

La progettazione del software nelle piccole o micro imprese

Il contenuto di questo documento è strettamente confidenziale, la visione dello stesso è consentita solo al personale di FadeOut Software Snc e della società a cui esso è destinato

FadeOut Software

ICT & Automation Engineering

1. LA PROGETTAZIONE DEL SOFTWARE NELLE PICCOLE O MICRO-IMPRESA.....	4
1.1. INTRODUZIONE.....	4
1.2. BREVE STORIA DELL'INGEGNERIA DEL SOFTWARE	4
1.3. IL PROBLEMA.....	4
1.4. PRESENTAZIONE DEL PROCESSO	5
2. STUDIO DI FATTIBILITÀ PRELIMINARE.....	8
3. ANALISI DEL SISTEMA.....	9
4. PROGETTAZIONE	12
5. SVILUPPO	13
6. TEST	16
7. MANUTENZIONE.....	17
8. CONCLUSIONI.....	18
9. BIBLIOGRAFIA	19

1. La progettazione del software nelle piccole o micro-imprese

Ing. Paolo Campanella (p.campanella@fadeout.biz)

Ing. Alberto Tasso (a.tasso@fadeout.biz)

1.1. Introduzione

Il processo di ingegneria del software ha raggiunto ormai un elevato grado di maturità ed è possibile individuare una metodologia comprovata e formalizzata, che definisce i ruoli, gli elementi, le tempistiche e le azioni che ne fanno parte. Si tratta di un metodo di lavoro in grado di guidare l'ingegnere nell'analisi e nel progetto assicurando validi e dimostrabili risultati.

In questo articolo presenteremo un approccio all'ingegneria del software derivato dallo Unified Process (UP), che può ormai essere considerato uno standard "de-facto", in grado di adattarsi alle esigenze di una impresa di limitate dimensioni.

1.2. Breve storia dell'ingegneria del software

Lo standard dell'industria del software può sicuramente essere considerato ad oggi Unified Process, le cui radici risalgono al 1967, quando Ivar Jacobson, uno dei padri dell Unified Modeling Language (UML), il linguaggio che costituisce lo strumento di base dell'UP, era impiegato presso la Ericsson; l'azienda prese la radicale – per l'epoca – decisione di modellare un sistema complesso come l'insieme di diversi blocchi fra loro interconnessi. Dall'idea del "dividi et impera" il metodo si è evoluto negli anni, arricchito dai contributi di persone quali Booch e Rumbaugh, fino ad essere conosciuto in tutto il mondo tramite i prodotti di Rational Corporation, oggi parte di IBM, che si era assicurata i servizi dei padri del metodo. L'introduzione dello standard UML nel 1997 è stata seguita dal rilascio del RUP (Rational Unified Process) nel 1998 e dall'UP nel 1999.

1.3. Il problema

Come purtroppo spesso accade il materiale rintracciabile in letteratura è ad uso di imprese dotate di una grande struttura, affermate sul mercato e con forti capacità di investimento. Lo stesso UP, applicato alla lettera, è sicuramente un processo complesso da gestire essendo stato studiato e messo a punto negli anni da grandi imprese per la soluzione di problemi di vaste dimensioni, spesso per progetti della durata complessiva di anni. Esiste d'altro canto, specialmente in Italia, una grande forza di lavoro costituita dalle micro e piccole imprese, vero motore dell'economia nazionale.

In questo articolo presentiamo lo sforzo fatto per adattare e formalizzare i passaggi critici e fondamentali di UP in modo da poter ottenere la stessa indiscutibile qualità di progettazione anche per sistemi di dimensioni

Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 4 di 19

più comuni, come possono capitare alle tante piccole imprese o ai professionisti. Il metodo presentato è stato, ed è, utilizzato con successo dagli autori per gestire il proprio lavoro.

1.4. Presentazione del processo

Il processo di gestione delle commesse che verrà descritto in dettaglio nel seguito, è composto da sei fasi distinte, ognuna delle quali caratterizzata da un preciso diagramma di flusso. Per formalizzare il processo devono essere naturalmente accettate alcune approssimazioni al fine di non aumentare eccessivamente il dettaglio e la complessità dei diagrammi fino a renderli illeggibili. A questo fine va quindi specificato come il metodo sia in realtà un processo iterativo ed incrementale: specie al crescere delle dimensioni del problema le diverse fasi sono completate e ripetute al fine di ottenere, al posto di uno statico e monolitico ciclo, diversi cicli completi più dinamici e flessibili, in grado di adattarsi ai feedback ricevuti dal cliente durante le varie fasi del processo.

La correttezza di questo approccio è stata ormai da diversi anni universalmente riconosciuta ed approvata dal mondo dell'ingegneria del software [1].

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 5 di 19
--------------------	----------------------------	--	-----------------	----------------

