l'autore

Luciano Manelli

Scegliere e commissionare **un software**

Guida pratica alla gestione dell'offerta tecnica per l'acquisto dell'applicativo perfetto.

Casi reali

EPC EDITORE

vai alla scheda del libro

Luciano Manelli

Scegliere e commissionare un software

Guida pratica alla gestione dell'offerta tecnica per l'acquisto dell'applicativo perfetto

Casi reali

EPC EDITORE

SCEGLIERE E COMMISSIONARE UN SOFTWARE
ISBN: 978-88-6310-771-5

Ristampa dicembre 2016

EPC S.r.l. Socio Unico - Via dell'Acqua Traversa, 187/189 - 00135 Roma
www.epc.it
Servizio clienti: 06 33245277 - Fax 06 3313212
Redazione: Tel. 06 33245264/205

Il codice QR che si trova sul retro della copertina, consente attraverso uno smartphone di accedere direttamente alle informazioni e agli eventuali aggiornamenti di questo volume.

Le stesse informazioni sono disponibili alla pagina:
https://www.epc.it/Prodotto/Editoria/Libri/Scegliere-e-commissionare-un-software/3412

Ai miei figli Sara e Marco,
a mia moglie Stefania,
ai miei genitori Anna e Piero.

SOMMARIO

PREFAZIONE

Questo testo si propone come una guida sintetica e completa che aiuti il professionista, il consulente, il dipendente di un'azienda o il funzionario pubblico a gestire un'offerta tecnica per la fornitura di un software sviluppato per l'organizzazione per cui si lavora. Spesso infatti, per soddisfare esigenze di piccola e media entità legate ad informatizzazione dei processi, dematerializzazione e nuove opportunità, vengono ipotizzati e commissionati a fornitori esterni progetti software, che sono seguiti da personale che, per mission o dimensione aziendale, non sempre ha le competenze necessarie ed indispensabili per poter ottenere un prodotto funzionante ed efficiente. Di seguito verranno quindi fornite schematicamente alcune linee guida indispensabili per poter gestire un progetto anche quando non dovesse rientrare nel ruolo e negli skill del richiedente. Ciò al fine di definire al meglio le richieste e ridurre al minimo la confusione delle offerte di fornitura, in quanto ogni azienda di sviluppo software ha un proprio approccio commerciale, lavora su diverse tecnologie e fornisce servizi differenti.

La seguente dissertazione, basata su casi studio reali e su un'esperienza quindicennale di consulenza dell'autore, rappresenta il punto di partenza per chi desidera ottenere rapidi e concreti risultati. Ha quindi lo scopo di accompagnare il lettore nella gestione di un progetto, dai concetti preliminari, al rapporto con il fornitore, alla definizione dei requisiti, alla valutazione della documentazione, fino all'indicazione di metodologie per l'analisi manageriale ed il controllo dell'evoluzione dello stesso.

Contatti: *it.linkedin.com/in/lucianomanelli*
e-mail: *clienti@epc.it*

Pagine omesse dall'anteprima del volume

capitolo 1

SVILUPPO DI UN PROGETTO SOFTWARE

La necessità di commissionare un progetto software è legata all'esigenza organizzativa di condividere informazioni, informatizzare procedure e salvare dati al fine di poterli processare anche successivamente. Concettualmente, l'idea è quella di accedere da uno o diversi dispositivi mobili o fissi all'interfaccia grafica dell'applicazione, che funzionalmente soddisfi le esigenze del cliente. Lo strato sottostante è solitamente costituito da un Server (che esegue le elaborazioni e genera le risposte all'utente) e da un DataBase per il salvataggio dei dati.

