

Collaborazione nei progetti software

Quando cliente e fornitore collaborano per il successo dei progetti ...

Ercole Colonese, Consulente di Direzione e di Information Technology

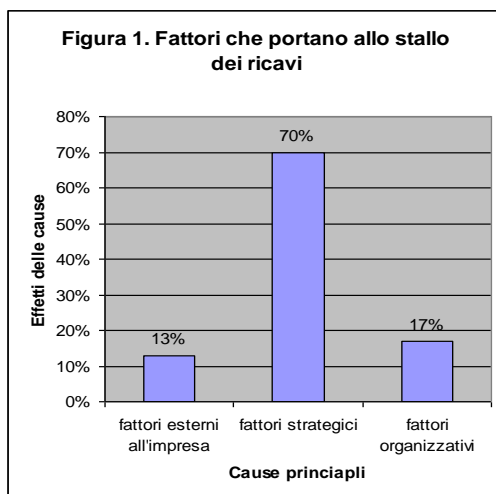
Articolo pubblicato sul numero 3-2009 della rivista *Qualità on-line di Aicq*.

Roma, gennaio 2009

La crisi economica attuale accentua la necessità di realizzare progetti innovativi in tempi brevi, a costi competitivi e con qualità adeguata. Tale obiettivo sembra essere difficile da raggiungere con le competenze attuali e gli approcci seguiti. Studi sul fallimento di moltissimi progetti lo confermano. Lo scenario italiano attuale non è confortante. Cosa fare allora? L'approccio suggerito è quello di trasformare l'attuale rapporto cliente-fornitore, spesso conflittuale ed in contrapposizione, in uno più evoluto in cui le due parti condividono i rischi ed il successo. L'evoluzione verso un rapporto di "partnership" richiede una pari trasformazione culturale che veda le due parti – pur in ruoli distinti e con responsabilità separate - collaborare per il raggiungimento di un comune obiettivo: il successo dei progetti strategici. Nell'articolo si forniscono alcuni suggerimenti pratici che facilitano la collaborazione nelle diverse fasi del ciclo di vita del software in base ai rischi di progetto tipici delle singole fasi.

L'attuale crisi economica si ripercuote negativamente sui mercati mettendo in seria difficoltà le imprese e, talvolta, la loro stessa sopravvivenza. La competizione globale pone quindi nuovi problemi da affrontare con modalità altrettanto originali. L'innovazione costituisce un primo elemento strategico, la chiave per fornire nuove soluzioni ai nuovi problemi. La realizzazione di progetti innovativi è quindi tatticamente importante ai fini della competitività. Il successo dei progetti economicamente rilevanti è assolutamente prioritario. La capacità di realizzare prodotti e servizi innovativi di qualità, a prezzi competitivi ed in tempi brevi rappresenta la chiave per la competitività e la crescita.

Ma come affrontare tali temi? Harvard Business Review riporta lo studio¹ effettuato sulle cause profonde che hanno portato 50 aziende-campione ad uno stallo dei ricavi. Gli studi hanno identificato tre fattori principali: *fattori strategici* (70%), *fattori organizzativi* (17%) e *fattori esterni all'impresa* (13%) (Figura 1). I fattori strategici che maggiormente contribuiscono a risultati negativi sono: *prigione di una posizione premium, incapacità di gestire l'innovazione, abbandono prematuro del core business, carenza di talento manageriale*.



Relativamente all'ultimo fattore menzionato, il presente articolo fornisce alcuni suggerimenti utili per migliorare la gestione dei progetti IT.

I progetti software

Il software, e le soluzioni IT in generale, rappresentano un elemento strategico per il business delle aziende. Dalla capacità di realizzare soluzioni innovative dipende la possibilità di competere in un mercato globale sempre più difficile. Soluzioni innovative sono richieste in tempi brevi, a costi competitivi e con qualità adeguata. I fornitori trovano difficoltà a soddisfare tali richieste. I clienti fanno fatica a selezionare fornitori qualificati.

La sfida, dunque, vede impegnati clienti e fornitori. Le strategie degli uni dipendono dalle capacità realizzative degli altri.

¹ Matthew S. Olson, Derek van Bever e Seth Verry, *Quando l'azienda entra in stallo*, Harvard Business Review, aprile 2008 n. 4.

Progetti, obiettivi e ritorni economici

Un progetto di sviluppo software rappresenta sempre un investimento importante per il cliente. Il suo obiettivo principale è quello di realizzare una soluzione che soddisfi appieno le esigenze del mercato, nei tempi richiesti e con costi adeguati.

Rispettare i tempi di consegna, contenere i costi entro il budget stabilito e produrre una soluzione che indirizzi correttamente tutti i requisiti sono obiettivi imprescindibili di ogni progetto strategico.