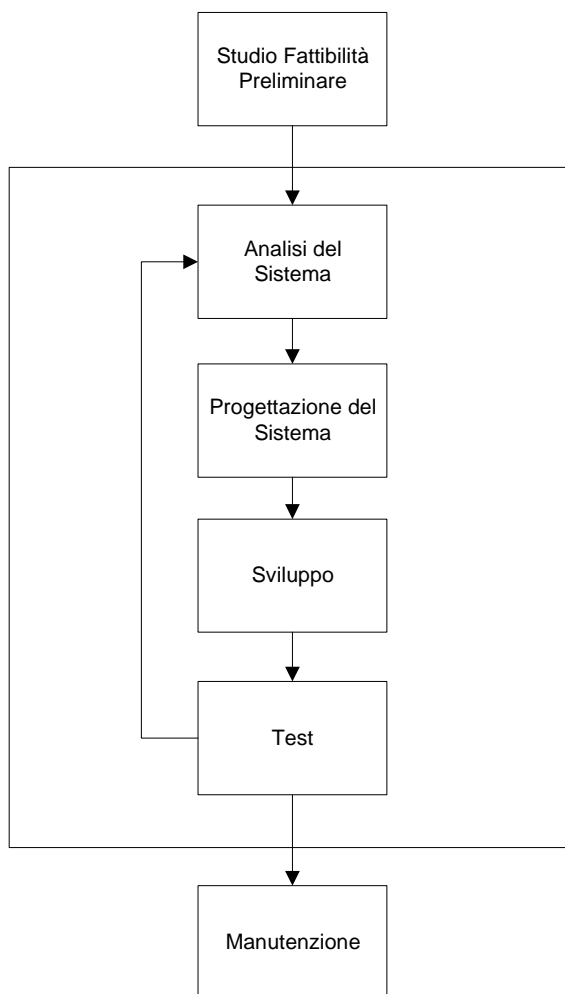


Figura 1- Le fasi del processo adottato da FadeOut Software

Presentiamo velocemente, prima di entrare nel dettaglio, le fasi del processo:

Studio di Fattibilità preliminare: è la fase iniziale del processo. Si attiva al contatto con il cliente ed il suo scopo è comprendere quali siano gli obiettivi dello stesso per la commessa e le caratteristiche salienti del sistema da implementare;

Analisi del Sistema: è la fase in cui si estrapola dal cliente quali siano i reali vincoli e requisiti del sistema definendone quindi il dominio e le funzionalità nel dettaglio;

Progettazione del Sistema: in questa fase vengono utilizzate tecniche di ingegneria per progettare un sistema che sia in grado di implementare tutti i requisiti emersi dall’analisi;

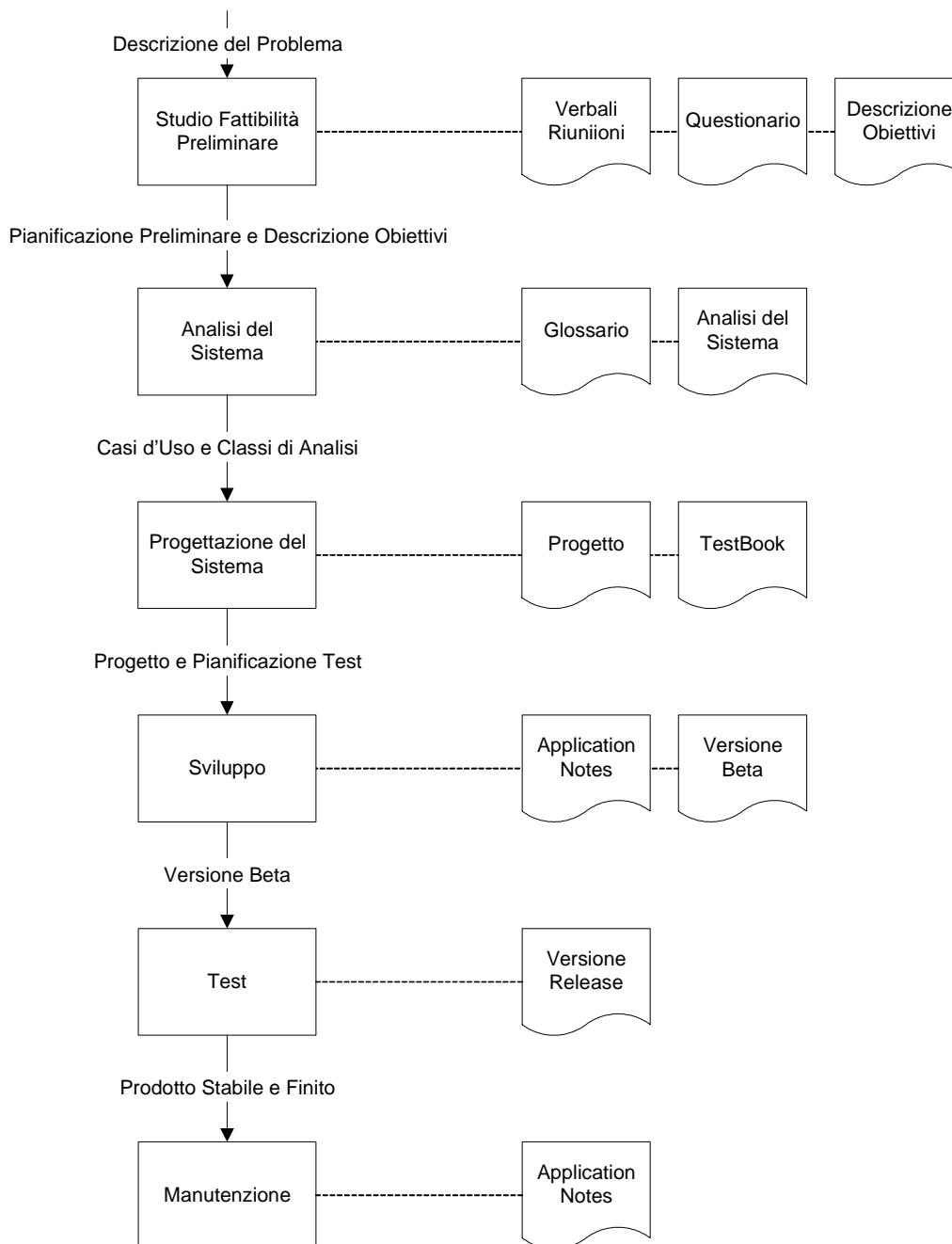
Sviluppo: è lo sviluppo reale del sistema e quindi la realizzazione del sistema progettato precedentemente. Al termine dello sviluppo viene rilasciata la prima versione del prodotto;

Test: il sistema viene provato intensamente in modo da renderlo il più possibile esente da errori o anomalie. Scopo del test è arrivare al corretto collaudo del sistema in modo da rilasciarne la versione definitiva;

Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 6 di 19

Manutenzione: nel caso in cui si verificassero anomalie dopo il rilascio del prodotto, queste vengono analizzate e risolte ed il prodotto viene aggiornato di conseguenza;

Ognuna di queste fasi produce artefatti quali diagrammi, documenti o componenti software, ed è soggetta ad una interazione con il cliente che può vedere il sistema crescere nel tempo, potendo quindi immediatamente notare e risolvere eventuali incomprensioni.



Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 7 di 19

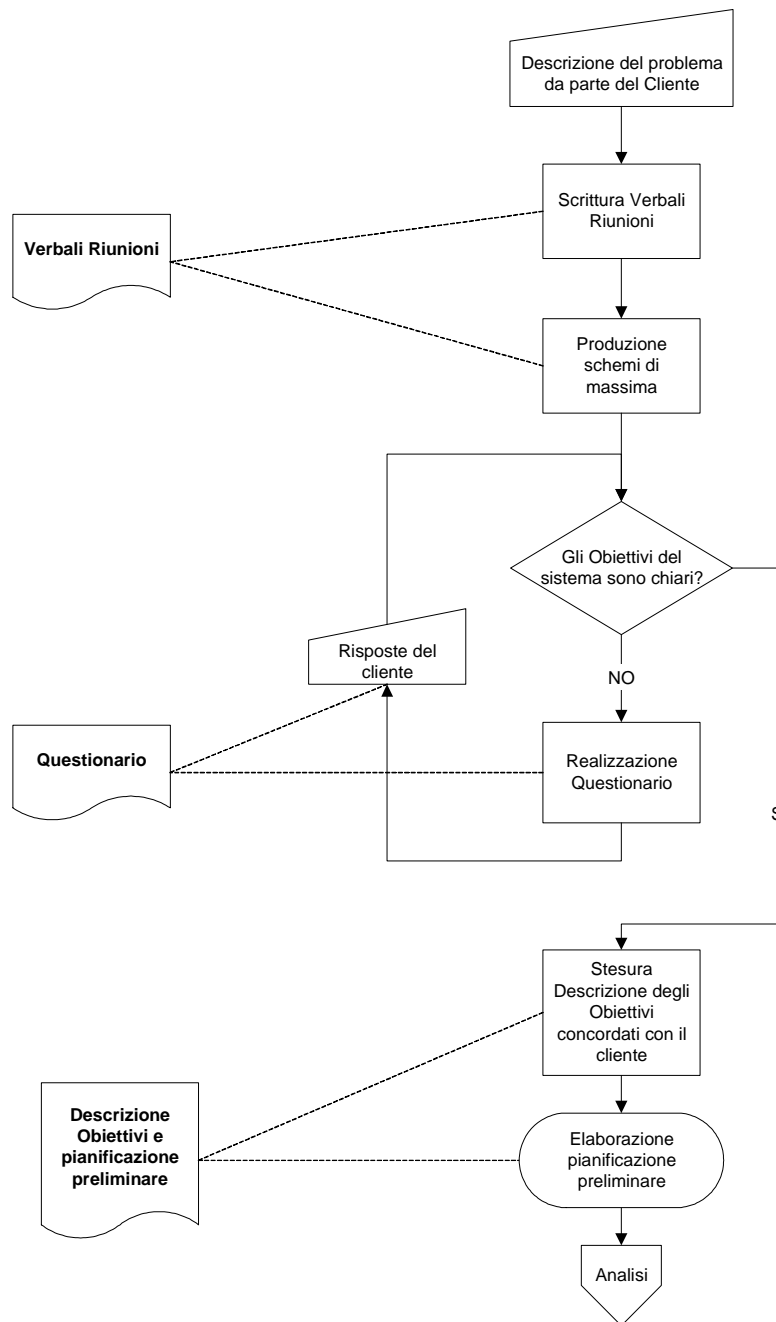
2. Studio di Fattibilità preliminare

Lo studio di fattibilità preliminare comincia con la descrizione, da parte del cliente, delle esigenze che hanno portato alla richiesta ed ad una prima definizione del problema da risolvere. Questo avviene durante varie riunioni, al termine delle quali viene redatto un verbale che riporta quanto discusso e che è condiviso fra il fornitore ed il cliente.

Durante questa fase sono utilizzati al più possibile schemi "logici" del problema e della soluzione in modo da facilitare lo scambio di informazioni.

Vengono quindi analizzati e definiti gli obiettivi del sistema, la cui perfetta comprensione è fondamentale per garantire il successo del progetto complessivo; per questo viene redatto un questionario che viene sottoposto al cliente le cui risposte sono verbalizzate, e che nella pratica risulta essere estremamente utile per focalizzare le caratteristiche e gli obiettivi del sistema.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 8 di 19
--------------------	----------------------------	--	-----------------	----------------



3. Analisi del Sistema

Definiti gli obiettivi del progetto devono essere dettagliate le funzionalità del sistema: i progettisti si concentrano sulle stesse (ponendosi domande quali Esistono possibili eccezioni? Ci sono vincoli impliciti?...) e proseguono la compilazione del questionario in modo da fissare e comprendere a fondo il problema e le richieste del cliente. In questa fase viene redatto il glossario di progetto: ogni dominio di problema ha un suo linguaggio specifico; lo scopo primario dell’ingegneria dei requisiti e dell’analisi è proprio quello di comprendere e fissare quel linguaggio [2].

Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 9 di 19

Appena si gode di una conoscenza sufficiente del sistema comincia l'individuazione degli attori e la modellazione dei casi d'uso.