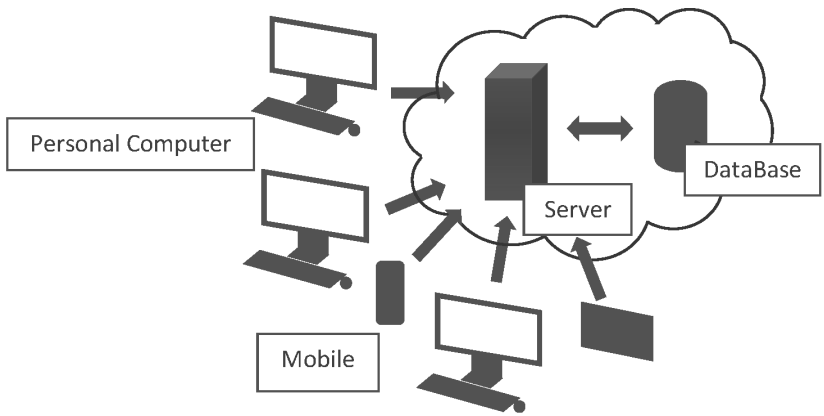

Fig. 1.1 – *Accesso da personal computer e dispositivi mobile alle applicazioni*

È necessario far notare che le tecnologie alla base dei sistemi software risultano complesse e varie, al fine di ottenere sempre un'applicazione usabile, sicura ed aperta. Diverse potranno essere le modalità di gestione del progetto e quindi differenti saranno gli strumenti, gli ambienti di sviluppo e i tempi di realizzazione da parte dei fornitori per il raggiungimento degli obiettivi del cliente. Conseguentemente, le valutazioni di seguito riportate sono valide sia per applicazioni web che per applicazioni desktop e mobile, in quanto individuano gli elementi minimi, ma indispensabili, di scambio informazioni e confronto tra cliente e fornitore.

1.1. Fattibilità e funzionalità del software desiderato

Anzitutto è importante stabilire esattamente "cosa" si vuole intraprendere al fine di valutare la fattibilità di un progetto, evitando di sprecare tempo e confondersi tra un ventaglio di offerte più o meno ampio e vario, che porterebbe a deviare dal reale scopo dell'applicativo chiesto e necessario agli scopi organizzativi. Conseguentemente, è necessario porsi differenti quesiti. Il progetto è fattibile? È realmente utile? È urgente? Porterà reali e concreti benefici? Quale budget si ha a disposizione per svilupparlo e fare manutenzione? È presente personale che possa seguire l'evoluzione del progetto e dedicarsi al test del software?

Solitamente, l'esigenza dello sviluppo di un nuovo software è legato a scopi organizzativi (al fine di efficientare gli applicativi di back-office, ovvero di ottimizzare i processi di lavoro interni quale evoluzione o reingegnerizzazione o rifacimento di un software per l'automazione di ufficio), alla semplificazione, ovvero dematerializzazione, degli strumenti per la gestione delle pratiche (utenti, personale, clienti, merci, ecc.), allo sviluppo, in ultimo, di nuove ed innovative soluzioni (di vendita, di affiliazione cliente o di gestione magazzino). Bisogna quindi dare particolare risalto agli obiettivi di business, che il progetto intende supportare, cercando di definire criteri oggettivi e quantificabili, che consentano di monitorare e valutare il successo del progetto stesso. A tal fine, è importante valutare, in funzione dell'ambito di riferimento, il contesto di mercato del servizio da sviluppare, gli operatori e gli utilizzatori

del sistema che si vuole sviluppare, i principali Stakeholder (interni ed esterni alla struttura organizzativa) che possono interagire direttamente con il progetto e con il servizio da richiedere, le eventuali applicazioni esterne coinvolte nel progetto e, in ultimo, sia lo scenario iniziale nel quale il progetto andrà ad intervenire, che il contesto dell'applicazione a regime. Soprattutto, è indispensabile individuare la persona di riferimento del progetto interna all'organizzazione, o esterna (consulente), in quanto principale (o unico) punto di contatto e primo utilizzatore e valutatore del sistema che si vuole sviluppare.

Successivamente, si deve comprendere, nel dettaglio, cosa realmente l'applicazione ha l'obiettivo di fare, ponendosi semplici (e mai esaustive) domande. È una nuova applicazione software? È la modifica o l'evoluzione di un'applicazione esistente? È necessario attivare dei servizi collegati? È necessario formare il personale preposto? È necessario attivare un servizio di assistenza? È importante impostare le giuste interazioni fra i soggetti coinvolti, definire in modo chiaro e dettagliato sia le motivazioni che hanno indotto la richiesta, che gli obiettivi che si desiderano raggiungere. Stabiliti gli elementi (obiettivi e persone) manageriali, sarà possibile quindi passare alle valutazioni funzionali (e quindi tecniche) che portano alla richiesta di un offerta da parte di uno o più fornitori. Infatti, al pari di un qualsiasi acquisto (casa, abbigliamento, ecc.), successivamente alla valutazione dei fabbisogni, è importante "cercare in giro", valutando qualità e convenienza, per poi stabilire se l'acquisto, in funzione del budget ipotizzato e delle reali esigenze, sia effettivamente fattibile.

