infine, dopo avere insegnato questo tipo di comunicazione in molte edizioni di corsi per Business Analyst e Project Manager, mi sono reso conto della necessità di applicare tali metodologie in tanti altri contesti lavorativi, sia professionali che aziendali, ricavandone infine un corso specifico che ho erogato con successo in contesti complessi quali ad esempio gli ordini professionali e gli istituti di ricerca. A riprova ulteriore di come il tema sia assolutamente sensibile.

Per comprenderne i "fondamentali" di tale approccio, partiamo dunque da un presupposto di base, ovvero:

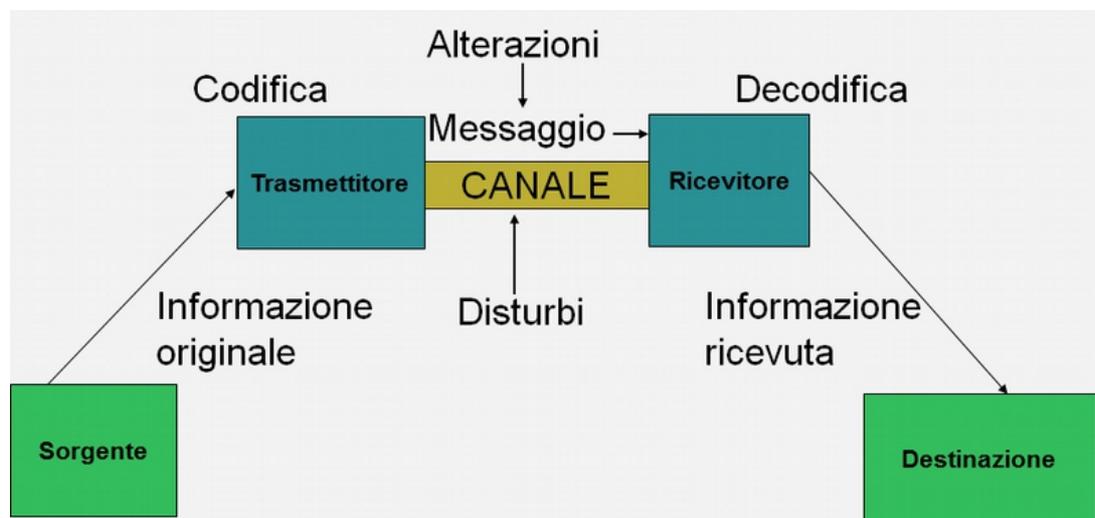
Cosa è la comunicazione?

Tradizionalmente, essa si definisce attraverso gli scopi che persegue nonché gli attori che investe (e il loro contesto) e infine la tecnologia che utilizza per dispiegarsi. Il filo "narrante" del processo comunicativo è così riassumibile:



- ✓ *lo scopo della comunicazione è trasmettere informazione da un mittente ad un destinatario;*
- ✓ *ricevere informazione aumenta la conoscenza del destinatario;*
- ✓ *il destinatario, di solito, attiva una re-azione all'informazione ricevuta.*

Il modello base per la comunicazione è mostrato nella figura sottostante:



A questo punto, è importante soffermarci su alcuni concetti specifici.

Nel caso della comunicazione interpersonale, infatti, la sorgente è la mente del mittente, che contiene le "idee" che a loro volta vengono convertite in parole attraverso la conoscenza del mondo del mittente (ossia il suo "modello mentale") e le relative associazioni idee-parole (tramite il processo di codifica delle idee in parole).

Nel caso di linguaggio parlato, inoltre, è bene ricordare che, accanto alle parole, tante informazioni vengono trasmesse anche



dal modo con cui le pronunciamo (linguaggio paraverbale) e dalle espressioni del viso e del corpo con cui le accompagniamo (linguaggio non verbale).

Occorre anche avere presente che – come dimostrato già negli anni '60 da Albert Merhabian – in caso di non coerenza fra il contenuto verbale, paraverbale e non verbale di un messaggio, l'ultima e la seconda prevalgono sulla prima. E di questo ci accorgiamo in tutti quei casi in cui, pur essendo il contenuto verbale impeccabile, "qualcosa" ci ha trasmesso non coerenza, creando in noi un senso di sospetto e diffidenza.

Nel caso del **linguaggio scritto**, infine, **la punteggiatura, il corsivo e il grassetto possono corrispondere alla componente paraverbale**, mentre l'uso dei *font* e la formattazione possono **ricorrere alla componente non verbale**.

In generale, comunque, accade che **il messaggio, sotto forma di onde sonore** (per le componenti verbali e paraverbali) **e di luce** (per le componenti non verbali) **viaggia nell'aria e giunge ai ricevitori del destinatario** (ossia gli organi di senso), **che compiono la prima parte della decodifica**.

Ma non tutto fila sempre così liscio, anzi: **durante il tragitto il messaggio può subire alterazioni** (per esempio, in una telefonata con qualità dell'audio pessima) **o essere offuscato da una serie di disturbi** (si pensi, ad esempio, a una conversazione in ambiente rumoroso). **Facilmente, quindi, potrà giungere al destinatario in modo imperfetto**.

Come se non bastasse, una volta raggiunto il destinatario, **il messaggio, per essere compreso, necessiterà di un ulteriore e complesso processo di elaborazione da parte del ricevente**.

Se ne deduce che **ciascun mittente** – se vuole conseguire il suo scopo ed essere efficace nella sua esposizione, riducendo a priori i possibili rischi di fraintendimento o non comprensione – **deve conoscere e comprendere a priori tutte queste caratteristiche**



della comunicazione, in maniera da garantire il più possibile il suo approdo efficace alla destinazione finale. Destinazione che – nel caso la comunicazione sia di tipo tecnico – è definibile come la componente logico-razionale della mente del destinatario.

Essere consapevoli di ciascuno di questi meccanismi, inoltre, non serve solo per acquisire leadership o vendere bene, ma anche e soprattutto per lavorare meglio ogni giorno.

E' dunque importante che questo tipo di competenza – conforme tra l'altro a quanto richiesto dalle normative – diventi progressivamente sempre più patrimonio di base dei professionisti dei settori tecnici ed organizzativi.



05 Il compito fondamentale in ICT: lo sviluppo del codice sorgente.

In alcuni capitoli precedenti, dopo il riconoscimento del ruolo fondamentale dell'IT nel XXI secolo, **abbiamo trattato le figure professionali e la competenza che dovrebbe essere fornita dalla formazione accademica**

In questo capitolo ci concentriamo su una delle fasi più importanti dell'IT, lo sviluppo o scrittura del codice, e sulla figura professionale deputata a tale compito, lo sviluppatore o developer.

Per realizzare una soluzione o un servizio ICT è infatti sempre necessaria una fase di scrittura di codice, che si può suddividere tra la personalizzazione di software esistente e la scrittura vera e propria di nuovo software. Anche alcuni compiti di amministrazione di sistema richiedono la scrittura di codice (ad esempio i cosiddetti programmi batch del settore bancario, o gli shell script tipici del mondo UNIX).

Nel dettaglio, la fase di sviluppo consta materialmente nella scrittura di un insieme di file di testo contenenti istruzioni, organizzate secondo regole sintattiche molto precise e rispondenti alla grammatica del particolare linguaggio di programmazione scelto. L'insieme di tali file di testo che forma un'applicazione viene definito codice sorgente dell'applicazione stessa.

Nella maggior parte dei casi un apposito programma chiamato compilatore traduce poi tali moduli in una forma direttamente eseguibile dal computer. In altri casi il codice viene direttamente "interpretato" ed eseguito senza bisogno di compilazione ad opera



di un altro tipo di programma chiamato appunto interprete.

Codice e linguaggio? Andiamo alla sorgente...

Il codice sorgente viene scritto entro un programma apposito, chiamato editor, molto spesso facente parte di un ambiente integrato di programmazione (chiamato anche IDE, dalle iniziali delle parole inglesi Integrated Development Environment) che contiene tutte le funzioni per compilare il codice sorgente e trasformarlo in un programma eseguibile direttamente dal computer.