Gli attori identificano il ruolo che un'entità esterna (es. una persona, un altro sistema software, un macchinario esterno) assume quando interagisce direttamente con il sistema in oggetto.

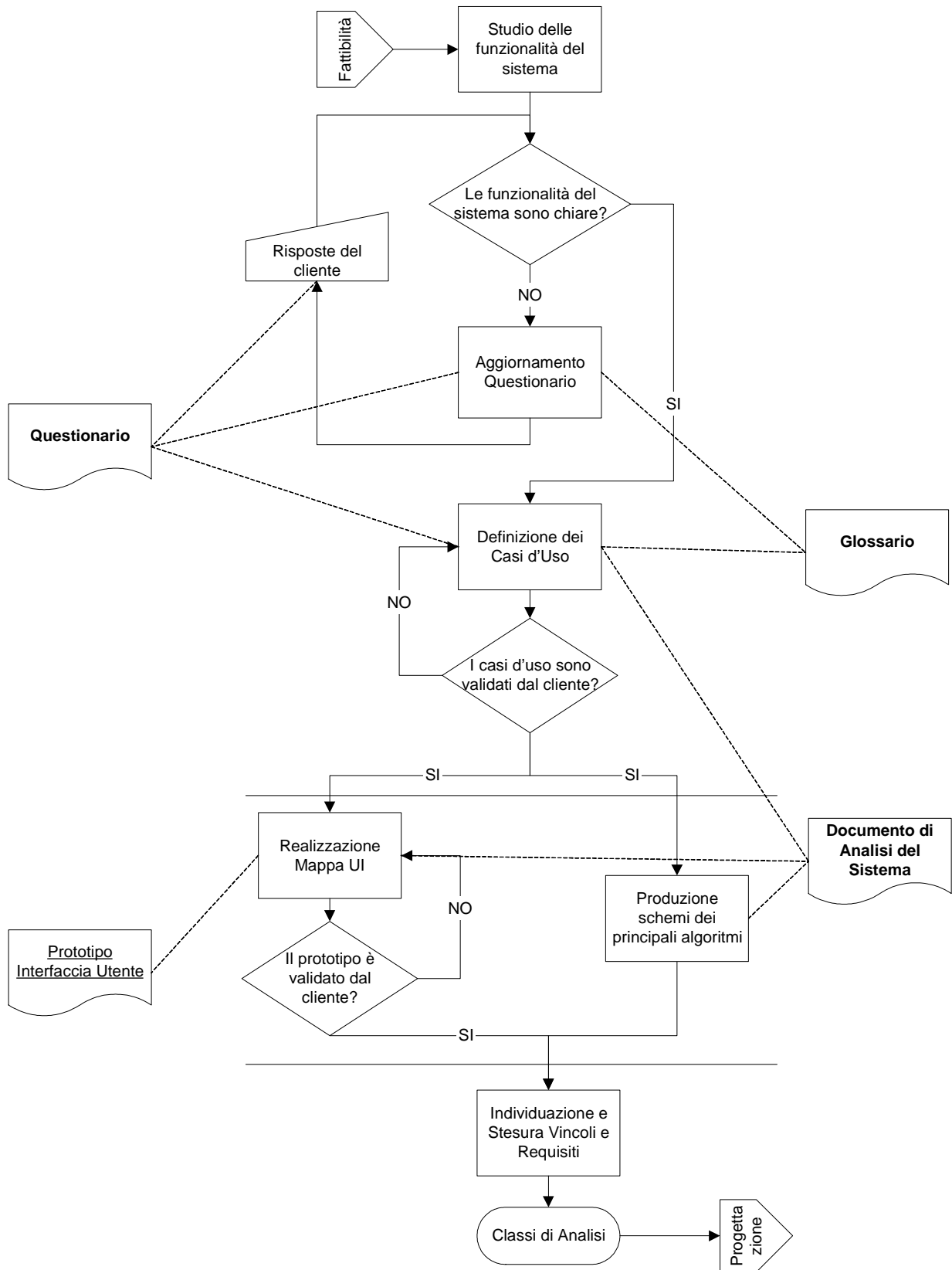
Un caso d'uso è definito come "La specifica di una sequenza di azioni, incluse eventuali sequenze alternative e sequenze di errore, che un sistema, un sottosistema o una classe può eseguire interagendo con attori esterni" [3]. Un caso d'uso è qualcosa che un attore (il cliente) vuole che il sistema faccia; una specie di manuale di istruzioni di come usare il sistema prima di avere lo stesso.

I casi d'uso sono letti insieme al cliente in modo che se ne possa verificare la validità; in questa fase subiscono correzioni ed approfondimenti fino a modellare esattamente ciò che il cliente preferisce.

Per completare l'analisi non resta che realizzare, sulla base dei casi d'uso, un prototipo completo dell'interfaccia utente che permetta al cliente di vedere ed eventualmente correggere il prodotto finale come esso apparirà ed individuare e formalizzare i principali algoritmi, quelli eventualmente critici per il funzionamento del sistema.

Il risultato principale dell'analisi è l'elenco dei vincoli e dei requisiti del sistema che sono emersi dalla stessa; il successo di un progetto infatti dipende dalla capacità dei membri del team di progettazione e dei clienti di condividere una visione chiara degli scopi e dei requisiti di progetto [4].

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 10 di 19
--------------------	----------------------------	--	-----------------	-----------------



Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 11 di 19

4. Progettazione

La fase di progettazione ha naturalmente un ruolo cruciale: quello di tradurre i requisiti nell'architettura del sistema che verrà poi sviluppato con tecniche di programmazione OOP (Object Oriented Programming o Programmazione Orientata agli Oggetti); tale fase è completamente basata sui risultati dell'analisi.