1.2. Sito vetrina o web application?

Il primo passo è valutare se il servizio che si vuole far sviluppare serve nell'ambito dell'organizzazione (intranet) per l'informatizzazione di procedure interne, o all'esterno come vetrina su internet, o per interscambio di informazioni, ovvero per il caricamento di dati da soggetti terzi. È interessante valutare i due ambiti fondamentali (interno o esterno) in quanto variano sia gli aspetti di sicurezza che di interfaccia grafica e quindi il costo: chiaramente la cura di tali aspetti diviene alquanto importante nello sviluppo di portali ed applicativi esposti in rete.

1.3. Interfacciamento con il mondo esterno

Altro aspetto è legato alla necessità di interfacciare l'applicazione con "qualcosa di già presente", elemento importante in quanto può influenzare la tecnologia di sviluppo del software: infatti dovrà essere anche valutato il costo del connettore (implementato ad hoc o acquisito da terze parti). Si immagini un sistema di gestione di magazzino in 3D che si interfaccia con i dati del software gestionale preesistente, o un sistema di gestione clienti che si interfaccia con un nuovo CRM (Customer Relationship Management).

1.4. Cosa realmente si vuole dall'applicazione

Si passi, in ultimo, ad analizzare nel dettaglio il "cosa", ovvero alla redazione dei requisiti funzionali, partendo dagli aspetti generali e approfondendo ciò che realmente serve. Questa è la fase di generazione dei macro-requisiti. Non interessa avere un dettaglio approfondito sugli stessi in questa fase, in quanto non indispensabile e in quanto non è detto che esista personale specializzato all'interno dell'organizzazione che possa indicare tali dettagli. Sarà compito del fornitore indagare ed approfondire i requisiti in funzione di ciò che serve realmente. L'importante è non farsi "trasportare" dai fornitori che propongono soluzioni spesso talmente ampie da perdere il reale focus per cui è nato il progetto, rendendo inutilizzabili (e costose) la maggior parte delle funzionalità.

È necessario quindi valutare, studiare e scrivere le esigenze (il "cosa") che l'applicazione deve soddisfare: la persona preposta a tale compito deve avere piena conoscenza dei processi organizzativi e delle necessità aziendali, ovvero del "dominio" applicativo. Successivamente, sarà anche necessario affrontare le modalità con le quali le esigenze possono essere soddisfatte (il "come"), affinché possa essere implementato un applicativo utile, durevole, sicuro e scalabile (ovvero capace di crescere modularmente ed incrementare le proprie prestazioni senza la necessità di essere riprogettato).

Pagine omesse dall'anteprima del volume

capitolo 3

LA COMPLESSITÀ NELLA GESTIONE DI UN PROGETTO SOFTWARE

Il committente di un offerta, conoscitore del dominio applicativo e degli ambiti procedurali della propria organizzazione, deve saper impostare e dettagliare al meglio un'offerta tecnica al fine di ottenere il massimo risultato e facilitare il compito dei fornitori. Il lavoro non è complicato, anche per i non addetti ai lavori, in quanto ciò che conta è tenere sotto controllo gli elementi basilari principali e schematizzare le proprie richieste.

Nel seguente capitolo verranno forniti, di conseguenza, i principi base di gestione di un progetto software sia per comprenderne la complessità, sia per avere una supervisione consapevole dello stesso, dalla genesi alla consegna. Infatti, se si aumenta l'effort iniziale da parte del committente, sarà possibile, anche per chi andrà materialmente a sviluppare l'applicativo e quindi a formulare un'offerta economica, poter lavorare su richieste e funzionalità oggettive, concrete ed esaustive.