Purtroppo, moltissimi progetti non raggiungono gli obiettivi stabiliti con immenso danno per il business dei clienti e dei fornitori.

I progetti in difficoltà

L'ingegneria del software, benché ricca di studi e ricerche, è ancora poco applicata come disciplina nella pratica quotidiana. A differenza di altre branche dell'ingegneria, nel campo del software spesso si "naviga a vista", senza l'ausilio di strumentazione adeguata. Si produce "manualmente", piuttosto che con strumenti di produttività. Si "spera" in risultati positivi, e non si adottano metodiche di previsione. Le metriche sono poco utilizzate se non a posteriore per ratificare risultati negativi. Tutti gli studi condotti in proposito rilevano una scarsa conoscenza di metodi, tecniche, strumenti e metriche del software da parte degli addetti. In particolare, lo studio condotto da Standish Group nel 2003 su più di 30.000 progetti ha rilevato che solo il 28% di essi raggiunge pienamente gli obiettivi. Tutti gli altri raggiungono solo parzialmente ed un terzo di essi li fallisce nella loro totalità (Tabella 1).

Tabella 1. Progetti e loro risultati.	
Progetti completati con successo (nei tempi, costi e qualità attesi)	28%
Progetti con forte ritardo rispetto ai tempi previsti	63%
Progetti con costi oltre il budget previsto	49%

Una nota di ottimismo è data dal fatto che le percentuali di progetti che falliscono sembrano diminuire negli ultimi anni, specialmente nelle organizzazioni più mature nell'ambito del project management. Tuttavia, moltissime organizzazioni si avvalgono ancora di competenze approssimative e non di professionisti. Le competenze si limitano agli aspetti puramente tecnici (linguaggi, compilatori, sistemi e sottosistemi, framework, ecc.), poco su metodi e tecniche, e ancor meno su metriche. Lo scarso utilizzo di strumenti automatici completa il quadro negativo.

Problemi dei progetti software

Gli studi condotti sul fallimento dei progetti hanno evidenziato gravi carenze nelle aree:

- interpretazione dei requisiti;
- stima e pianificazione dei progetti;
- collaudo dei prodotti;
- attenzione alla qualità.

Esse evidenziano una scarsa attitudine a collaborare, all'interno dei progetti, tra committente e fornitore. Le cause di tali problemi confermano le ipotesi. L'approccio collaborativo presentato tende a superare tali problemi e a migliorare la percentuale di progetti di successo.

10 azioni suggerite per superare le cause principali dei problemi

Di seguito sono elencate dieci azioni determinanti suggerite per superare i principali problemi che affliggono i progetti software; tali azioni sono state sperimentate con successo dall'autore in progetti reali e sono stati inclusi nella metodologia di sviluppo software messa a punto e presentata con il nome di "Metodologia collaborativa per lo sviluppo del software":

1. Dedicare maggiore attenzione all'analisi dei requisiti

I requisiti risultano spesso incompleti, poco chiari, non controllati, indirizzati parzialmente e non secondo l'interpretazione degli utenti.

La collaborazione tende a garantire che i requisiti siano ben chiari, compresi, definiti ed indirizzati correttamente nel progetto. Da ciò può dipendere il successo del progetto.

2. *Aumentare la qualità della progettazione*

La progettazione dei sistemi software risulta spesso non basata su di una architettura consolidata, poco strutturata e con componenti poco coesi, con base dati poco strutturate, poco flessibile e difficile da espandere.

La collaborazione deve mirare ad affidare la progettazione a personale tecnico con competenze adeguate a garantire un'architettura tecnica, applicativa e basi dati di qualità.

3. *Effettuare stime accurate dei progetti*

Stime poco accurate rappresentano un dato molto preoccupante in quanto risulta scarsa la cultura di effettuare stime attendibili e realistiche, basate su progetti analoghi, eseguiti da persone esperte, seguendo una WBS, ecc.

La collaborazione affida le stime a persone esperte in grado di garantire accuratezza e correttezza.

4. *Pianificazione "realistica" dei progetti*

Piani troppo ottimistici o condizionati dalle aspettative della direzione rappresentano un altro aspetto molto negativo; sono molti i casi di stime fatte per "accontentare" la direzione o il cliente; si tratta della sindrome dello struzzo: nascondere la testa sotto la sabbia!

La collaborazione garantisce piani realistici in grado di essere rispettati. Entrambe le parti si impegnano sullo stesso piano. Il buon senso vince!

5. *Gestione adeguata dei rischi*

La situazione attuale rilevata mostra una mancata individuazione e valutazione dei rischi di progetto in fase iniziale, azioni inadeguate, reazione tardiva ai problemi e poca prevenzione.