I linguaggi di programmazione in uso sul mercato oggi sono tanti, dall'antico e sempreverde COBOL, ai più moderni Java e C#, al Python, ai linguaggi PHP e JavaScript specifici per il Web, a linguaggi dedicati come l'ABAP di SAP, a linguaggi tipici per le App ecc... Molti di questi linguaggi hanno anche il proprio IDE di riferimento, come VisualStudio per il C#, oppure esistono IDE che possono consentire di scrivere in diversi linguaggi, come Eclipse con Java, C/C++, PHP ecc...

Il developer o sviluppatore è la figura professionale cui è demandato il compito di scrivere materialmente tutto il codice sorgente e/o le personalizzazioni. Questa figura professionale rientra fra quelle della normativa UNI11506-11621 ed esistono molte certificazioni proprietarie per developer, legate a linguaggi e tecnologie. Taluni paragonano ingiustamente il developer al muratore che costruisce i pilastri ed i muri che sono stati progettati da qualcun altro o all'operaio sulla linea di montaggio.

In realtà, la scrittura del codice è un'attività non evitabile. Dietro ad ogni software, dai sistemi operativi (ossia i software di base che consentono ai computer, agli smartphone ed ai tablet di operare), ai programmi come Office, ai programmi gestionali, alle



applicazioni Web, alle App, ai programmi che controllano robot e macchine industriali... **c'è sempre questa fase di scrittura del codice sorgente che forma il software stesso.**

La scrittura del codice è anche un'attività critica, in quanto errori commessi in questa fase si traducono in malfunzionamenti e/o costi maggiori durante le fasi successive di manutenzione ed evoluzione dei componenti software delle soluzioni. **Molte delle problematiche di sicurezza informatica riscontrate negli ultimi anni sono proprio dovute ad errori commessi durante la fase di scrittura del codice.** Il costo del software e del suo sviluppo in molti casi supera oggi di molto il costo dell'hardware che lo fa funzionare.

L'evoluzione quasi incredibile che c'è stata nell'hardware e nelle sue potenzialità non è stata purtroppo accompagnata da una contemporanea evoluzione del software. Nonostante fin dal 1968 siano state evidenziate le varie problematiche legate alla produzione del software[1] e al fatto che la scrittura di codice è un'attività compiuta a mano da operatori umani ancora oggi alcune problematiche permangono.

La fase di sviluppo...

Per capire meglio occorre analizzare i dettagli. **Anzitutto quali sono le tipologie di errori commessi durante la fase di sviluppo?** Essenzialmente sono le seguenti:

- ✓ *Errori di interpretazione delle richieste del committente, che di solito sono risolti dall'esperienza del Business Analyst (o del Product Owner in un contesto di organizzazione Agile);*
- ✓ *Errori di strutturazione del software, ad esempio di*



suddivisione dei compiti fra i vari moduli che formano un applicativo, che dovrebbero essere ridotti od evitati a priori con un buon disegno tecnico del software stesso, demandato all'esperienza dei System Analyst e System Architect;

✓ *Errori veri e propri di scrittura del codice che possono essere dovuti a:*

✓ *Problemi di trasmissione delle informazioni fra i membri del gruppo di lavoro (ad esempio ricadere nei casi precedenti, con analisi superficiali e/o incomplete).*

✓ *Problemi di poca conoscenza del linguaggio e/o di sue librerie di funzioni.*

✓ *Problemi di tempo, ossia di dimensioni del lavoro e quindi della quantità di codice da scrivere non adeguata ai tempi concessi per lo sviluppo.*

✓ *Problemi di test e verifiche inadeguate.*

✓ *Problemi di competenze di base degli sviluppatori.*

✓ *Problemi di errori presenti entro strumenti utilizzati per lo sviluppo (in primis librerie di funzioni preconfezionate).*

Alcune problematiche sorgono dal fatto che sino a pochi anni fa si pensava al software come un edificio, ossia come qualcosa che deve essere completato e poi rimane costante o quasi, mentre oggi è chiaro (o dovrebbe esserlo) che il software è un servizio, che evolve in continuazione per seguire le esigenze del business per cui esso è nato.

I software deve quindi essere sin dall'inizio progettato per poter evolversi nel tempo con la massima efficienza possibile.



Diversi rimedi sono stati applicati nel tempo:

Costruzione di software semi-lavorati con conseguente ridotta necessità di scrittura di codice, che si riduce a parametrizzazioni (ossia impostazioni di parametri di configurazione) per adattare software medio-piccoli alle esigenze dei clienti finali. Nel caso di software più grandi, le operazioni di adattamento richiedono comunque una parte di programmazione, detta personalizzazione.

Costruzione di librerie di funzioni molto ricche e molto testate, per ridurre la quantità di codice che gli sviluppatori devono scrivere (e quindi, statisticamente, anche gli errori che possono commettere), vendute commercialmente sul mercato.

Costruzione di strumenti grafici o semi-grafici (ad esempio i cosiddetti RAD) che consentono di generare il codice sorgente a partire da diagrammi e rappresentazioni grafiche di compiti, evitando di scrivere a mano il codice stesso.

Ma quasi tutti questi rimedi hanno portato anche svantaggi, principalmente inefficienza e vincoli di rigidità sui prodotti software ottenuti. Negli ultimi anni la rivoluzione agile, attraverso l'adozione di nuovi metodi di lavoro, ha portato finalmente un miglioramento deciso. In particolare l'adozione del DevOps [2], una metodologia che si concentra sia sullo sviluppo sia sulla fase di esercizio operativo del software, sta portando un miglioramento deciso nella qualità del software stesso.

Per poter essere veramente efficace tuttavia questa nuova organizzazione prevede che i singoli sviluppatori abbiano una notevole abilità ed esperienza o siano coadiuvati da un abile system architect, per costruire un software ben strutturato e ben scritto. E che i team di lavoro siano bene organizzati.

E questo è purtroppo in alcuni casi un punto dolente. Molti grandi progetti software italiani sono ancora basati su metodologie

antiquate (anche per la non conoscenza delle nuove da parte dei capi progetto) e/o soggetti ad un rapido turn-over degli sviluppatori, oltre che a tempi compressi per sviluppo e test. Il risultato è software di cattiva qualità, che poco può competere sui mercati esteri.

Quindi, per poter raggiungere l'obiettivo di fornire produzione di software sul mercato internazionale, dato che la maggior parte dei paesi dell'est Europa e delle software house indiane ha già adottato le metodologie moderne, occorre migliorare la qualità dei processi interni, introducendo le nuove competenze necessarie.

E' una questione strategica, necessaria se non si vuole perdere questa grande opportunità per il nostro sistema paese. E occorre anche che i nuovi programmatori siano adeguatamente preparati e che quelli esistenti siano adeguatamente aggiornati.



In alcuni capitoli precedenti, dopo il riconoscimento del ruolo fondamentale dell'IT nel XXI secolo, abbiamo trattato le figure professionali e la competenza che dovrebbe essere fornita dalla formazione accademica e nel capitolo precedente ci siamo occupati della fase di sviluppo e della figura professionale ad essa deputata, il developer.

La fase di sviluppo è importantissima per le qualità che il servizio IT realizzato avrà, **ma il servizio esprime il suo valore per gli utenti, adempiendo lo scopo per cui è stato creato, nella successiva fase di operations o di run, indicata in italiano anche come fase di produzione** (cui potremmo sottointendere produzione di valore) o di esercizio.

Questa è la fase principale della vita di un servizio IT o anche di un semplice strumento software. Questa è la fase in cui gli utenti usano il servizio o il software, traendone vantaggi. Ad esempio, in questa fase gli utenti interagiscono con un sistema di posta elettronica usando le caselle, con un sistema di geolocalizzazione e mappe virtuali, o con un software come la videoscrittura, traendone vantaggio per il proprio lavoro e la propria vita.