3.1. Il concetto di qualità

La qualità in un progetto software è facilmente e fondamentalmente rappresentato dalla soddisfazione del cliente. Il raggiungimento di tale obiettivo è chiaramente legato alla struttura organizzativa, alla definizione di responsabilità specifiche ed, in ultima analisi, alla gestione formale di controllo qualità (con le relative certificazioni) per

la migliore conduzione aziendale e progettuale. La pianificazione della qualità inizia il suo percorso dalla presentazione dell'offerta per poi concretizzarsi e razionalizzarsi nel momento della stesura del contratto e nelle fasi successive di sviluppo del progetto software. La qualità è, infatti, alla base di un rapporto di affiliazione e di collaborazione certificato, che rassicura il cliente e rafforza aziendalmente il fornitore. Gli obiettivi principali sono: sviluppare al meglio il software richiesto e restare confinati nei limiti economici stabiliti, attraverso un processo di accettazione delle funzionalità e controllo del progetto in maniera trasparente e condivisa. Contestualmente, è indispensabile evidenziare e mettere in conto che un progetto ben posto e sviluppato, ovvero di qualità, possa avere costi leggermente superiori a vantaggio di prospettive di elevata competitività, durabilità, manuntenibilità, scalabilità e modularità.

3.2. Principi di Sviluppo di un progetto software

Come indicato in premessa, è utile conoscere i principi di sviluppo e gestione di un progetto software per definire i requisiti base, gli obiettivi, i livelli di qualità attesi e i parametri di controllo al fine di relazionarsi al meglio con il fornitore, non solo in fase preliminare, ma durante l'arco di vita del progetto stesso. Si illustrano di seguito gli elementi fondamentali alla base del management di un progetto software.

3.2.1. Figure di progetto

Di fondamentale importanza sono le figure attuative presenti nelle varie fasi di progetto, quali soggetti standard per la promozione, la gestione e lo sviluppo dello stesso. Sicuramente è necessario individuare la persona o le persone (interne o esterne) di riferimento del committente (quale soggetto promotore del progetto) e il Project Manager che seguirà il progetto lato fornitore, oltre a tutte le figure di differente professionalità tecnica e manageriale interessate all'implementazione, alla manutenzione, al controllo e alla supervisione dello stesso.

3.2.2. Fasi di progetto

Un progetto è suddivisibile in diverse fasi che partono dall'ideazione dello stesso, alla sua qualificazione tramite un Business Plan, alla sua pianificazione tramite un Project Plan, alla sua esecuzione (che comporta un piano esecutivo e un piano di controllo), alla sua chiusura. Sinteticamente le fasi del ciclo di vita di un progetto software sono:

- Studio di fattibilità: rappresenta un elaborato che viene sviluppato prima dell'avvio di un progetto sulla base di un'idea di massima legata ad un nuovo prodotto o servizio. Nel caso in esame, tale studio rappresenta il primo step del committente per poter presentare una richiesta ai diversi fornitori e partner.
- Scrittura dei requisiti: successivamente allo studio di fattibilità diviene necessario stilare la documentazione relativa ai requisiti di progetto, basata su un'approfondita indagine conoscitiva e una dettagliata raccolta ed organizzazione di informazioni e risorse che permettano di stilare le caratteristiche che il sistema atteso deve avere (il "cosa").

Qui si ferma il lavoro preliminare del committente: lavoro che, in ottica della migliore esplicazione dei requisiti, necessiterà da parte dei vari fornitori vari ricicli, eventualmente accompagnati da strumenti che, graficamente, possano aiutare l'analisi e la comprensione dettagliata delle esigenze espresse. È possibile quindi passare alla fase successiva che comporta un effort da parte dei fornitori anche al fine di presentare un'offerta economica.

- Analisi dei requisiti: individua una descrizione completa delle funzionalità raggruppate solitamente in macrofunzioni e, gerarchicamente, in funzioni e sotto-funzioni che ne dettagliano le attività. Vengono anche stabiliti i requisiti sistemistici, eventualmente condivisi con il committente. In tale fase sarà necessario anche analizzare ulteriori elementi tecnico-prestazionali ed economici che vanno dalle risorse disponibili ai tempi e ai costi, al fine di generare la relativa offerta economica.