La collaborazione evita di nascondere i rischi e li evidenzia con tutti i loro impatti negativi sul progetto. La condivisione dei rischi permette di intraprendere le misure di contenimento più adeguate e di tenerli sotto controllo durante l'intera durata del progetto.

6. *Risorse adeguate alla complessità del progetto*

La scarsità, come numero e come competenze, di un adeguato numero di professionisti del software è, a mio avviso, l'aspetto più preoccupante. La "costruzione" di una popolazione di bravi progettisti e programmatori non si improvvisa dall'oggi al domani. Si tratta di un debito professionale di competenze che il nostro paese accumula ogni anno. Nel resto d'Europa il problema è indirizzato in maniera decisa. Noi stentiamo a trovare una soluzione. A metterla in pratica nemmeno a pensarci!

La collaborazione tende a selezionare le risorse con le competenze adeguate e a gestire correttamente l'eventuale mancanza (trovare altrove le risorse mancanti, posticipare funzionalità in rilasci successivi, ecc.).

7. *Revisioni tecniche dei prodotti intermedi realizzati*

Risulta quasi del tutto assente l'utilizzo della tecnica di ispezionare i documenti realizzati nelle fasi alte del progetto. Le poche revisioni tecniche che si fanno sono su base spontanea, viste come momenti di confusione, poco produttive, mai pianificate. Invece sbagliamo.

La collaborazione permette di pianificare le revisioni tecniche in quanto "indispensabili" per verificare la correttezza della progettazione e delle scelte tecniche, in grado di rimuovere gli errori nelle fasi alte del progetto, ridurre la propagazione degli errori dalla progettazione al codice; l'unica tecnica a basso costo per controllare la qualità del software!

8. *Adeguatezza dei test eseguiti*

I test sono valutati in media solo il 5-10% del costo complessivo del progetto. Spesso sono pianificati con risorse non adeguate (poco "skillate", come suol dirsi). La progettazione non copre mai tutti i requisiti, specie quelli qualitativi. L'esecuzione è spesso spostata in avanti causa il ritardo nella programmazione. L'esecuzione è generalmente interrotta quando finisce il tempo o le risorse a disposizione. Credo siano commessi pressoché tutti gli errori che il buon senso suggerirebbe di evitare.

La collaborazione tende a considerare il test come attività chiave per la qualità del prodotto finale garantendone la corretta pianificazione in base alla criticità dell'applicazione per il business, alla complessità dell'architettura, ai rischi individuati. Il test non potrà essere minore del 25-30% dell'impegno totale del progetto.

9. *Utilizzo di strumenti per la gestione delle modifiche, della configurazione e degli errori*

Pochissime organizzazioni adottano metodi, tecniche, processi e strumenti per la gestione di queste tre componenti importanti del software: modifiche, errori, configurazione.

La collaborazione tende a stabilire l'importanza del tema e, quindi, di conseguenza, l'adozione di strumenti, anche molto semplici, come un foglio elettronico!

10. *Controllo della qualità*

Il piano della qualità è presente in quasi tutti i progetti, anche perchè richiesto dalle norme. L'esecuzione del piano, il controllo dell'esito delle attività, la valutazione dei risultati risulta invece pressoché assente nella maggior parte dei progetti, se non quando previsto dal monitoraggio esterno di terza parte. La qualità è spesso sacrificata sull'altare del compromesso tra tempi da rispettare e budget insufficiente.

La collaborazione tende a sviluppare la cultura della qualità intesa come "attitudine", "modo di agire", "risultato delle attività quotidiane", ecc. Assegnare risorse allo svolgimento delle attività previste dal piano della qualità permetterà di raggiungere l'obiettivo del progetto: l'accettazione del prodotto finale da parte degli utenti.

Il contesto italiano attuale

Il contesto IT italiano, oggi, vede luci e ombre. In parte esse rallentano lo sviluppo del settore e dell'intero sistema paese e, dall'altro, offrono spunti di ottimismo per il futuro.

Un primo dato negativo è offerto dai vari studi condotti dagli enti di ricerca sullo stato dell'ICT in Italia. I risultati mostrano un netto ritardo dell'Italia in termini di disponibilità di competenze adeguate, rinnovamento tecnologico e di investimento.

L'attuale crisi finanziaria ed economica accentua il problema, sottrae risorse agli investimenti, genera un clima di sfiducia che impedisce di affrontare il problema con la giusta determinazione.

Segnali di debole ottimismo sono forniti dallo sforzo che l'Amministrazione pubblica sta compiendo già da qualche anno per recuperare terreno ed allineare la "macchina dei servizi" allo standard europeo ed internazionale.

Altro segnale positivo, anche se debole, è dato dalla richiesta di formazione professionale.

I segnali sono pochi ma mostrano spiragli di luce. La fiducia non può e non deve mancare.