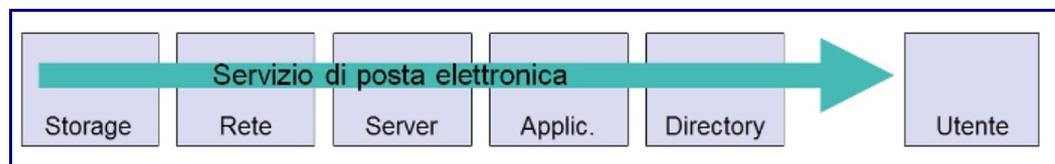
In molti casi, ingenuamente, si può pensare che questa fase avvenga in modo quasi automatico, ma non è così. Infatti, se consideriamo un PC o uno smartphone, l'uso di un software da parte dell'utente con quel dispositivo hardware presuppone che il dispositivo stesso sia stato configurato a dovere da parte di un **tecnico specialista** (o, in taluni casi, dall'utente stesso guidato da un sistema di aiuto scritto da un tecnico specialista) e può esservi



necessità, in caso di guasti o errori gravi compiuti da parte dell'utente, di un intervento tecnico e specializzato.

A maggior ragione questo è vero per i grandi servizi che oggi consideriamo indispensabili per la nostra vita, come ad esempio Google Maps, e per i grandi software aziendali, come ad esempio l'ERP (il software di gestione delle operazioni dell'azienda).

Infatti, i sistemi informatici che rendono possibile il funzionamento dei servizi IT sono composti di molte parti, sia hardware sia software. Ad esempio, un sistema di posta elettronica è formato da tanti elementi diversi, come indicato in figura:



- ✓ *Un sistema di memorizzazione permanente (storage) dove sono memorizzati tutti i file e le cartelle che formano le caselle di posta elettronica.*
- ✓ *Una infrastruttura di rete, che collega le postazioni di lavoro, gli smartphone ed i tablet con cui gli utenti accedono alle caselle di posta con i server e i server di posta fra loro.*
- ✓ *Un computer server, dove opera il programma applicazione di posta elettronica.*
- ✓ *Un'applicazione di posta elettronica, che opera gestendo lo storage ed i file e cartelle che formano le caselle, nonché gli accessi in lettura e le spedizioni e recapiti dei vari messaggi.*
- ✓ *Un sistema di directory (vale entro le aziende) che*



contiene la rubrica aziendale degli indirizzi.

✓ *Una postazione di lavoro (o smartphone o tablet), non indicata in figura.*

A sua volta, il server potrebbe essere virtuale ed operare entro un sistema di virtualizzazione con sottostante cluster hardware di server fisici, aggiungendo ulteriore complessità al sistema.

Tutti gli elementi, coordinati tra loro, contribuiscono al servizio “posta elettronica”. Un malfunzionamento in uno degli elementi può portare al blocco del servizio. L’utente vede il servizio come una unica entità, a cui accede attraverso una opportuna interfaccia utente (grafica, app o web nella maggior parte dei casi).

Il funzionamento ottimale dei sistemi informatici viene garantito da figure professionali dedicate, gli amministratori di sistema. Spesso essi sono coadiuvati da altre figure professionali, gli specialisti hardware e software.

La figura professionale dell’amministratore di sistema è in realtà suddivisa in vari profili specifici, definiti anche nella normativa UNI-11506 – UNI-11621-2:

- **System Administrator**, che opera normalmente su hardware, sistemi operativi come Windows o Linux, sistemi di gestione credenziali e diritti di accesso a risorse, applicativi specifici, svolgendo azioni di installazione, configurazione, monitoraggio e risoluzione di incidenti (ossia interruzioni non programmate del funzionamento normale) e di problemi (ossia le cause dei malfunzionamenti);
- **DBA (DataBase Administrator)**, che opera specificamente sui DBMS, ossia i sistemi di gestione dei dati, per ottimizzarne il funzionamento e garantirne le prestazioni ottimali rispetto agli specifici insiemi di dati che contengono, oltre che per risolvere problematiche ed errori che si possono verificare sui dati stessi durante le normali operazioni; spesso deve anche realizzare



strumenti di reportistica o integrazione fra basi dati diverse;

- **Network Administrator**, che opera sugli strumenti di rete, dai router agli switch, spesso gestisce anche i sistemi di cavi che collegano i computer tra loro, che stabilisce anche gli accordi con i fornitori di connettività dati/fonia, e di solito è anche responsabile dei firewall.

-

Nelle aziende molto piccole può esistere una unica figura, magari un professionista esterno, che fa tutto, mentre nelle aziende grandi e strutturate esistono team diversi che coprono ciascuno dei tre ruoli. Un compito trasversale, spesso affidato a più di una figura specializzata, è il backup o salvaguardia dei dati. E' un compito fondamentale: i dati informatici devono essere salvaguardati da incidenti software e hardware, da disastri come inondazioni o incendi, e da attacchi e sabotaggi deliberati, oltre che da errori involontari degli utenti.

Gli specialisti hardware e software, definiti nella UNI-11506 – UNI11621-2 dal ruolo Technical Specialist, **sono tecnici altamente specializzati su un prodotto hardware** (un tipo di server, un router, un dispositivo...) **o software** (un sistema di virtualizzazione, un ERP...), **quasi sempre appartenenti a ditte partner del produttore dell'hardware o del software**, che hanno seguito percorsi di formazione specifici e che si occupano sia della installazione iniziale sia della risoluzione di problematiche specifiche come i cambi di versione.

Quindi possiamo affermare che **tutti i sistemi IT che rendono possibili i servizi IT funzionano grazie a team di lavoro composti da queste figure professionali.** Le competenze dei singoli individui sono sicuramente importanti per il buon risultato, ma altrettanto lo è l'organizzazione dei team e, in particolare, l'uso di buone pratiche per la gestione delle attività, come quelle definite in ITIL e in ISO20000, framework e standard internazionale per la gestione

dei servizi IT (IT Service Management) [1].

Pertanto la qualità di un servizio IT dipende, quasi in egual misura, sia dalla qualità dei componenti software e hardware che lo formano, e quindi dalla qualità professionale degli sviluppatori che hanno scritto il codice e dalle metodologie di sviluppo ([come visto in questo articolo](#)), sia dalla qualità professionale degli amministratori e dalle metodologie da essi seguite.

Le figure sopra descritte hanno competenze e capacità che variano molto in funzione del ruolo e del contesto. In un ambiente piccolo/semplice possono essere sufficienti poche nozioni per svolgere al meglio il proprio lavoro. Nello specifico, quelle necessarie per governare i sistemi e i servizi IT affidati nelle operazioni “quotidiane”.

Per esempio una buona conoscenza di windows 10 può rendere un tecnico un ottimo amministratore di postazioni di lavoro (ossia del servizio IT “postazione di lavoro”). In contesti più grandi/complessi, come un data center, possono essere necessarie ottime conoscenze di più di un sistema operativo (es. Windows Server e Linux), di sistemi di virtualizzazione (es. VMWare ESX, KVM, Hyper-V...), di piattaforme e application server (es. JBoss, Docker...), di tanti DBMS (es. Oracle, MS SQL Server, IBM DB2...).

Vediamo più da vicino i percorsi con cui si raggiungono tali traguardi professionali...

Recenti rapporti di UnionCamere (si vedano [qui](#) e [qui](#)) dimostrano che in Italia gli specialisti IT (sia nell’area dell’amministrazione di sistema sia in quella dello sviluppo di codice) sono tra le figure più difficili da trovare.

L’amministratore di sistema junior (così come, del resto, lo sviluppatore junior) è molto spesso il primo ruolo con cui si entra



nel mondo del lavoro ICT. Un amministratore di sistema dovrà sempre tenersi aggiornato durante tutta la sua vita professionale, che potrà evolvere per diventare amministratore di sistemi complessi, o tecnico specialista, o system architect.

Il **System Architect** è una figura professionale (anche essa definita nella UNI -11506 . UNI-11621-2) che ha il compito di **progettare, integrare e realizzare soluzioni e servizi IT complessi da un punto di vista tecnico.** Deve anche garantire che le soluzioni tecniche, le procedure e i modelli di sviluppo siano aggiornati e conformi agli standard, oltre che agire spesso da team leader verso tecnici specialisti.

Un altro percorso conduce alla posizione di **CIO** (Chief Information Officer) o **IT Manager**, ovvero il **responsabile dei sistemi informativi** (spesso avente la qualifica di dirigente) di un'azienda o di una organizzazione [1].