Una volta approvato il progetto e scelto il fornitore, parte lo sviluppo vero e proprio del progetto software.

- Progettazione: caratterizzata solitamente dalla progettazione dei dati e dell'applicazione. La prima riguarda tutte le fasi di progettazione di un DataBase, nell'altra si definiscono le caratteristiche dei programmi applicativi.
- Implementazione e relativa produzione del codice: rappresenta la fase di sviluppo e di scrittura del codice al fine di realizzare l'applicativo secondo le indicazioni individuate nella fase di progettazione. In tale fase vengono anche impostati e configurati i server e viene costruito e popolato il DataBase.

Una volta terminato lo sviluppo (parziale o totale) è necessario testare il sistema e collaudarlo per ottenere il riscontro da parte del committente o rientri per eventuali ricicli.

- Test, collaudo e validazione della committenza: si verifica il corretto funzionamento dell'applicazione attraverso una serie di test stabiliti sulla base del documento di analisi e progettazione, in modo da valutarne sia la correttezza delle specifiche implementate che la correttezza formale e grafica, valutando anche aspetti di accessibilità e usabilità oltre che le performance sistemistiche.
- Rilascio del software: caratterizzato dall'attivazione del sistema presso il cliente, dal rilascio della documentazione stabilita contrattualmente e dalla formazione eventuale del personale.
- Funzionamento e manutenzione: in tale fase il sistema informativo è operativo. Nel periodo di garanzia e assistenza si interviene in caso di malfunzionamenti evidenti e riproducibili e si lavora in manutenzione ordinaria (o eventualmente evolutiva) legata alla gestione dell'applicativo. La manutenzione, infatti, può essere impostata (contrattualmente) su tre livelli: manutenzione correttiva, manutenzione adattativa e, in ultimo, manutenzione perfettiva o evolutiva.

Queste fasi non sono rigide, ma portano a successivi ricicli fino ad un assestamento definitivo e alla conclusione del progetto software.

In alcuni casi è interessante, in fasi prestabilite contrattualmente, la presentazione di un prototipo o di uno sviluppo parziale (a funzionalità ridotte) al fine di confrontarsi con la committenza ed evitare problematiche di comunicazione ed errata interpretazione dei requisiti. Risultano quindi necessarie oltre alla competenza tecnica del fornitore, anche collaborazione, assertività, condivisione con la committenza e un buon management del progetto.

3.2.3. Metodologie di sviluppo

Lo sviluppo dei progetti presenta diverse metodologie a seconda dell'approccio del fornitore. Il modello a cascata, rigido, è stato uno dei primi utilizzati in cui ogni fase produce un ben preciso output documentato che viene utilizzato come input per la fase successiva, fino alla versione definitiva. Il modello a spirale si basa su un modello ciclico, consentendo lo sviluppo di versioni più ampie e complete del software ad ogni ciclo. In ultimo, il modello agile cerca di coinvolgere il più possibile il cliente, in modo tale da poter avere riscontri rapidi ed efficaci ad intervalli di tempo brevi e si fonda sul cambiamento rapido senza per questo rinunciare alla qualità del risultato. Qualunque sia, la metodologia scelta ottempererà alla soddisfazione di una serie di fasi, rilasci e milestone e presenterà sempre la produzione di documentazione formale, in ottica del piano di qualità presentato all'accettazione del contratto.

3.2.4. Analisi dei processi (BPMN) e degli scenari (Use Case)

È possibile accompagnare alle prime fasi del ciclo di vita di un progetto, soprattutto a livello di business e per progetti importanti, una modellazione dei processi ad alto livello che permetta agli Stakeholder di poter verificare e validare, a livello di management, gli elementi su cui si baserà lo sviluppo del progetto stesso. L'analisi dei processi di business è sicuramente l'ambito più rilevante del Business Modeling, in quanto è il punto di partenza per l'implementazione dei processi con tecnologie basate su motori di Business Process Management

(BPM). In tale situazione, il BPMN (Business Process Model and Notation) è lo strumento che permette di definire una notazione standard e grafico/visuale per modellare i processi, comprensibile da differenti attori (dai manager, da chi analizza i processi e da chi ne segue l'implementazione tecnologica) e riconosciuto a livello internazionale in quanto consente la scrittura di una notazione non tecnica ed intuitiva (riprendendo la logica dei diagrammi di flusso) per la visualizzazione dei processi di business.