La chiave per invertire la tendenza e dare impulso alla crescita e alla competitività è, come già detto, la "crescita professionale". Di tutti. Nessuno escluso. Ciascuno nel proprio ruolo!

L'ingegneria del software non basta

Il ritardo cronico nella consegna dei progetti, i costi oltre i budget e la qualità non sempre all'altezza delle aspettative sono i problemi che da sempre affliggono i progetti software.

Negli anni i ricercatori hanno trovato soluzioni diverse a problemi specifici. Le soluzioni risultano appropriate ma scarsamente adoperate. Non tanto perchè queste siano difficili o poco applicabili ma, piuttosto, perchè richiedono "coerenza" e "metodicità"; due qualità, purtroppo, poco presenti nella maggior parte degli ambienti di sviluppo.

La creazione di una nuova disciplina - l'Ingegneria del software (eravamo nel 1968) – non è riuscita a risolvere totalmente il problema. Le nuove tecnologie offrono potenzialità di sviluppo notevole, ma la loro adozione richiede organizzazioni mature in grado di trarre il beneficio atteso dalla loro applicazione.

La definizione di modelli per migliorare le prestazioni delle organizzazioni software – ISO 9001:2000, EFQM, CMMI e altri modelli - prosegue con impegno e fornisce risultati anche molto positivi quando questi vengano applicati con buon senso.

Molte sono le istituzioni che in Italia – tra queste Aicq in prima linea - e a livello internazionale (PMI, SEI, ...), operano per diffondere la cultura della qualità, stabilire regole, fornire linee guida, proporre metodologie consolidate, preparare i professionisti.

La soluzione passa quindi attraverso lo sviluppo delle competenze a tutti i livelli, tecnici e professionali, manageriali e dirigenziali. Lato cliente e lato fornitori.

Sviluppo delle competenze

Lo sviluppo del software è un'attività tecnica e creativa complessa ad alto contenuto umano che richiede, oltre a competenze tecniche, altre capacità: professionalità, maturità e costante aggiornamento. L'evoluzione rapida delle tecnologie offre nuove possibilità di realizzazione, ma richiede un continuo aggiornamento delle competenze e nella capacità di gestire la maggiore complessità. Lo sviluppo delle competenze IT è richiesto a tutti i livelli: direzione tecnica, commerciali, capi progetto e personale tecnico.

Al buon esito dei progetti è legato il successo delle aziende. Occorrono quindi professionisti - a tutti i livelli - in grado di assicurare tali risultati. Spesso ci si focalizza solo sui capi progetto, addossando loro tutte le responsabilità e gli oneri. Un progetto, invece, è un tipico lavoro di gruppo il cui risultato dipende dalla competenza di ciascuno: dalla capacità di definire strategie vincenti da parte della direzione alla concretezza del commerciale, dalla capacità organizzativa del capo progetto alla competenza tecnica del gruppo di lavoro.

Lo sviluppo delle competenze IT, inquadrato in un piano di formazione continua, è un elemento chiave per la competitività delle aziende.

Partnership

I clienti affrontano nuove sfide e chiedono ai fornitori di condividere rischi e successi. Il tradizionale rapporto cliente-fornitore si evolve in quello più maturo di "partnership" dove entrambi collaborano per raggiungere lo stesso obiettivo: il successo nel mercato in cui si opera. Tale trasformazione richiede alle imprese di innovarsi ed affrontare le sfide del mercato con maggiore determinazione e professionalità. Ai loro fornitori-partner i clienti chiedono affidabilità, competenza e flessibilità nell'affidare loro i progetti strategici.

Soluzioni e progetti di realizzazione pongono quindi l'accento sull'innovazione da un lato e sul raggiungimento degli obiettivi dei progetti dall'altro.

Nel contesto italiano attuale è ancora alto il numero di problemi gravi nei progetti: ritardo nelle consegne, costi oltre il budget, qualità inferiore alle attese. Nasce quindi l'assoluta necessità di migliorare per competere!

Il miglioramento vede l'organizzazione nel suo complesso: competenze delle persone, adozione di processi maturi e di best practice, utilizzo di strumenti per la produttività e la qualità.

Il primo elemento – competenze delle persone – rappresenta sicuramente quello più critico. Lo studio HBR cui si è fatto riferimento prima ha già messo in luce l'importanza del talento manageriale. "Sviluppare software – scrive Adriano Comai² - è un mestiere che ha implicazioni sociali, e comporta un insieme di responsabilità nei confronti di chi lo ha commissionato, di chi lo acquista e di chi lo utilizza ..."