I casi d'uso infatti permettono di individuare una bozza dell'architettura delle classi, cioè i modelli degli oggetti che costituiranno il sistema software. Analizzando quest'ultima e confrontandola con i requisiti si arriva a stendere la gerarchia delle classi ed i diagrammi di associazione, che definiscono quali siano le interazioni che intercorrono tra gli oggetti stessi.

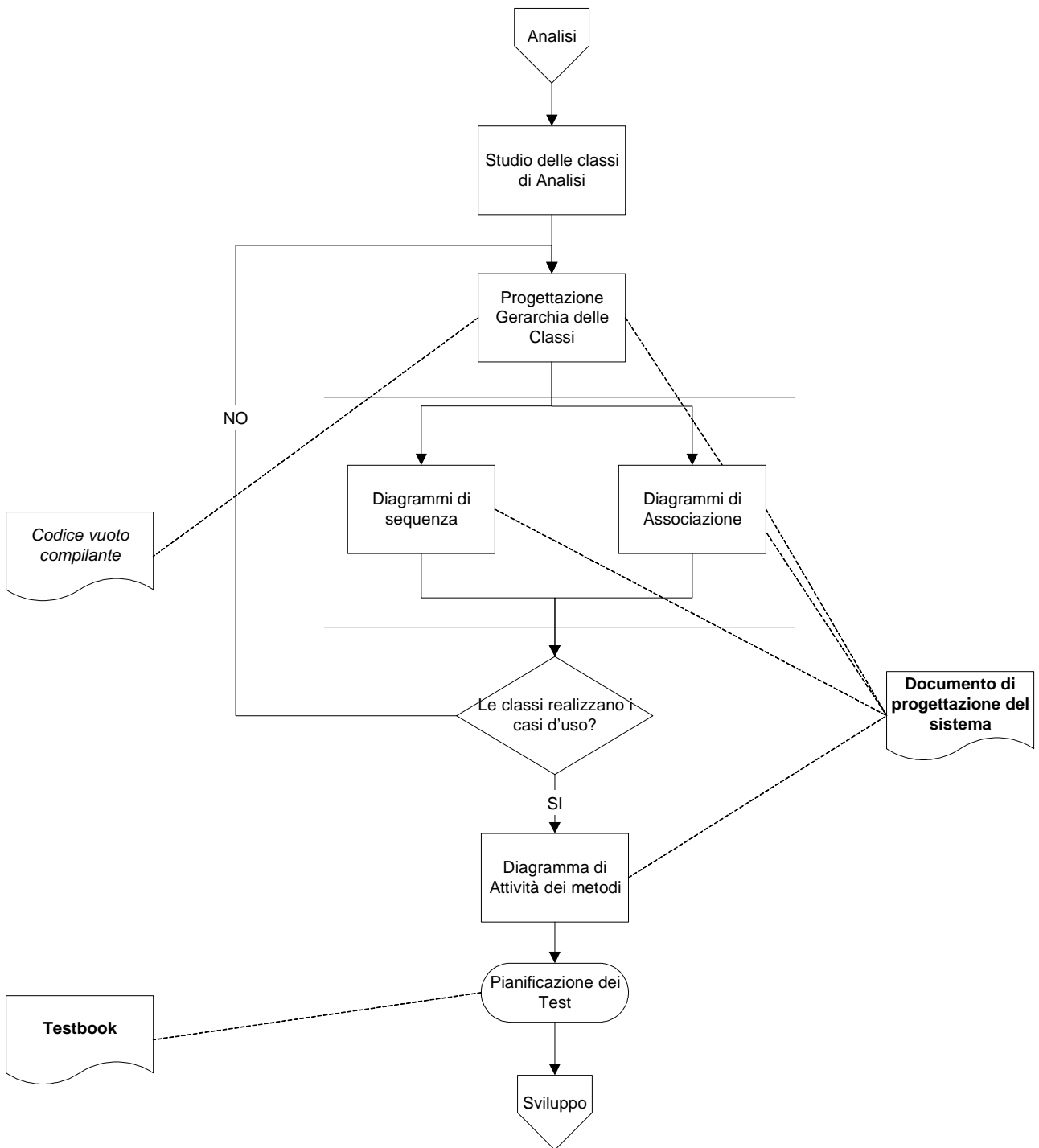
Si passa quindi ai diagrammi di sequenza, in grado di ordinare nel tempo il flusso delle chiamate, cioè di descrivere come i diversi oggetti collaborino per realizzare i requisiti. Si verifica iterativamente che l'architettura progettata sia in grado di realizzare i casi d'uso riportati nel documento di analisi per assicurarsi che vengano rispettate tutte le funzionalità emerse.

Durante tutto il processo, ma in particolare nella fase di progetto, viene utilizzato UML, che può essere definito come segue: "l'Unified Modelling Language è un linguaggio visuale di modellazione dei sistemi.[...] L'UML è stato studiato per incorporare tutte le migliori pratiche di modellazione e di ingegneria del software correntemente utilizzate." [2].

Quando l'architettura risulta stabile e completa, il progettista dettaglia per lo sviluppatore i diagrammi di attività (diagrammi di flusso) di tutti i metodi del software e pianifica i test che andranno eseguiti sul sistema.