L'analisi degli scenari nasce in un momento successivo dalla necessità di poter studiare in maniera chiara e partecipativa i requisiti di un sistema software, in modo da garantire una vista ad alto livello e stabilire le funzionalità che il sistema realizzerà. Infatti, a tale livello, è possibile concentrarsi sul "cosa" fare, astraendo dall'implementazione. Gli scenari si basano sulla necessità di individuare quali sono gli attori di un sistema e sulla necessità di conoscere cosa andranno a fare nel sistema. In tale situazione, gli Use Case del linguaggio UML (Unified Modeling Language) forniscono una descrizione del comportamento del sistema dal punto di vista dell'utente senza dover specificare come tale comportamento venga realizzato. Tale strumento permette di definire una notazione standard e grafico/visuale per modellare gli scenari, chiarendo la modalità di inizio, le risposte che l'utente si attende dal sistema e la sequenza di passi con cui l'interazione si svolge, con l'eventuale partecipazione di altri soggetti coinvolti anche esterni al sistema. Risulta utile, per la riuscita di un progetto software, poter ragionare con gli Stakeholder in termini di casi d'uso, agevolando la scoperta dei requisiti ed il loro approfondimento, costituendo, tale studio, il punto di partenza per le attività di analisi, progettazione, implementazione e test del sistema.

Inoltre lo UML rappresenta il linguaggio di punta nella descrizione di un software e nella gestione della relativa documentazione. Infatti, il diagramma delle classi permette di descrivere i tipi di entità, con le caratteristiche e le eventuali relazioni fra loro e risulta utile per esplicitare, ad esempio, un software sviluppato con linguaggi ad oggetti (quali C++ o Java). Il diagramma di sequenza viene utilizzato per rappresentare come avviene la comunicazione tra oggetti in relazione allo scorrere

del tempo. In ultimo, il diagramma dei componenti permette di evidenziare i moduli di sistema (definiti in esso come black-box) e le relative correlazioni.

3.2.5. Pianificazione del lavoro: WBS e cronoprogramma

Un progetto, non necessariamente informatico, è individuato da una serie di attività che dovrebbero essere il più possibile dettagliate, quasi atomiche, per poter averne un controllo completo ed esaustivo tale da definirne esattamente tempi di sviluppo e criticità e, quindi, costi per poi poter fornire il costo completo di progetto. Un progetto può essere scomposto (graficamente e, quindi, strutturalmente) disponendo gerarchicamente le attività ad un livello di dettaglio sempre maggiore attraverso la WBS (Work Breakdown Structure) che risulta uno strumento organizzativo indispensabile di gestione, in quanto permette di pianificare, assegnare e monitorare l'evoluzione del progetto stesso. La realizzazione della WBS è legata al processo di pianificazione, in quanto consente di costruire il reticolo delle attività e la relativa schedulazione attraverso lo sviluppo di un diagramma che definisce i tempi di consegna in base alle risorse e alle criticità: in tale ambito sono usati diagrammi quali PERT o GANTT in quanto determinano una vista temporale, ovvero un cronoprogramma, sulle attività individuate dalla WBS che considera il progetto solo dal punto di vista dei contenuti.

Volendo impostare una WBS legata alle specifiche funzionali individuate nell'analisi effettuata, sarebbe possibile definire, ad esempio, blocchi per la gestione dell'anagrafica azienda, per la gestione dei dati dell'impiegato, per la ricerca e per il layout da poter essere ulteriormente dettagliati.

Il relativo Cronoprogramma sarà legato alle attività della WBS in funzione dei tempi previsti per ogni singola attività e definiti dal Project Manager relativamente alle figure professionali selezionate (con costi differenti, ad esempio, per analisti e sviluppatori). I Cronoprogrammi sono uno strumento fondamentale per definire Project Lifecycle, MailStone e valutare il Critical Path (percorso critico individuato da attività

che non possono essere svolte parallelamente in quanto logicamente dipendenti tra loro in ottica di sequenza temporale), ma anche per dilatare o ripianificare tempi e consegne a seguito di analisi di Risk Management o di ricicli con il committente.