Il ruolo del committente

"Un paradosso – dice ancora Comai -: Il committente è il principale decisore dei progetti di sviluppo software, ma non esiste una formazione per diventare committente: la strada più comune è quella di imparare dai propri sbagli". In generale, il committente è la figura disposta a sostenere i costi per il progetto, e che ha la responsabilità di valutare se il prodotto risultante è accettabile. E' il principale decisore del progetto, ed il suo ruolo deve essere noto ed accettato da tutte le parti interessate all'andamento ed agli esiti del progetto (gli "stakeholder"). Tra le responsabilità tipiche di un committente troviamo:

- partecipare in modo attivo al progetto;
- coinvolgere nel progetto le altre parti interessate;
- esprimere requisiti, e chiarimenti sui requisiti;
- definire le priorità sui requisiti;
- controllare l'avanzamento dei lavori, ed i prodotti / servizi che costituiscono il risultato del progetto
- gestire i rischi di progetto;
- prendere decisioni in caso di conflitto (ad esempio, di contrasto tra i punti di vista dei vari stakeholder), o di incongruenza tra risorse e obiettivi (ad esempio, in caso di ritardi nello sviluppo non rimediabili, il committente deve decidere se è meglio rispettare una data di rilascio prevista rimandando lo sviluppo di alcune funzionalità oppure se completare tutte le funzionalità rimandando il rilascio).

² Adriano Comai, *Come sopravvivere a un progetto software – alcuni consigli per i committenti*, www.analisi-disegno.com

Cosa pensano i committenti

“... a prescindere dalle condizioni economiche dell'azienda cui fanno riferimento, i budget IT continuano ad essere inesorabilmente tagliati, rendendo la contesa sempre più aspra. “E’ quando scrive Massimo Messina³, responsabile IT di un grande gruppo bancario, nel suo blog ed intervistato dall'autore. “I clienti si lamentano – aggiunge - che i loro fornitori non li capiscono; dicono che hanno una serie di attese non corrisposte in termini di flessibilità, qualità ed efficacia, lasciando intendere che il problema non è quanto si spende ma come si spende”. Egli identifica sei atteggiamenti di sicuro effetto positivo, intuibili ma scarsamente applicati:

- Conoscere il cliente, pensare ed agire come se fosse il cliente stesso, capire le sue reali esigenze e criticità specifiche per fornire soluzioni adatte ai suoi problemi, piuttosto che offrire soluzioni a catalogo, come spesso avviene;
- Trattare clienti diversi in maniera diversa, in base alle sue caratteristiche specifiche, piuttosto che in maniera asettica come un target prestabilito; le soluzioni devono essere personalizzate ma a prezzi paragonabili ai prodotti e servizi base;
- Agire come business partner, per creare nuove opportunità utilizzando le proprie competenze specifiche e peculiari;
- Capire e guidare il cambiamento, agendo da abilitatore del processo, sia proponendo il cambiamento che realizzandolo, traendo vantaggio dall'esperienza di progetti simili;
- Fornire un servizio di qualità, garantendo continuità ed omogeneità della prestazione;
- Imparare a collaborare in modo competitivo, negli ambienti complessi, con gli altri fornitori presenti dallo stesso cliente in ottica di integrazione delle soluzioni offerte, semplificazione delle comunicazioni, gestione dei problemi.

Cliente e fornitori collaborano per il successo del progetto

Un approccio di collaborazione tra cliente e fornitore può creare le condizioni necessarie per garantire il successo del progetto. Ciascuna parte, ognuna nel proprio ruolo, collabora per definire le esigenze di business e le funzionalità necessarie, identificare la migliore soluzione tecnica, concordare il giusto compromesso tra qualità, tempi e costi. L'area di collaborazione, come mostrato nella Figura 2, è ampia.

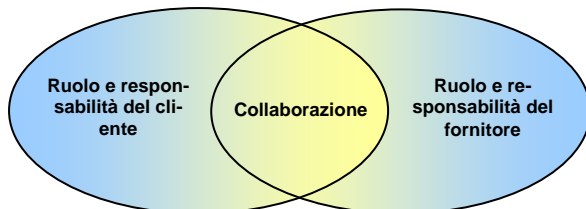


Figura 2. Collaborazione cliente-fornitore.

Le relative pratiche sono rivolte sia ai responsabili IT dei clienti che ai professionisti che operano come fornitori. Entrambi, infatti, sono interessati al successo dei progetti.

I primi hanno bisogno di soluzioni efficaci per la crescita del loro business, i secondi legano al successo dei progetti realizzati la crescita della propria azienda.

Nella realtà, invece, ognuno svolge il proprio ruolo, spesso in contrapposizione, dimenticando che il successo dipende anche, e soprattutto, dalla loro collaborazione. La collaborazione è dunque d'obbligo. Vediamo alcuni esempi positivi.

