

Perché i progetti informatici falliscono



Mille ragioni contribuiscono al fallimento di un progetto informatico. L'azienda o l'organizzazione che li promuove ha un solo modo per ridurre i rischi: cercare un buon capoprogetto e dargli l'autonomia necessaria.

Nessuno ne parla volentieri: né le aziende e le organizzazioni committenti, né tantomeno le software house che li realizzano o vi collaborano. I non addetti ai lavori lo vengono a sapere per caso: un'interrogazione parlamentare, il report di un analista se si tratta di un progetto importante. Altrimenti... gli eufemismi abbondano: "il progetto ha subito un ritardo", "è stato interrotto per un riesame strategico", ecc...

Ma perché tanti progetti informatici falliscono? Molti erano velleitari e impossibili da realizzare o legati a tecnologie ancora da definire e non testate, d'accordo. Ma parliamo di quelli realistici e realizzabili con le risorse in campo e analizziamo i problemi chiave che possono intervenire nella realizzazione di un grosso progetto informatico (per esempio, lo sviluppo o l'adattamento di una grossa applicazione alla realtà aziendale) in ordine decrescente di probabilità.

Mancanza di controllo sul perimetro del progetto. Il modo più facile per mancare il bersaglio è spostarlo. Una volta definito lo scopo del progetto, questo non deve più essere modificato. Le tentazioni e le pressioni possono essere forti, ma il tempo è l'elemento cruciale e si rischia di non rispettare più i termini originali (cosiddette milestone, pietre miliari). Si può prendere nota di eventuali richieste di modifica o di aggiunte, ma le stesse potranno essere effettuate solamente con rilasci successivi del software.

Disegno tecnico scadente. Il disegno dell'architettura tecnica è la parte meno intuitiva dei passi di un progetto, e richiede un grosso lavoro. I progetti falliscono perché sono basati su un'architettura debole e incompleta. Per certi versi l'architettura è ancora più

critica che non lo scopo del progetto, in quanto se nel primo caso si può rimediare con qualche ritardo, nel caso di un'architettura instabile si rischia di dover buttare tutto dalla finestra e rifare il lavoro da capo.

Risorse quantitativamente o qualitativamente insufficienti. Per 'far passare' un progetto capita spesso di sottovalutarne i costi in fase di preventivo. 'Costi' significa soprattutto persone e risparmiare significa utilizzare risorse non sufficientemente competenti e collaudate, oppure minori del dovuto. Questo errore può portare a risultati disastrosi. Un altro aspetto da non sottovalutare è l'ammontare di supervisione necessaria. Senza la necessaria guida, le risorse, a tutti i livelli del team, perdono di vista il loro compito principale e si perdono nei meandri: **Amministrazione inadeguata delle risorse.** Troppo spesso le (poche) persone chiave sono 'tirate' da tutte le parti. Una persona non può pianificare, supervisionare e sviluppare nello stesso momento. Un livello separato del gruppo di progetto dovrebbe essere disponibile per la pianificazione, il controllo e la risoluzione dei problemi.

Errori nella migrazione del sistema. Questo tipo di errori si manifesta all'ultimo momento nella fase di attivazione in produzione ed è dovuto a molte ragioni:

- un riconoscimento inadeguato delle necessità di purificare i dati provenienti dai vecchi sistemi;
- interfacce inadeguate tra il gruppo di sviluppo e il gruppo di migrazione;
- il nuovo software è stato collaudato con dati reali insufficienti o non rappresentativi;
- gli utenti reali non sono stati coinvolti nelle prove del nuovo sistema;

Siti da visitare

<http://spartathlon.webvista.net>

Sono 246 km da Atene a Sparta in un'unica tappa con tempo limite di 36 ore.

Colli e montagne da superare seguendo sentieri non tracciati.

www.spdo.com

'Foulée de la soie'. 250 km con tappe dai 10 ai 25 km, per 12 giorni lungo la via della seta in Cina. Caldo e umidità i nemici più feroci.

www.sandmarathon.com

La classica corsa nel deserto, con centinaia di partecipanti, organizzata in Marocco.

- alla fase di conversione non sono state assegnate sufficienti risorse.
- Funzionalità troppo pesanti.** Anche questo è un classico errore: in fase di progetto (o in corso d'opera) i progettisti non sono capaci di dire no alle mille richieste, o devono rispettare troppe compatibilità. Il risultato è un'applicazione pesante o complessa, che risulta ingestibile. A volte sono gli stessi progettisti a disegnare applicazioni over-engineered: troppo elaborate. Il meglio è nemico del bene, afferma un proverbio. Che cosa si deve fare per evitare un fallimento? Le soluzioni sono mille, ma al committente ne interessa soprattutto una: scegliere un buon capoprogetto, delegargli l'autorità necessaria e... ascoltare le sue richieste.

Renato Giovanelli

Ated
C.p. 572
6512 Giubiasco