La difficoltà di reperire tali figure non è una novità: anche in passato molte persone assorbite nel mondo IT e ICT avevano alle spalle percorsi di studi completamente diversi. **Quello che ora cambia è la estrema complessità delle tecnologie, che rendono difficile** (e soprattutto lungo, cosa molto pesante per le aziende che devono eventualmente finanziare un percorso di formazione e crescita professionale) **l'apprendimento "sul campo" per chi non ha basi scolastiche e/o accademiche nel mondo ICT.**

Molti posti di lavoro restano scoperti... mentre tante persone sono disoccupate, avendo seguito percorsi formativi con pochi sbocchi di mercato.

E spesso queste persone devono "riconvertirsi", frequentando corsi di formazione professionale "compatti" e limitati nel tempo, che rendono possibile acquisire le nozioni minime necessarie per un ingresso nel mondo del lavoro ICT, seppure a prezzo di enormi

sforzi per gli allievi.

Unico neo di questa situazione? La scarsa presenza femminile.

Nel mondo dello sviluppo software si varia da aziende in cui la presenza è inferiore al 10% ad aziende in cui si arriva al 40-45%, quindi quasi un rapporto di parità.

Nel mondo della amministrazione di sistema invece la presenza femminile è più bassa, tipicamente inferiore al 20%, soprattutto nel comparto della amministrazione di infrastrutture o di data center complessi, dove esistono molti gruppi di lavoro numerosi composti da soli uomini.

Questo è essenzialmente effetto di uno stereotipo culturale: i migliori sistemisti sono “nerd” e uomini. E, per la mia esperienza, completamente errato: ho conosciuto donne amministratore di sistema, tecnico specialista, system architect ed anche CIO, tutte estremamente competenti e valide nel loro ruolo.

Un'altra differenza che si osserva entro il mondo degli amministratori di sistema è quella tra coloro che per il loro ruolo (soprattutto gli amministratori di applicativi) hanno a che fare con gli utenti (e con la risoluzione dei loro problemi e necessità) e quelli che invece hanno poco a che fare con gli utenti (soprattutto gli amministratori di componenti infrastrutturali come i server) e “vedono” gli utenti attraverso gli amministratori di applicativi con cui interagiscono (e, spesso, si scontrano).

Chi opera a contatto con gli utenti deve sapersi mettere nei loro panni, stabilire la giusta comunicazione con loro.

Ma, in generale, il contatto fra il reparto ICT di un'azienda e gli utenti è qualcosa che va governato e gestito a livello aziendale,



come consigliano tutti i sistemi di buone pratiche.

Altrimenti il rischio di incomprensioni e di nascita di fratture, con l'arroccamento del reparto ICT sulla difensiva, è molto alto. In generale tutto questo va gestito all'interno del ciclo di evoluzione dei servizi IT, di cui tratteremo nel prossimo capitolo.



Sviluppo versus funzionamento operativo di servizi IT: coordinare esigenze in apparenza contrapposte.

Abbiamo già trattato due figure professionali fondamentali dell'IT: lo sviluppatore di software (o developer) e l'amministratore di sistema (o sysadmin). Due professionalità molto importanti che in Italia sono spesso anche il ruolo di ingresso nel lavoro, talora punto di inizio di carriere professionali che conducono a posizioni come responsabili commerciali, manager, imprenditori.

In questo capitolo esaminiamo invece i processi di cui queste figure sono protagonisti: **lo sviluppo del prodotto software e/o del servizio IT** cui esso da origine e la fase di esercizio di quel software ovvero del servizio IT su di esso basato.

Apparentemente essi hanno caratteristiche ed esigenze di ottimizzazione opposte e che potrebbero venire in contrapposizione.

Lo sviluppo del software, in special modo quello del software custom, cioè costruito ad hoc per le esigenze specifiche di un cliente è un progetto, è definito, **seguendo la terminologia "canonica" del Project Management Institute**: *"uno sforzo temporaneo intrapreso allo scopo di creare un prodotto, un servizio o un risultato unici"*, dove temporaneo significa limitato nel tempo.

Il progetto software ha un inizio, che di solito parte con l'analisi, **e una fine con un suo esito**: il prodotto software viene realizzato e gli errori corretti (in gergo *debuggato*) o almeno con la maggior parte di essi corretti e consegnato al cliente o rilasciato sul



mercato. Il modello “canonico”, tutt’ora molto usato, è il modello a cascata o waterfall (si veda [1]). La manutenzione ordinaria ed evolutiva (quindi, con l’aggiunta di nuove caratteristiche) di software come, ad esempio, MS Word ha portato ai cicli di sviluppo, un primo passo verso il passaggio da progetto (attività una tantum) a processo (attività ripetitiva).

Ma, quasi sempre, fino ad anni recenti, il focus manageriale in relazione allo sviluppo software è stato incentrato sul completare la fase di sviluppo e rilasciare il software o la versione X del software, caratterizzata da un certo insieme di caratteristiche funzionali, nel minor tempo possibile, con la migliore qualità possibile (caratteristica essenzialmente letta come “con il minor numero di errori possibile”) e al minor costo possibile. Non sempre queste tre esigenze sono compatibili fra loro. Il fatto che poi il software entra in uso in azienda o presso privati e diventa il componente fondamentale di un servizio IT, che deve produrre un valore per il cliente con il proprio funzionamento non era adeguatamente considerato.

La fase di esercizio di un software a formare un servizio IT è un processo, ossia *“un insieme di attività coordinate tra loro atte a produrre un valore per i clienti/utenti del processo stesso e ripetute nel tempo”*. Sono quindi attività che si ripetono, in modo più o meno ciclico. L’ottimizzazione di tali attività ha come obiettivo produrre un servizio misurabile con **parametri opportuni**, come per esempio la velocità di risposta di un programma gestionale a un’interrogazione dell’archivio clienti, il cui valore si mantiene costante nel tempo con un costo per unità di tempo circa costante nel tempo. Quindi le caratteristiche più apprezzabili sono stabilità e duratività.

In alcuni casi questo ha portato all’estrema conservatività: le persone incaricate dell’esercizio dei sistemi tendevano a voler conservare l’esistente nel corso degli anni, opponendosi addirittura



ai cambiamenti nei sistemi stessi, in maniera più o meno evidente.

L'evoluzione temporale dei sistemi informativi e dei loro servizi IT

All'interno delle aziende o delle organizzazioni come la pubblica amministrazione i **servizi IT sono un supporto per le attività di business**. L'arroccarsi sulla conservazione dell'esistente ("ciò che funziona non si cambia") si scontra quindi con due esigenze che sono diventate nel tempo sempre più importanti:

- **Disporre di soluzioni software conformi alle esigenze di business dell'azienda;** all'inizio dell'introduzione dell'IT in azienda (in Italia, tra gli anni '60 e gli anni '80) tali esigenze erano stabili o variavano poco nel tempo, consentendo quindi pause di stabilità abbastanza lunghe fra un aggiornamento software ed un altro; oggi, in un mercato *instabile*, l'azienda deve cogliere *tutte le opportunità possibili* ed *l'IT deve seguire e supportare* queste esigenze di business;
- **Ridurre i costi di esercizio:** tecnologie proprietarie di aziende vendor, che sino a non molto tempo fa dominavano il mercato oggi sono molto meno frequenti per gli alti costi di esercizio pagati sotto forma di licenze d'uso e canoni di manutenzione di software ed hardware.

A partire da fine anni '90 in poi questo ha prodotto, nelle piccole e medie imprese, il passaggio ad architetture più aperte (spesso basate sui PC), con costi di acquisto e manutenzione molto minori e la scoperta successiva di un costo non considerato, legato alla minore qualità di esercizio: il **tempo di fermo macchina**, in cui l'azienda subisce il danno della impossibilità, da parte degli operatori, di usare le risorse IT nel proprio lavoro.