Alcuni esempi

Il cliente che collabora alla condivisione dei requisiti fornisce un contributo prezioso allo sviluppo di una soluzione che indirizzi realmente le proprie esigenze. Verificare che siano effettuate stime accurate e siano prodotti piani realistici consente di ridurre drasticamente il rischio di progetto. Condividere le priorità e le criticità dei requisiti è importante per pianificare rilasci in sequenza. L'efficacia della tecnica “time boxing” è ormai consolidata: essa permette di realizzare progetti in grado di rispettare i tempi di consegna e di contenere i costi entro il budget. Condividere una strategia di test basata sui requisiti permette di verificare la qualità del prodotto finale e assicura un esito positivo del collaudo di accettazione.

Condividere l'identificazione dei rischi del progetto e la valutazione del loro impatto, e concordare le azioni più opportune accresce notevolmente le probabilità di successo e ridurre gli imprevisti.

Condividere inoltre gli impatti delle modifiche sui piani e sui costi del progetto permette di concordare le azioni più opportune ai fini del successo finale: rivedere le priorità ed i contenuti dei rilasci, i costi a beneficio di un rapporto equilibrato tra contenuti, tempi, costi e qualità.

³ Massimo Messina, *Il System Integrator e le attese del cliente*, www.massimomessina.it

Il cliente che, volutamente o meno, sottovalutasse l'importanza della collaborazione diventerebbe corresponsabile di eventuali problemi e ritardi nei progetti.

Gli esempi relativi al beneficio che ne può trarre il fornitore dal clima di collaborazione sono altrettanto numerosi e importanti. La condivisione dei requisiti, delle loro priorità e della loro importanza costituisce la base di partenza del progetto. La condivisione di quali requisiti includere nei vari rilasci è il secondo tassello. La condivisione di stime accurate e di piani realistici aggiunge un terzo importante tassello. La condivisione della soluzione tramite la presentazione di prototipi rappresenta il quarto tassello. La condivisione della strategia di test e collaudo basata sui requisiti fornisce un importante quinto tassello. La condivisione dei rischi di progetto, dello stato di avanzamento delle attività e dei problemi incontrati completa i tasselli su cui costruire il successo del progetto!

Il fornitore che non riuscisse a collaborare con il cliente, volutamente o non, sarebbe artefice di eventuali problemi e ritardi dei progetti.

La soluzione migliore vede quindi l'alleanza strategica tra cliente e fornitore, entrambi impegnati a vincere la "stessa" sfida. La scelta è dunque obbligata: collaborare affinché gli investimenti nei progetti diano i risultati attesi. Il successo dei progetti dipende da entrambi. Nessun progetto può fallire.

Il ruolo del cliente

Il cliente accorto definisce le strategie per competere nel mercato e sceglie il "partner" che meglio lo sostenga in tale iniziativa. Scegliere il partner giusto significa accertarsi delle sue reali capacità in termini di:

- Affidabilità, capacità di rispettare gli impegni, anche nei momenti di difficoltà;
- Competenza, capacità di trovare soluzioni innovative e di realizzarle nei tempi e con i costi pianificati e con la qualità attesa;
- Flessibilità, capacità di adattarsi positivamente alle mutate esigenze che necessariamente si presenteranno lungo il percorso.

Scelto il partner giusto, occorre condividere rischi e benefici. Questo significa chiedere ciò che realmente si possa e si debba fare. L'opposto è dannoso per entrambi. La definizione di progetti realistici con obiettivi raggiungibili, con ambito definito e requisiti non ambigui è responsabilità di entrambe le parti. Il successo finale dipenderà da ciò.

Il ruolo del partner

Nel quadro di alleanza strategica per competere e vincere sul mercato, le aziende informatiche sono chiamate a:

- Produrre a costi inferiori (a costi competitivi);
- Ridurre i tempi di realizzazione (rilasciare i prodotti nei tempi richiesti dal mercato e dal cliente);
- Migliorare la qualità dei prodotti realizzati (essere aderenti ai requisiti richiesti dal mercato e dal cliente).

La strada da percorrere è dunque ben definita: migliorare per competere. Sviluppare le competenze tecniche e professionali, migliorare i processi produttivi e gestionali, dotarsi di strumenti che aumentino la produttività e migliorino la qualità.

La sfida consiste nel trovarsi pronti a condividere con i propri clienti rischi e benefici. Il mezzo è assicurare il successo dei progetti. L'iniziativa è creare le competenze necessarie. L'obiettivo finale è creare e mantenere la fiducia dei clienti.

La "soddisfazione del cliente" è il termometro delle capacità acquisite; la "fidelizzazione" è il beneficio concreto, la misura del successo basato su di un rapporto di fiducia duraturo nel tempo.