Si passa quindi allo sviluppo, consegnando il TestBook, che raccoglie tutti i test previsti, ed il documento di progetto.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 12 di 19
--------------------	----------------------------	--	-----------------	-----------------



5. Sviluppo

Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 13 di 19

Se analisi e progettazione sono necessarie alla realizzazione di un prodotto di qualità queste non sono comunque sufficienti. Anche durante le fase di sviluppo devono essere seguite alcune linee guida che consentano di ottenere:

- la leggibilità del codice;
- la possibilità di una facile manutenzione;
- la maggior riduzione possibile degli errori;
- la gestione delle eccezioni, ossia delle situazioni di errore;
- la tracciabilità delle operazioni eseguite dal sistema;

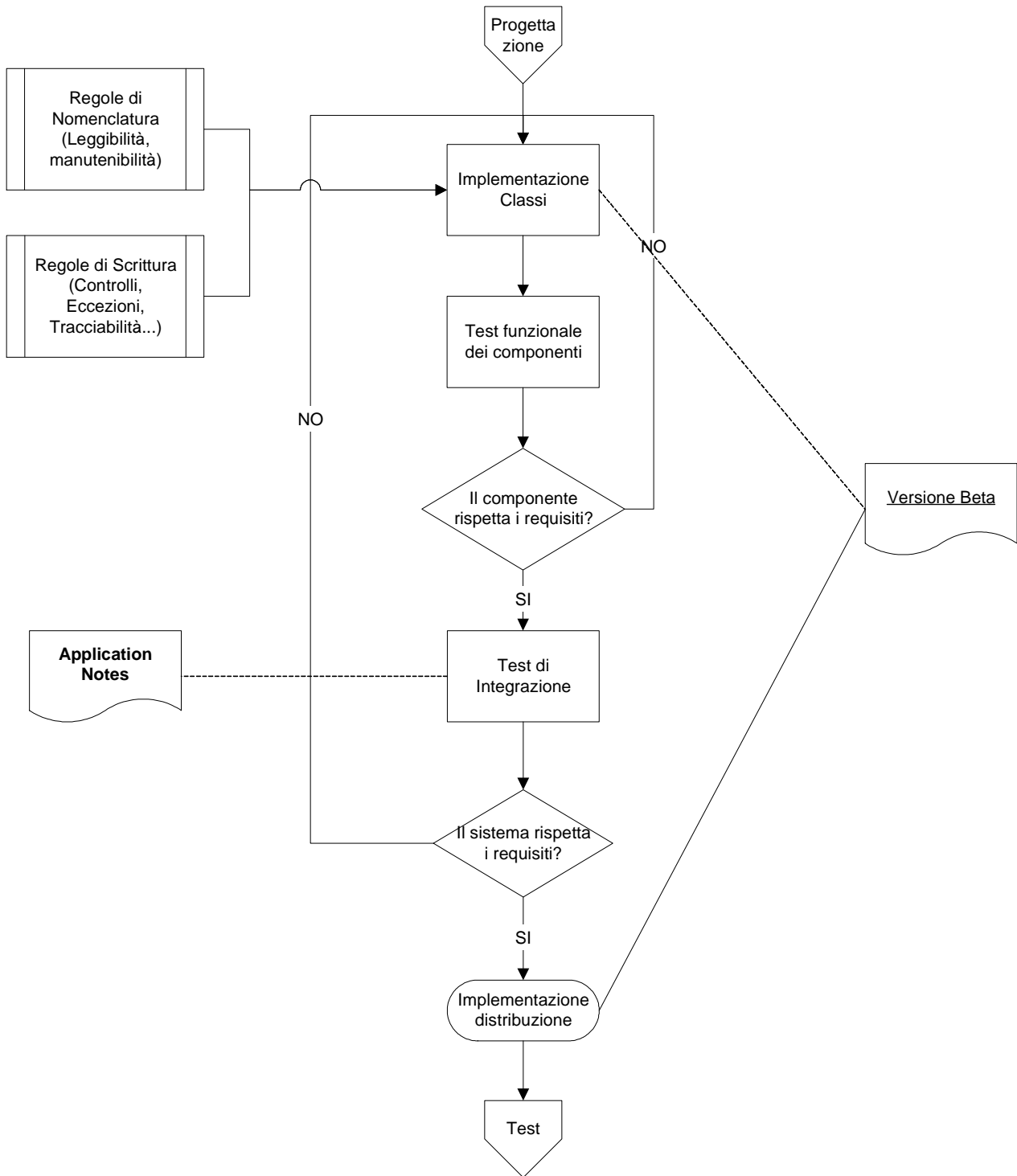
Per assicurare questi risultati dovranno essere curate una serie di regole predefinite di nomenclatura e scrittura del codice che saranno applicate a qualunque software prodotto. Queste regole permettono l'immediata comprensione del codice scritto indipendentemente da chi lo abbia sviluppato ed il controllo e la gestione di errori ed eccezioni.

Applicando queste regole viene implementato il codice progettato; il sistema viene provato componente per componente assicurandosi di non avere trascurato i requisiti funzionali, o di non averli realizzati in modo errato.

Al termine dell'implementazione vengono eseguiti i test di integrazione dei componenti, analizzando il comportamento del sistema nel suo insieme, applicando le dovute correzioni al codice e ripetendo di conseguenza tutti i test necessari.

La fase di sviluppo termina con il rilascio della prima versione del sistema, usualmente chiamata versione Beta, condivisa con il cliente, che realizza tutte le funzionalità richieste ma deve essere stabilizzata. Spesso vengono anche prodotte Application Notes che mostrano e focalizzano particolarità del sistema eventualmente emerse durante il lavoro.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 14 di 19
--------------------	----------------------------	---	-----------------	-----------------



Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 15 di 19

6. Test

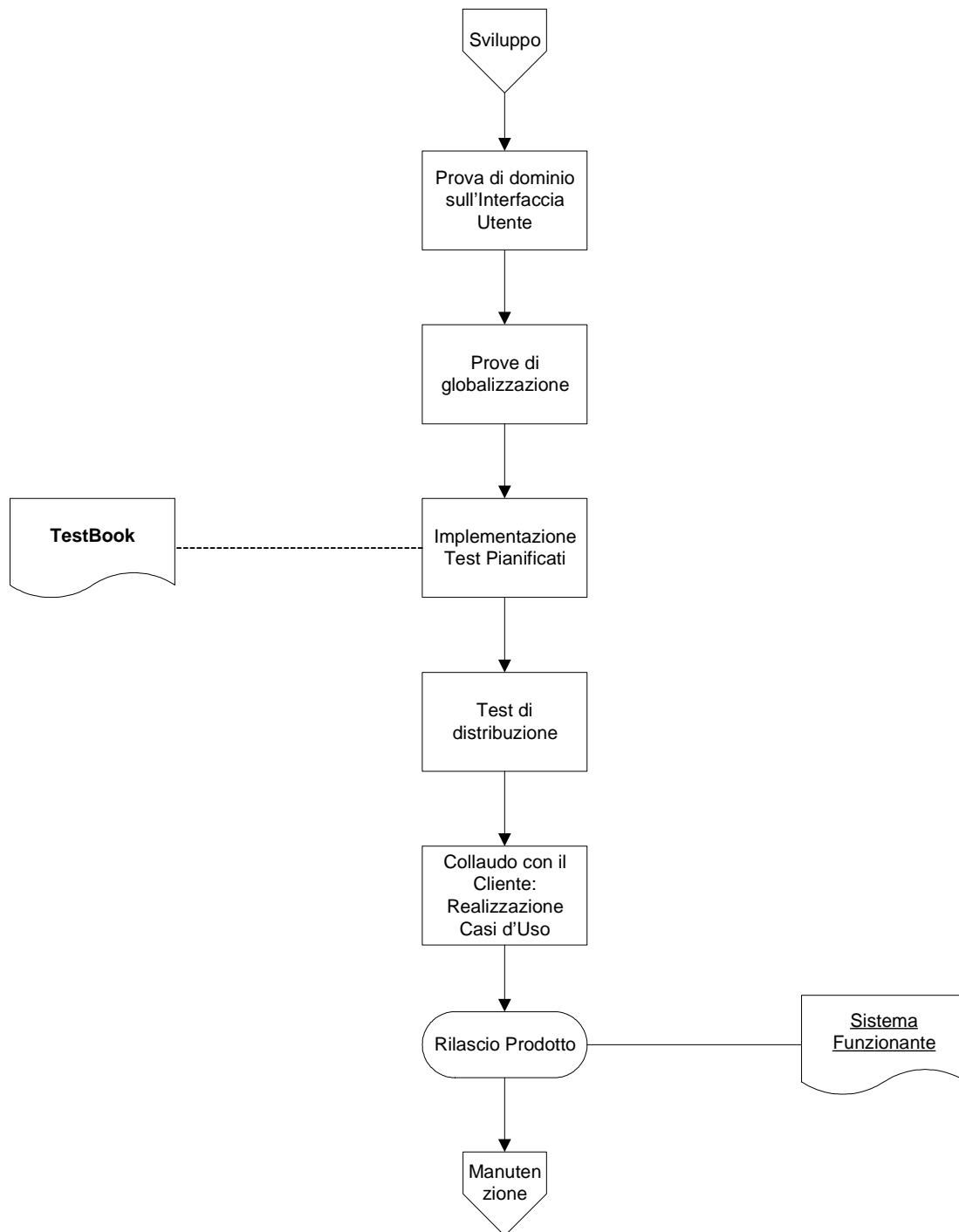
“Errare è umano, recuperare gli errori è compito dell’ingegneria del software” [5]; questa frase caratterizza la natura dello sviluppo software, che nonostante gli accorgimenti presi non può mai ipotizzarsi esente da errori. Lo scopo della fase di Test è proprio quello di eliminare tali errori e di stabilizzare definitivamente il sistema. Anzitutto si sfrutta la mappa dell’interfaccia utente per provarne tutti i percorsi e si verifica, quando necessario, il supporto alle caratteristiche di globalizzazione.

Vengono poi realizzati tutti i test pianificati nel TestBook e ne viene documentato il risultato e le eventuali correzioni apportate.

Una volta che il sistema è stabile viene provata la distribuzione dello stesso nelle varie configurazioni e piattaforme su cui potrà o dovrà essere ospitato.

La fase di test si conclude quindi con il collaudo presso il cliente: si tratta di una riunione in cui viene installato il sistema e si realizzano con il cliente tutti i casi d’uso del documento di analisi per assicurarsi che non vi siano eventuali problemi o incomprensioni. Superato il collaudo il progetto è concluso ed il sistema è rilasciato.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 16 di 19
--------------------	----------------------------	--	-----------------	-----------------