3.2.6. Strumenti di gestione e valutazione di un progetto software

Per sviluppare al meglio un progetto software è necessario adoperare alcuni strumenti indispensabili in ambito progettuale, di sviluppo e di gestione. L'obiettivo è quello di incrementare la comunicazione e il coinvolgimento degli Stakeholder e migliorare la capacità di risposta ai cambiamenti con un approccio preferibilmente iterativo, ottimale per ottenere una maggiore qualità del software realizzato, grazie anche all'adozione di *best practice*. Tra le principali linee guida e metodologie adottate per la migliore conduzione del progetto è possibile citare: ITIL (Information Technology Infrastructure Library), CMMI (Capability Maturity Model Integration), PMBOK (Project Management Body of Knowledge), SCOR (Supply Chain Operations Reference) e DCOR (Chain Operations Reference) e, in ultimo, le norme ISO (International Organization for Standardization). Tutte offrono i riferimenti e le indicazioni per migliorare processi, erogare servizi e gestire progetti di qualità al fine di raggiungere, coerentemente alle strategie di business, gli obiettivi della mission aziendale. Inoltre AgiD (Agenzia per l'Italia Digitale) definisce, in vari documenti, i criteri per la progettazione e lo sviluppo di un progetto software, la qualità, il monitoraggio ed il controllo dei livelli di servizio per i sistemi ICT relativi ai contratti della Pubblica Amministrazione (e non solo) e per la rendicontazione attraverso una soluzione orientata ai processi di business. Inoltre, possono essere usati strumenti concordati con il fornitore al fine di dare evidenza alle evoluzioni e alle aspettative di progetto, quali il SAL (Stato Avanzamento Lavori), gli indicatori KPI (Key Performance Indicator) e le SLA (Service Level Agreement). In ultimo, in alcuni ambiti, può essere interessante avere una stima oggettiva del lavoro da effettuare nel suo complesso attraverso strumenti, quali i Function Point, che comprendono fattori tecnologici e fattori umani.

3.3. Come impostare una richiesta tecnica: migliorare i requisiti del Caso Studio

La definizione delle specifiche rappresenta il momento più importante all'interno del ciclo di sviluppo dei sistemi software. Le analisi che emergono, infatti, servono a definire come il software debba funzionare e, successivamente, come lo stesso debba essere implementato. Una serie di specifiche rappresentano affermazioni precise sui requisiti che un sistema deve avere e soddisfare, specifiche che devono risultare comprensibili, chiare, non ambigue e complete e che si concretizzano in un accordo tra il committente e il fornitore. Conseguentemente, risulta necessario affrontare in maniera dettagliata i requisiti funzionali, ovvero le richieste del "cosa" e del "come" per darne evidenza ai potenziali fornitori, declinando l'analisi al Caso Studio affrontato. Il committente deve descrivere correttamente le funzionalità e i servizi che il sistema deve presentare e che devono essere implementati. Di seguito, dopo un'introduzione teorica, verranno esaminati i requisiti funzionali (il "cosa") in maniera descrittiva e schematica al fine di massimizzare la comprensione degli stessi. La profondità delle spiegazioni è legata alle necessità oggettive di progetto. I requisiti non funzionali (il "come"), diversamente, non sono legati alle funzionalità in senso stretto, ma a vincoli di contorno al sistema, ovvero alle modalità con le quali le esigenze possono essere soddisfatte, quali usabilità e accessibilità, business continuity e disaster recovery, sicurezza, preferenza di layout particolari e aderenza a norme legislative.

Si consideri, di conseguenza, nuovamente la proposta effettuata al paragrafo 2.2.

L'applicazione che si chiede servirà per inserire i dati delle aziende e sarà divisa in due sezioni, una riferita all'azienda e una agli impiegati. I dati saranno salvati su DataBase.

Nel modulo allegato è possibile reperire tutti i dati che servono.

Nella sezione dedicata alle aziende dovrà essere collegato l'elenco dei rispettivi lavoratori che dovrà presentare un collegamento al profilo del singolo. Nella sezione impiegati, sia nell'elenco che nella scheda del sin-

Pagine omesse dall'anteprima del volume