La collaborazione durante il ciclo di vita del software

Un progetto, si è detto, rappresenta un investimento strategico per le imprese ed il suo successo è un elemento fondamentale per la competitività. Un approccio di collaborazione tra cliente e fornitore crea le condizioni necessarie per il suo successo. Cliente e fornitore, ciascuno nel proprio ruolo, collaborano per definire la soluzione più adatta alle esigenze del business e concordare il migliore compromesso tra qualità, tempi e costi.

Gli ingredienti per realizzare progetti di successo sono quindi: la collaborazione offerta dal cliente, l'affidabilità del fornitore, la competenza e la professionalità delle persone coinvolte nel progetto, l'utilizzo di un metodo di lavoro semplice ed efficace.



Figura 3. Ciclo di vita del software

La collaborazione mira ad indirizzare i rischi di ciascuna fase del ciclo di vita del software mostrato nella Figura 3. I rischi specifici di ciascuna fase e le relative azioni suggerite sono riassunti nella Tabella 2 e poi descritti in dettaglio nel seguito.

Tabella 2. Elementi di collaborazione nel ciclo di vita.	
Fase	Collaborazione
Ideazione	Requisiti: condivisione di dettaglio per eliminare ambiguità, incertezze, mancanze. Soluzione: condivisione generale e verifica della sua corrispondenza alle esigenze di business. Stime: verifica della loro accuratezza e correttezza. Piani: verifica che siano realistici, realizzabili e basati sui requisiti. Rischi: valutazione e condivisione della stima degli impatti sul progetto e dell'efficacia delle azioni.
Progettazione	Soluzione applicativa: valutazione della corrispondenza alle esigenze di business tramite i prototipi realizzati. Architetture: condivisione della pianificazione delle revisioni tecniche e dei risultati. Risorse: valutazione dell'adeguatezza in termini di numero e competenze. Modifiche al disegno: condivisione degli impatti sul progetto: soluzione, tempi e costi.
Realizzazione	Sviluppo del software: condivisione puntuale dello stato di sviluppo del prodotto, dei problemi e dei rischi. Test: condivisione del livello di copertura dei requisiti e dell'efficacia del test eseguiti. Modifiche: condivisione degli impatti sulla qualità del prodotto.
Rilascio	Collaudo: condivisione degli obiettivi, del piano, dei risultati e delle correzioni richieste. Rilascio in esercizio: condivisione dello stato di approntamento della versione finale del prodotto, degli ambienti e della formazione degli utenti.

Collaborazione durante la fase di "Ideazione"

Il rischio connesso a questa fase è principalmente di tipo economico. Esso è legato all'indeterminazione che esiste nella valutazione dell'impegno richiesto per sviluppare la soluzione basata su un'analisi del contesto, delle esigenze e dei limiti certamente non affinata per ovvi motivi.

E' importante, in questa fase, collaborare per capire il meglio possibile le reali esigenze e poter offrire la soluzione più adatta. E' utile consentire ai fornitori capaci di offrire soluzioni basate su di una corretta interpretazione delle esigenze. La collaborazione in questa fase può includere la valutazione dei prototipi con gli utenti finali e la valutazione dei criteri di completamento della fase.

Collaborazione durante la fase di "Progettazione"

La fase di progettazione è quella a rischio più alto ai fini del successo del progetto. Il rischio maggiore è di tipo tecnico-architettonico. L'architettura deve infatti rispondere pienamente e correttamente a tutte le esigenze applicative, tecniche e prestazionali specificate nei requisiti. Le stime devono essere accurate ed il piano di progetto deve risultare realistico ed eseguibile nei tempi e con i costi previsti. Le persone coinvolte in questa fase sono di livello alto ed il loro impegno costituisce una porzione rilevante dei costi complessivi del progetto (anche il 40%). Gli errori nella progettazione creano impatti negativi sul resto del progetto: qualità del software, costi e tempi di realizzazione. Alcuni errori potrebbero anche non essere più risolvibili nelle fasi successive, compromettendo anche il completamento del progetto! L'approccio iterativo adottato nella fase e la realizzazione di prototipi permette di minimizzare il rischio.

L'importanza della collaborazione tra cliente e fornitore serve proprio a ridurre il rischio di cui si è detto prima. Condividere stime e piani riduce l'indeterminazione nei tempi e nei costi del progetto. Condividere i prototipi per tempo e verificare la capacità della soluzione proposta a rispondere alle esigenze del business del cliente è un'attività essenziale per ridurre il rischio.