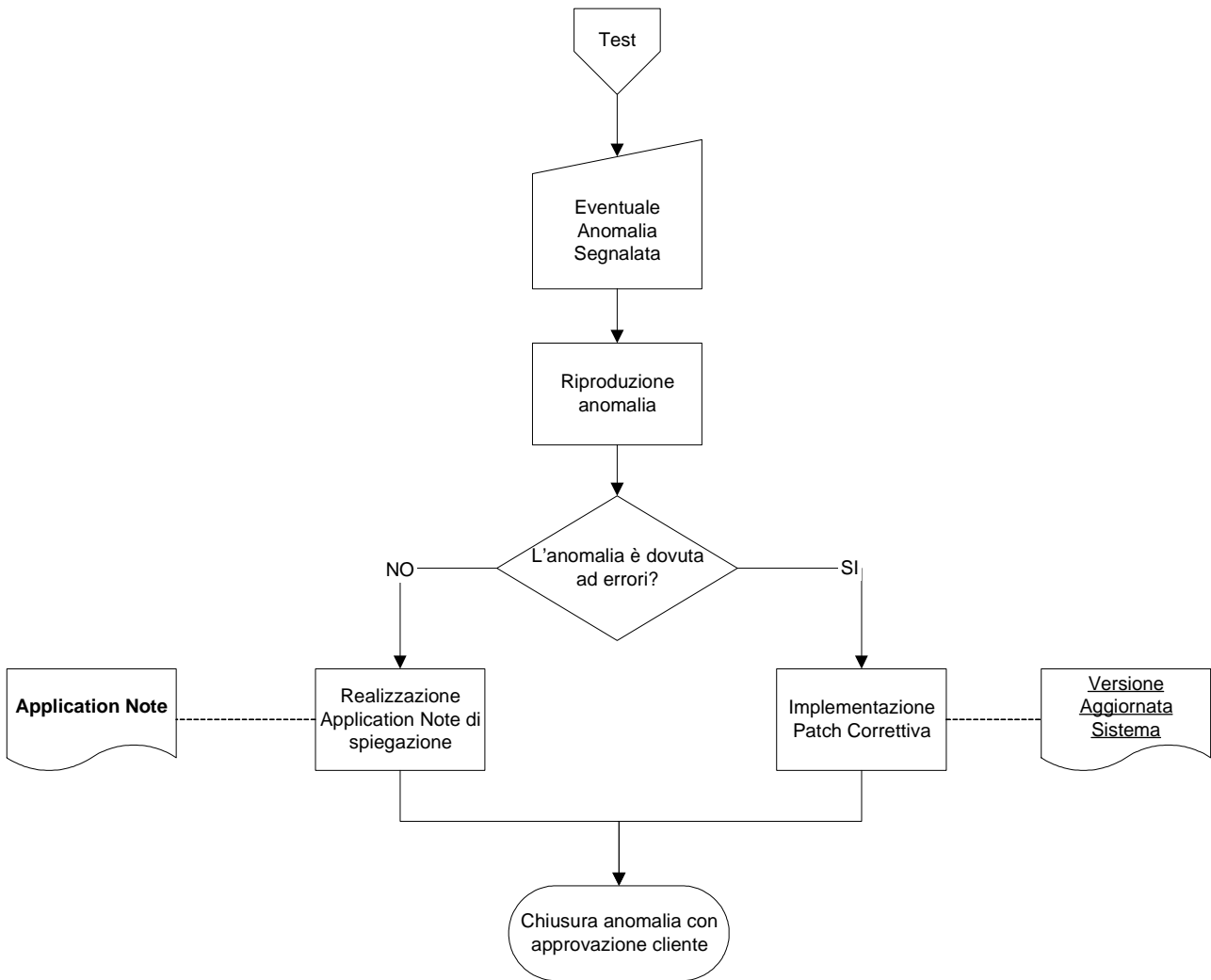


7. Manutenzione

Non esiste un buon servizio al cliente che non preveda l'assistenza: quando, ad esempio, un software è destinato alla rivendita il numero di combinazioni di funzionamenti e configurazioni su cui girerà il software potrebbe fare emergere anomalie non individuate in precedenza. Il fornitore si deve fare carico di risolvere

Data	Autore	Progetto	Versione	
02/04/2007	Campanella;Tasso	La Progettazione del Software	1.1	Pagina 17 di 19

questi eventuali problemi per i propri clienti al fine di garantire la qualità promessa. Questo non solo è corretto, ma soprattutto il miglior modo di verificare eventuali imperfezioni per studiarle e mettere a punto strategie che le impediscano in futuro.



8. Conclusioni

Attenersi a questo processo è indubbiamente sostenibile anche per un singolo professionista: tutto ciò di cui si necessita è una suite software di office automation facilmente reperibile a costi accessibili sul mercato e qualche modello di documento predefinito. I risultati diventano evidenti anche per progetti della durata complessiva di un mese: diminuiscono i tempi di sviluppo e test, scompaiono i cambi di funzionalità a sviluppo in corso, aumenta la riusabilità del codice e diminuiscono gli interventi di manutenzione.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 18 di 19
--------------------	----------------------------	---	-----------------	-----------------

La qualità del lavoro è percepita dal cliente grazie a diversi aspetti: anzitutto nella fase di analisi lo stesso cliente ha occasione di apprezzare la funzionalità dei casi d'uso, chiarendosi le operazioni che vorrà vedere eseguite dal software.

Grazie al prototipo il cliente vede subito, potendo quindi valutarlo e correggerlo, l'aspetto finale del prodotto: dallo stesso prototipo emergono anche sempre eventuali incomprensioni che possono essersi verificate, il tutto prima di mettere mano al progetto.

Grazie alla progettazione ed alle regole di sviluppo diminuiscono di molto i crash della versione beta, che tipicamente, seppur giustificati, macchiano la percezione di qualità da parte del cliente.

Le application notes fissano e prevenegono eventuali comportamenti anomali.

Il collaudo permette al cliente di provare il prodotto completo prima della chiusura della commessa.

Per concludere non resta che notare come sia comunque insostituibile la capacità personale del progettista di modellare del sistema per distinguere un lavoro ben organizzato da una brillante soluzione.

9. Bibliografia

- [1] Ivar Jacobson, Grady Booch, James Rumbaugh "Unified Software Development Process" Addison-Wesley, 1999.
- [2] Jim Arlow, Ila Neustadt, "UML e Unified Process" McGraw-Hill 2003.
- [3] James Rumbaugh, Ivar Jacobson, Grady Booch "The Unified Modeling Language Reference Manual", Addison-Wesley, 1998.
- [4] AA.VV., "Requisiti ed architettura delle soluzioni Microsoft .NET", Microsoft Press - Mondadori Informatica, 2003.
- [5] Walter Oney, "Programming Microsoft WDM", Microsoft Press 2003.

Data 02/04/2007	Autore Campanella;Tasso	Progetto La Progettazione del Software	Versione 1.1	Pagina 19 di 19
--------------------	----------------------------	--	-----------------	-----------------