Negli anni, trend di mercato tesi a riportare indietro le cose a



soluzioni più “centralizzate” e in apparenza più controllabili, si sono alternati a ondate di decentralizzazione, spesso in entrambi i casi pilotati abilmente dagli uffici marketing delle grandi aziende di informatica interessati a vendere. Questo ha prodotto, in molte aziende, soprattutto in Italia, una sfiducia generale nell’IT, vista come *“un male necessario”* e non uno strumento di innovazione. In molti casi, tale visione ha scavato un solco sempre più profondo fra le posizioni degli sviluppatori o dei responsabili degli applicativi da un lato e quelle dei sistemisti dall’altro. Questo, sia nel caso in cui gli sviluppatori siano esterni all’azienda che usa il software sia, spesso a maggior ragione, nel caso in cui siano interni. Estremizzando, secondo gli sviluppatori i sistemisti sono spesso *“conservatori sino all’eccesso, si oppongono a tutte le novità, hanno bisogno che gli si scriva tutte le informazioni relative ad un software perché non sono capaci di arrivarci da soli e sono pigri”*. Dall’altra parte, secondo i sistemisti, gli sviluppatori *“lavorano male, non fanno adeguate verifiche, non ci danno le informazioni necessarie per installare e gestire i software che, quasi sempre, sono pieni di errori e dobbiamo essere noi a correggere e fare funzionare”*.

Queste posizioni, in apparenza inconciliabili, stanno venendo superate, come è necessario che sia. In diverse mie attività di consulenza e coaching aziendale ho operato proprio in questo senso: la prima azione necessaria è stato applicare le tecniche di coaching per fare superare la diffidenza reciproca ai reparti.

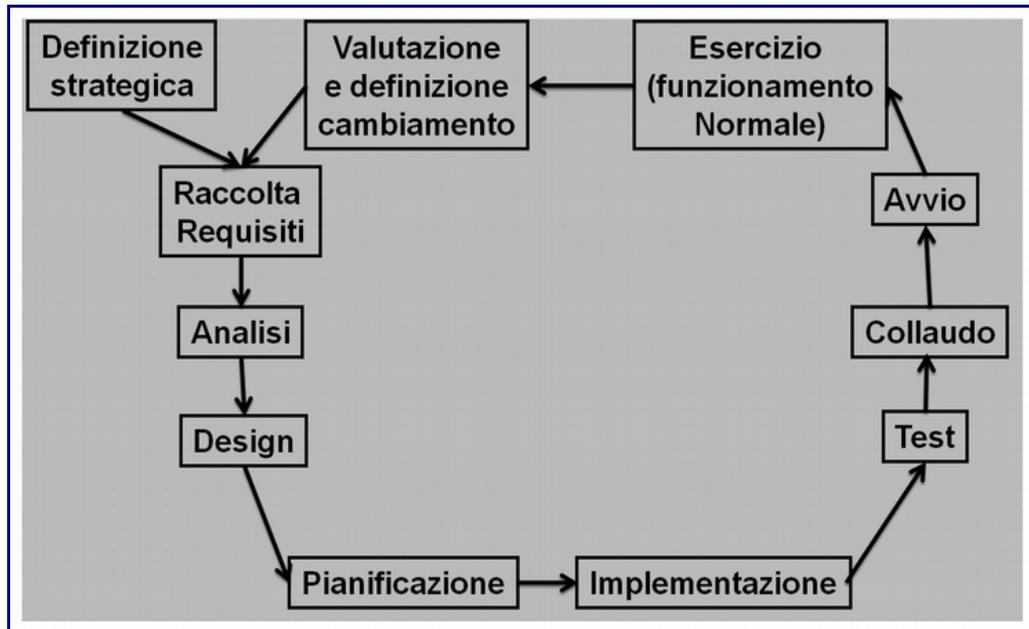
La soluzione: gestire tutto il ciclo di vita

Per superare le problematiche il primo stadio è stato l’avvento di modelli di gestione riguardanti tutta la fase di vita di un prodotto



software, dal momento in cui questo viene concepito, come evoluzione di sistemi precedenti o ex-novo, al momento in cui entra in esercizio e per tutta la durata dell'esercizio, sino a versioni successive o alla sua dismissione (si veda anche [1]).

Un esempio di ciclo è rappresentato nella figura sottostante:



Il modello forse più diffuso è quello del ciclo di vita del servizio del framework ITIL, ripreso poi nello standard ISO20000 e descritto in [questo articolo](#).

Essenzialmente un ciclo di questo tipo si può suddividere in 5 fasi:

- **Azione strategica (in ITIL: Service Strategy)**, in cui viene deciso, in funzione delle esigenze di business se un servizio o un software deve essere realizzato o evoluto e quali obiettivi esso dovrà raggiungere, oltre che quale budget dedicare al progetto di realizzazione e/o evoluzione; corrisponde ai blocchi di definizione strategica del ciclo in figura;
- **Azione progettuale (in ITIL: Service Design)**, in cui vengono raccolti i requisiti di dettaglio, compiuta un'analisi funzionale completa e svolta la progettazione, pianificato il lavoro di



realizzazione e svolta l'implementazione (blocchi raccolta requisiti, analisi, design, pianificazione e implementazione in figura);

- **Azione di installazione (in ITIL: Service Transition)**, in cui il prodotto realizzato viene verificato, collaudato e installato (blocchi test, collaudo, avvio);
- **Azione di esercizio (in ITIL: Service Operation)**, in cui il servizio IT basato sul prodotto realizzato funziona normalmente ed eroga il valore per cui è stato creato al business;
- **Azione di monitoraggio e valutazione necessità di cambiamento (in ITIL: Continual Service Improvement)**, in cui periodicamente si valuta la rispondenza del servizio ai bisogni presenti del business e le eventuali necessità di modifica e/o evoluzione.

L'insieme dei prodotti (deliverable) di ogni fase diventa poi l'ingresso della fase successiva.

Da un punto di vista manageriale, seguire il modello significa sia gestire perfettamente ogni fase (garantendo che il prodotto relativo sia il migliore possibile) che, soprattutto, gestire il coordinamento del passaggio di consegne tra le fasi.

Tipicamente, infatti, **le persone coinvolte operativamente nel ciclo sono impegnate solo in una delle diverse fasi**, mentre **tra i prodotti di ogni fase deve essere realizzata e condivisa la documentazione necessaria alla trasmissione di ogni informazione necessaria**. Questo rende indispensabile la presenza di un manager che sovrintende a tutto il ciclo.

Il problema legato alla implementazione pratica e operativa di tutto ciò è la scarsa percezione del valore delle documentazioni: se non c'è il passaggio delle informazioni necessarie tra le fasi tutti i problemi sopra descritti riemergono con forza.



L'evoluzione: l'approccio Agile e il DevOps

L'evoluzione progressiva ha portato prima all'introduzione delle metodologie Agili (si veda [2]) e in particolare del metodo SCRUM, che segmenta un macro progetto in tante fasi di durata limitata, ciascuna delle quali deve produrre un risultato significativo per il business, con un rischio basso.

E poi all'adozione dei processi di Continuous Development [2], ossia un insieme di tecniche per permettere lo sviluppo iterativo continuo delle applicazioni che formano i servizi IT.