Collaborazione durante la fase di "Realizzazione"

La fase di sviluppo e test è una tipica fase del processo produttivo i cui rischi sono legati all'utilizzo delle risorse, ai tempi di sviluppo ed ai costi di produzione. In questa fase occorre sapere esattamente cosa fare. I documenti di disegno, le specifiche e gli standard di programmazione descrivono cosa fare. I programmatori

ed i tester conoscono il loro mestiere. Il rischio è quindi legato all'efficacia della gestione delle risorse (persone presenti ma che non possono lavorare, ambienti non pronti, rifacimenti e duplicazioni, attività non eseguite per mancanza di personale, ecc.). La fase vede generalmente l'impiego di un grande numero di persone. Una giornata spesa male, ad esempio, non equivale solo a perdere un giorno di calendario nel piano temporale delle attività, ma incide negativamente sui costi di progetto per un ammontare pari al numero di persone impegnate!

La collaborazione riguarda, in questa fase, il controllo puntuale dello stato di avanzamento del progetto e l'identificazione delle azioni più opportune in caso di scostamento dai piani.

La fase necessita anche della collaborazione degli utenti finali (o loro rappresentati) per l'esecuzione di test di usabilità quando previsti. La scelta del campione rappresentativo degli utenti finali è importante ai fini della valutazione finale data al termine del test.

Collaborazione durante la fase di "Rilascio"

La fase di Rilascio è a rischio basso in quanto ci si aspetta, se il processo di sviluppo è stato seguito correttamente, che il prodotto finale abbia la qualità attesa. In questo caso, si assisterà alla scoperta di alcuni errori da parte degli utenti ed alla opportunità di migliorare qualche funzionalità non ben realizzata. Si tratta di correggere gli errori segnalati e completare/migliorare le funzionalità richieste. Il rischio è quindi basso.

Se così non fosse, siamo in presenza di un prodotto non adeguato alle necessità, che non offre le funzionalità richieste e che ha una qualità non accettabile. In questo caso, il rischio è molto alto e mette in forse l'intero progetto.

L'importanza della collaborazione tra cliente e fornitore nella fase di rilascio serve ad esercitare il prodotto in modo da poter valutare la completezza delle funzionalità previste e la stabilità del software. Gli utenti finali utilizzeranno il prodotto svolgendo i compiti loro assegnati, verificheranno la capacità di completare tutti i task (compiti) e valuteranno il livello di qualità (facilità d'uso, tempi di risposta, robustezza, ecc.). L'attività non è banale né a basso costo. Solo l'obiettivo comune delle due parti - cliente e fornitore - di raggiungere un buon livello di qualità prima di andare in produzione, potrà giustificare l'investimento, in tempo e risorse, da parte di entrambi. Ovviamente, è necessario che il prodotto abbia già con un livello di qualità accettabile, lasciando a questa fase solo il compito di verificarla ed accettarla. Un livello basso di qualità richiederebbe invece un impegno notevole di tempo e risorse non accettabile che porterebbe a conclusione negativa del progetto.

Conclusioni

In un contesto di grande competitività, aggravato da una crisi economica in atto, la collaborazione tra cliente e fornitore è d'obbligo. Il nemico (se così si può chiamare lo spettro della recessione che si prospetta all'orizzonte) lo si combatte – come in ogni giusta battaglia – insieme e mai disuniti! Si tratta, infatti, di condividere rischi e successo. La sconfitta sarebbe letale per entrambi. Il modello di collaborazione proposto mira a superare le difficoltà attuali e a suggerire azioni specifiche per ogni fase del ciclo di vita del software.

La sfida, ancora una volta, non è nei processi e nelle tecnologie – pur sempre importantissimi - ma piuttosto nella “capacità dell'uomo” di reagire agli eventi, di trovare soluzioni innovative, di adattare il proprio comportamento alle nuove situazioni, di modificare schemi mentali e superare pregiudizi, di cambiare i paradigmi comportamentali.



L'autore

Ercole Colonese è Consulente di Direzione e di Information Technology.

Collabora con Quality Solutions nella realizzazione di soluzioni di qualità per il business dei clienti. È socio AICQ, l'Associazione Italiana per la diffusione della Cultura della Qualità, dove partecipa attivamente alla realizzazione di eventi nazionali (seminari, corsi, articoli, quaderni monotematici, ecc.). È socio APCO, l'Associazione Italiana dei Consulenti di Direzione e di Organizzazione. Collabora

con le università di Tor Vergata, La Sapienza e il CIRPS in attività di formazione e ricerca. Ha lavorato in IBM dove ha ricoperto ruoli tecnici, manageriali e dirigenziali.

Opera in diverse aree di consulenza: Software Engineering (SE), Project Management (PM), Change Management (CM), Software Process Improvement (SPI), Total Quality Management (TQM), Sistemi di Gestione Integrati (SGI) per la Qualità, la Sicurezza, l'Ambiente, Adozione di modelli di maturità (CMMI), di eccellenza (EFQM) e di gestione dei servizi IT (ITIL e ISO 20000), Governance dei Sistemi IT. Tiene corsi professionali sui vari temi trattati.

e-mail: ercole@colonese.it

sito Web: www.colonese.it