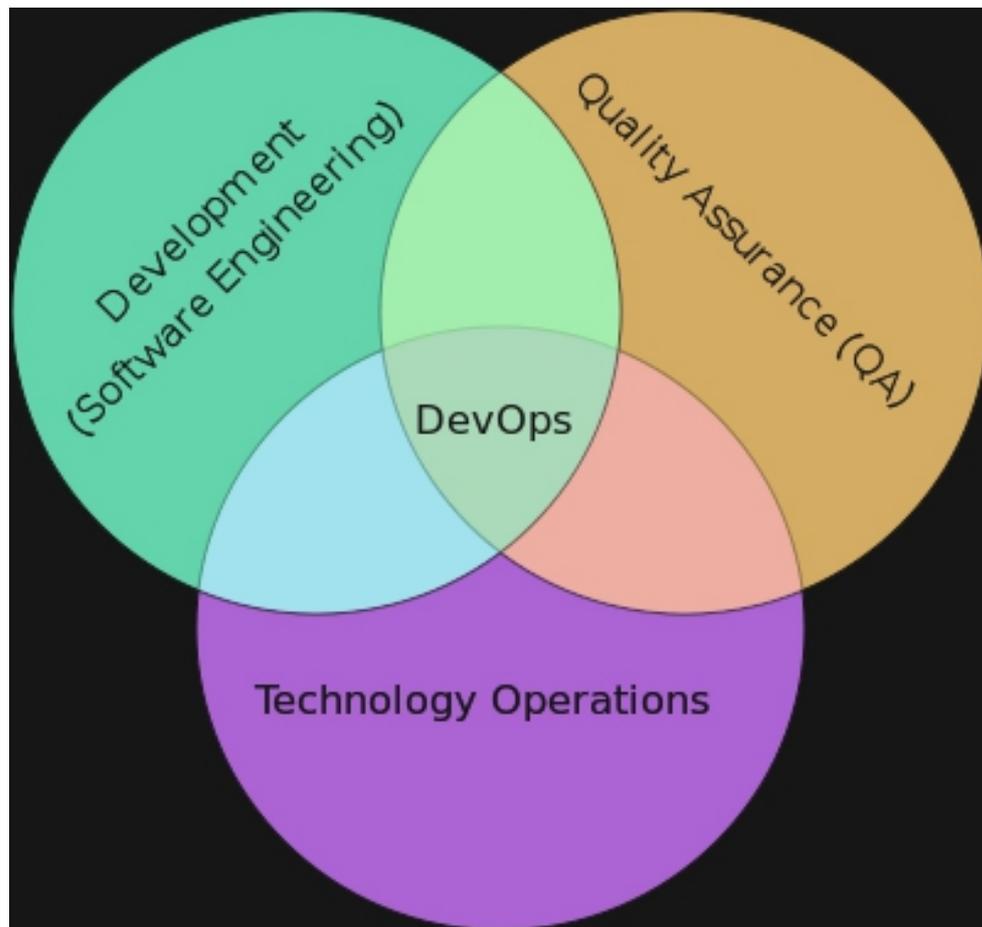
Le fasi di ogni iterazione *sviluppo-collauda-rilascio* sono ben codificate e coordinate in modo da ridurre la necessità di passaggio di informazioni al minimo indispensabile. Ove possibile, inoltre, le azioni ripetitive come le installazioni vengono automatizzate mediante ampio uso di strumenti software appositi, governati con opportuni linguaggi di vera e propria programmazione.

L'evoluzione ulteriore ha condotto al DevOps (Development and Operations), una metodologia di implementazione pratica della gestione del ciclo di vita, in particolare del passaggio dallo sviluppo alla messa in produzione. DevOps enfatizza al massimo la collaborazione, la comunicazione e l'integrazione tra sviluppatori software e professionisti IT (sysadmins, DBA, etc.).

Lo scopo di questa metodologia è mettere un'organizzazione in grado di fornire prodotti e servizi software in tempi rapidi, evitando i "conflitti" descritti sopra.

DevOps, non a caso, è nato nel contesto di aziende come Twitter ed Amazon, dove cioè il servizio IT e le sue componenti software sono parte fondamentale del business e dove il business stesso evolve rapidamente per seguire le necessità commerciali, così che il software deve essere in grado di stare al passo.





Questo significa che gli sviluppi e i rilasci in produzione (con sostituzioni di componenti precedenti) diventano una necessità globale, magari con più cambiamenti in una stessa giornata di lavoro. Come mostrato nella figura, infatti, **DevOps prevede l'integrazione globale fra i tre pilastri dei sistemi informativi rappresentati in figura: sviluppo, qualità (test, verifiche, collaudo) e operations.** Quindi coordinamento tra i reparti è la soluzione. Arrivare a questo obiettivo prevede un grande cambiamento dentro le aziende, possibile con vere e proprie azioni di coaching per i team.

Per chiudere un commento personale: ho vissuto sul campo la prova pratica di tutto ciò anche in ambito universitario. Alcuni anni fa ho infatti partecipato come relatore di tesi al progetto svolto sul tema di un brillante studente lavoratore che è poi diventato



responsabile sviluppo presso l'importante portale Web di servizi dove già lavorava.

L'introduzione del DevOps in questo caso ha portato alla riduzione del 40% della durata di ogni ciclo di sviluppo, con contemporanea riduzione a meno della metà degli errori riscontrati nei software prodotti a formare un servizio IT fondamentale per l'azione di tutto il portale.



Total cost of ownership: spese di costruzione e di esercizio di un sistema IT

Nei capitoli precedenti abbiamo trattato due figure professionali fondamentali dell'IT: lo sviluppatore di software (o developer) e l'amministratore di sistema (o sysadmin) e il problema generale della evoluzione dei servizi software. Il DevOps e la gestione del ciclo di vita dell'intero servizio hanno migliorato moltissimo le cose, riducendo i problemi e migliorando le performance.

Nel corso del ciclo di vita – dal momento della sua ideazione, passando per la sua progettazione, implementazione ed entrata in esercizio, sino al momento del suo ritiro definitivo – **un servizio IT subisce moltissime trasformazioni**. Nel panorama di oggi un servizio IT deve cambiare sia dal punto di vista tecnologico, seguendo l'evoluzione della tecnologia, sia dal punto di vista funzionale, aggiungendo via via nuove funzioni e capacità in base alle esigenze del business. Un servizio IT, dunque, non è qualcosa di statico che opera sempre allo stesso modo costante nel tempo, ma è qualcosa che si evolve in base alle esigenze del business e alla tecnologia. Il **servizio IT alternerà, nel corso degli anni, momenti in cui il suo funzionamento rimane costante a momenti in cui si evolve e si trasforma**.

In passato, la necessità di fare evolvere continuamente un servizio è stata affrontata con la distribuzione, magari tramite



CD o dischetti spediti via posta fisica, di aggiornamenti (patch) software, destinati ad essere applicati ai servizi seguendo opportune procedure. **Poi gli aggiornamenti sono stati scaricati da Internet**, come avviene, ad esempio, con gli aggiornamenti periodici dei browser. **Per quanto riguarda i grandi servizi**, come i portali Web, **l'aggiornamento diventa qualcosa di continuo**, con la necessità da un lato di evolvere le funzioni e dall'altro di seguire la tecnologia.

In questo capitolo **affronteremo il tema dei servizi IT e della loro evoluzione da due punti di vista collegati: quello architetturale e quello dei costi.**

Accanto al costo di progettazione e realizzazione di un servizio, ovvero il costo complessivo del progetto iniziale che porta a costruire e rendere disponibile per gli utenti il servizio, c'è, infatti, anche il costo di mantenimento, suddiviso fra le spese necessarie per il funzionamento quotidiano e le spese per i progetti di trasformazione che, per i motivi suddetti, il servizio IT dovrà subire, per rimanere allineato nel tempo con gli obiettivi di business. **In termini economici, quindi, occorre – in un progetto – considerare il “[Total Cost of Ownership](#)”, ossia il costo totale di possesso, somma di tutti i costi di realizzazione e mantenimento in esercizio, all'interno del ciclo di vita di un servizio IT.**

Progettare per l'evoluzione: controllo dei costi

Partiamo con un esempio (*ogni riferimento a fatti realmente accaduti è...*). Supponiamo di avere realizzato un servizio IT fruibile via Web, ad esempio un portale di e-commerce, e che il servizio abbia successo ed il suo uso si espanda. Ecco i fattori di carico che aumentano:

- **Il numero totale degli utenti del servizio cresce:** statisticamente, anche gli utenti collegati in un dato momento aumentano.



- Il numero dei dati contenuti nel servizio cresce progressivamente.
- Le singole operazioni svolte dagli utenti possono diventare più pesanti, sia perché vi sono più dati e più utenti collegati, sia perché, in una evoluzione funzionale, potrebbe aumentare il peso stesso delle singole operazioni, arricchite di nuovi componenti.

Risultato: le prestazioni rallentano e, di fronte al rischio di diventare inusabile per eccessiva lentezza, i gestori del servizio decidono di aumentare la potenza hardware sottostante (“il ferro – ossia l’hardware – costa poco...”). 10 anni fa questo significava acquistare nuovi server fisici e porli nel data center facendo funzionare anche su di essi l’applicazione, oppure sostituire direttamente l’insieme dei server con altri più potenti.

Ed ecco alcune domande in proposito:

- L’applicazione è veramente progettata per sfruttare al meglio una architettura hardware in cui vi sono più server?
- Di conseguenza, quanta potenza computazionale non stiamo usando al meglio?
- Le componenti di software proprietari presenti, ad esempio il DBMS, come aumentano le licenze al crescere della potenza hardware sottostante?

Poteva certo verificarsi il caso in cui occorre aggiungere molti server o molte CPU per ottenere nuovamente prestazioni accettabili. E questo faceva crescere di molto i costi delle licenze annuali del software proprietario, con conseguente aumento enorme del costo di esercizio della struttura.

“Poco male” – si potrebbe dire – “È il passato. Oggi, con i server virtuali ed il cloud...”

Bene: oggi, con i server virtuali diventa possibile aggiungere dinamicamente CPU e/o RAM e spazio disco ad un server



virtuale, se e solo se queste risorse hardware sono presenti nell'insieme di macchine fisiche su cui le macchine virtuali funzionano entro un data center aziendale.

Quindi, se non si sono tali risorse, sarà necessario nuovamente procedere all'acquisto. E la licenza del software proprietario segue la stessa politica, per cui aumentare il numero di CPU anche virtuali significa aumentare i costi delle licenze.

“Allora spostiamoci sul cloud, dove le risorse sono illimitate” – in molti penseranno a questo punto. E in parte potrebbe essere così: il cloud di grandi operatori di mercato, come Amazon, Microsoft Azure, Google, ecc... è basato su grandi data center e teoricamente illimitato come risorse.

Ma quando acquistiamo queste risorse, le paghiamo. Ogni CPU a nostra disposizione sul cloud viene pagata, lo spazio disco in più viene pagato. Software presenti come servizi nel cloud hanno un costo di licenza.

In alcuni tipi di contratto il pagamento viene fatto proprio in base ai cicli di funzionamento delle CPU virtuali del cloud usate dai servizi che abbiamo posto nel cloud stesso, calcolate dall'operatore come se fossero minuti telefonici. Quindi le risorse usate si pagano. E lo stesso vale per le licenze di software proprietario, quotate quasi sempre in modo proporzionale al numero di CPU coinvolte nel servizio basato su tale software proprietario.

Chiaramente, se il servizio è a pagamento, produce un valore per i clienti e questi pagano, il fatto che i clienti aumentano dovrebbe significare maggiori incassi e quindi coprire ampiamente le maggiori spese.

Ma è veramente così? Sicuramente, per servizi a pagamento, gli incassi aumentano. Ma per servizi interni ad una azienda e fruiti da dipendenti e collaboratori dell'azienda stessa? E,



soprattutto, **l'aumento delle richieste di risorse** (e quindi i corrispondenti costi di esercizio), **cresce in modo proporzionale all'aumento della qualità dei servizi erogati** (e quindi ai corrispondenti benefici ed eventuali incassi)?

Progettare per l'evoluzione: architetture ottimizzate e scalabili

Ricapitolando quanto appena visto, il risultato è quindi che il servizio IT richiede alla propria infrastruttura IT che ne consente il funzionamento:

- **Più CPU**, ovvero più potenza computazionale, per poter garantire al singolo utente tempi accettabili per la elaborazione dei propri dati e poterli trattare in mezzo ad una mole di dati crescenti;
- **Più memoria RAM** per elaborare i dati in memoria;
- **Più capacità di storage** per memorizzare i dati che crescono.

Questo aumento dipende dal tipo di soluzione software: se l'applicazione è costruita bene vengono minimizzate le necessità di aumento , che però sono fisiologiche e non sono eliminabili.

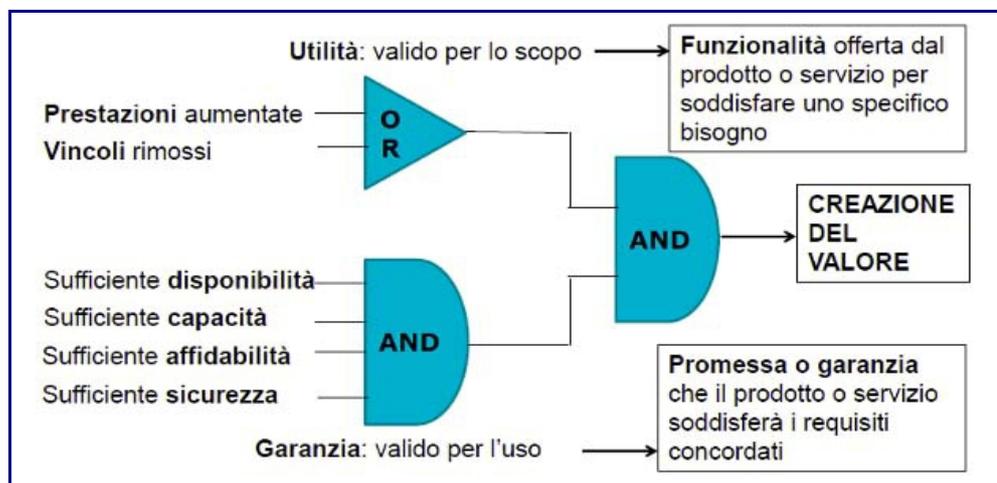
Ma c'è una bella differenza fra dover raddoppiare le risorse hardware/cloud se si raddoppiano, ad esempio, il numero di utenti o la quantità di dati e doverle quadruplicare, o più, perché il software non è stato progettato in modo da essere scalabile ed occorre trovare soluzioni.

Cosa significa questo in pratica? Che occorre cambiare il



modo di progettare le soluzioni IT. Se non sono veramente molto specifiche per risolvere un problema ben delimitato, dovrebbero essere pensate sin dall'inizio per poter evolversi:

- **In senso funzionale**, con l'arricchimento delle funzionalità che offrono.
- **Rispetto alla richiesta**, con l'aumento del numero di utenti.
- **Rispetto ai dati** in esse contenuti, al crescere dei dati stessi.
- **Rispetto alla disponibilità temporale**, qualora necessario che siano disponibili anche in ore inizialmente previste (a questo proposito: accettereste, per esempio, un sistema di home banking che non è disponibile dalle 23 alle 6 del mattino?).



Gli standard e i framework di buone pratiche dicono proprio questo.

Ad esempio, il **framework ITIL**, il più diffuso nell'ambito della gestione dei servizi IT, come schematizzato in figura, **afferma** che per ogni servizio IT il valore è dato sicuramente dall'aspetto funzionale, ma anche dalle caratteristiche di:

- **disponibilità (availability)**: il servizio deve essere disponibile solo in ore ufficio o 24x7x365 (24 ore per tutti i giorni della settimana per tutti i giorni dell'anno)?



- **Capacità (Capacity):** il servizio ha sufficienti capacità per garantire il suo uso accettabile? E' stato previsto un meccanismo di ampliamento delle capacità se necessario?
- **Affidabilità/continuità (continuity):** il servizio è robusto ed affidabile? In caso di guasto ci sono tutte le risorse tecniche ed organizzative necessarie per ripristinare tempestivamente le sue funzionalità?
- **Sicurezza (security):** il servizio è sicuro? I dati contenuti in esso sono mantenuti come:
 - **Riservati:** accesso garantito soltanto alle persone autorizzate e negato a chiunque altro.
 - **Disponibili:** accessibili tutte le volte che è necessario.
 - **Integri:** i dati non si corrompono e non subiscono modifiche non autorizzate.

E queste caratteristiche non devono essere aggiunte in seguito, ma essere presenti fin dalla fase di progettazione. Approccio pienamente conforme anche agli [articoli 25 e 32 del GDPR](#).

La progettazione tecnica quindi deve seguire queste linee guida. Una volta definito al meglio l'insieme dei requisiti funzionali che esprimono il valore di business che il servizio IT dovrà erogare quando entra in esercizio (compito che è in capo al [Business Analyst](#)), è necessario tradurre questi in una architettura tecnica che rispetti i principi sopra descritti.

E che quindi possa evolvere nel tempo seguendo le esigenze del business senza nel contempo avere l'esplosione dei costi dovuti a licenze o a massicci potenziamenti dell'hardware. Per molto tempo si è tralasciato questo aspetto, contando sul fatto che l'hardware costa poco... E in parte è davvero così, ma quello che esce dalla porta, entra poi dalla finestra, visti i costi dei servizi cloud e delle licenze software.

Bisogna quindi sviluppare una filosofia della gestione della



Capacità (Capacity Management) dei servizi IT, rendendoli il più possibile adattabili; inoltre, dove sono disponibili i dati relativi, occorre prevedere le curve di crescita della richiesta di capacità, soprattutto per quanto riguarda l'area di storage (memorizzazione di massa).

Un disco da un terabyte costa poco, oggi, ma – se vi salviamo dentro molti dati multimediali – fa anche presto a riempirsi. E, in un sistema di **storage aziendale RAID**, ove i dati sono protetti da incidenti grazie alla ridondanza multipla, il costo di un terabyte di spazio è decisamente superiore. Tutto ciò richiede una professionalità più elevata nei progettisti, e più tipicamente le competenze della figura del System Architect, di cui abbiamo già parlato verso la fine di **questo articolo**.

E' davvero necessario? Per servizi IT “piccoli”, e in cui non è prevista una crescita grande, direi di no, ma per servizi di maggiori dimensioni, in contesti ampi (cioè per le grandi aziende, o per l'esposizione su Internet), certamente sì, se si vuole risparmiare sui costi di mantenimento.

Purtroppo passare a questo tipo di visione, almeno in Italia, è un vero e proprio cambiamento di paradigma. Il problema della gestione della fase di esercizio e dei suoi costi non è infatti solo nell'IT, ma anche in altri settori strategici per il paese, come purtroppo hanno dimostrato recenti avvenimenti.

Conclusioni

Nel nostro White Paper **siamo partiti innanzitutto dalla visione del futuro legata alle recenti trasformazioni** rese possibili dalle nuove tecnologie e paradigmi organizzativi.

Le aziende infatti, anche manifatturiere, **stanno progressivamente passando da una vendita di prodotti ad una vendita di servizi**, di cui i prodotti sono la parte tangibile. Le tecnologie stesse stanno nei fatti creando vere e proprie “rivoluzioni” e cambi di paradigma, come accade ad esempio per la prevenzione medica di massa resa possibile da dispositivi che monitorano le nostre funzioni vitali.

Realizzare gli strumenti che rendono possibili queste trasformazioni diventa sempre più complesso nel tempo e richiede professionalità di livello per condurre al successo.

In primo luogo, dato un problema da risolvere (ovvero una spesa da ridurre), o data una opportunità da cogliere (ad esempio un ricavo da aumentare), occorre qualcuno che sappia definire con precisione le premesse e la soluzione che soddisfa l'esigenza. Ed ecco la figura professionale del **Business Analyst come primo attore** di un progetto teso a realizzare tale soluzione. **Il team di progetto** è poi formato da figure con professionalità diverse.

Business Analyst e team devono poter comunicare in modo preciso, efficace ed efficiente. **Ecco quindi le capacità di comunicazione tecnica efficace, che in molti ruoli diventano una necessità per i professionisti.**



Riprendendo gli “hard skill” tecnici, ecco il compito fondamentale: spiegare a un automa “stupido e veloce” come il computer cosa deve fare per produrre valore. Ossia scrivere il codice sorgente di una soluzione IT, che darà origine ad un servizio IT teso a produrre valore per chi lo utilizza entro un determinato contesto. Ed ecco la figura del **developer**.

Ma scrivere il codice non basta: una volta divenuto un insieme di componenti IT che formano un servizio, un software deve essere mantenuto per funzionare correttamente. E deve essere mantenuto allineato con le esigenze del business che potenzialmente cambiano con una velocità impressionante. Ed ecco la figura dell'**amministratore di sistema**, con le sue varie declinazioni.

Oltre a queste altre figure devono svolgere il loro ruolo per garantire sia una fase di costruzione/sviluppo di una soluzione, sia una fase successiva di esercizio e manutenzione. E questo comporta coordinamento, capacità di progettazione tecnica, di gestione del progetto di realizzazione e di gestione della successiva fase di esercizio, in cui la soluzione fornisce il valore per cui è stata progettata e costruita.

E il computo economico deve tenere conto delle spese durante tutto il ciclo di vita: dalla costruzione al mantenimento ed al ritiro finale, con un eventuale trasporto dei dati in un nuovo sistema. Così come avviene già in altri settori del comparto industriale.

Quindi, riassumendo, possiamo affermare che:

1. La tecnologia e la connessa organizzazione stanno producendo la nuova **rivoluzione digitale**, che sta trasformando il nostro mondo, la nostra società e il nostro modo di vivere;
2. **occorre governare questa trasformazione**, riducendo gli impatti negativi e



potenziando invece quelli positivi;

3. per poterlo fare **occorrono professionalità preparate**, in vari ruoli;
4. **i ruoli devono saper comunicare efficacemente tra loro**;
5. **soft skill e hard skill sono egualmente importanti** per raggiungere il risultato finale, e **alcuni compiti sono particolarmente importanti** per il risultato finale;
6. **l'analisi e la progettazione delle soluzioni devono tenere conto dell'aspetto economico nel presente e nel futuro**; ossia nella progettazione e realizzazione così come nella successiva fase di esercizio di un servizio IT;

Ed ecco che si torna a una necessità stringente, strategica e contingente, di competenza e di uso delle buone pratiche per ottenere il successo nella rivoluzione digitale.

Giulio Destri

Approfondimenti

- Robert Dilts – *Creare Modelli con la PNL* – Ed. Astrolabio, 2003
 - Shelle Rose Charvet – *Le parole della Mente: eccellere nel linguaggio di Influenza* – Ed. FrancoAngeli, 2013
 - Oren Klaff – *Pitch Anything: la presentazione perfetta* – Ed. ROI Edizioni, 2017
 - Daniel Goleman – *Lavorare con Intelligenza Emotiva* – Ed. BUR, 1998
 - Paolo Borzacchiello – *PNL per la vendita* – Ed. Alessio Roberti Editore, 2015
 - Giulio Destri – *Manuale di PNL per la pratica nella Vita Professionale* – Ed. Amazon, 2016
 - Giulio Destri “Sistemi Informativi il pilastro digitale di servizi e organizzazioni” Ed. FrancoAngeli, 2013
 - G. Destri e C. Lombardi [“I processi di Sviluppo Software: la storia”](#)
 - C. Lombardi e G. Destri [“I processi di Sviluppo Software: l'evoluzione agile e il DevOps”](#)
 - CREDITS
- Immagine di copertina (rielaborata): ID 70705464 di [nicoelnino](#)

About

MAPS GROUP

Dai *Big Data* ai *Relevant Data*, il gruppo sviluppa sistemi *software* che creano conoscenza a supporto dei processi decisionali. I prodotti Maps Group strutturano il patrimonio di informazioni di aziende private e Pubbliche Amministrazioni in *Data Warehouse*, gestionali ed analitici, che si pongono come strumenti di *governance* e di *business*.

6MEMES Quando si parla di Dati, l'attenzione si sposta su questioni numeriche o al limite statistiche, ma sotto a quest'aligida apparenza la realtà è un'altra. Il blog 6Memes, dedicato all'opera *Six Memos for the Next Millennium* di Italo Calvino, vuole mettere a nudo le potenzialità dei Dati, traducendoli nei linguaggi dell'Uomo: Cultura, Natura, Economia, Arte e, perché no, Ironia.

GIULIO DESTRI

Giulio Destri è ingegnere elettronico e Ph.D. in ingegneria informatica. Opera come Business Advisor nel settore ICT e dei sistemi informativi e tecnologici interni ad aziende e pubbliche amministrazioni. Dal 2003 è professore a contratto di Sistemi Informativi presso l'Università di Parma, per la quale ha scritto anche il libro di testo 'Sistemi informativi. Il pilastro digitale di servizi e organizzazioni'. Dal 2008 ha iniziato a svolgere attività di mentoring e business coaching. È certificato Oracle, ITIL, COBIT, SCRUM Master, NLP Coach con specializzazione in Business e Team Coaching, ed esaminatore UNI11506-UNI11621. Appassionato di arti marziali e trekking, scrive articoli, racconti e poesie e gli piace fare fotografie di viaggi e paesaggi.



memes

MAPS GROUP
www.mapsgroup.it