
PROGETTO SIGEOL

Piano di Qualifica

v1.1.0

Redazione: Carlo Scortegagna

21 gennaio 2009



quixoft.sol@gmail.com

Verifica:	Freo Matteo
Approvazione:	Grosselle Alessandro
Stato:	Formale
Uso:	Esterno
Distribuzione:	QuiXoft
	Rossi Francesca
	Vardanega Tullio
	Conte Renato

Sommario

Piano di Qualifica per il progetto "SIGEOL", necessario per regolamentare le operazioni di verifica, validazione e controllo qualità durante il periodo di sviluppo.



Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
2	Visione generale	1
2.1	Organizzazione	1
2.2	Obiettivi	2
2.3	Caratteristiche del prodotto	2
2.4	Responsabilità	3
2.5	Pianificazione	4
2.6	Risorse	5
2.7	Tecniche e metodi di verifica	5
2.7.1	Verifica del codice prodotto	5
2.7.2	Verifica della documentazione	6
2.7.3	Ambienti di prova e di collaudo	6
3	Automazione delle attività di verifica	7
3.1	Tickets e consultazione ore di lavoro	7
3.2	Validazione delle pagine web	7
3.3	TDD e Mocks objects	7
3.4	Metriche sulla copertura dei test sul codice	8
4	Gestione amministrativa	8
4.1	Comunicazioni	8
4.2	Risoluzione anomalie	9
4.3	Trattamento delle discrepanze	9

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è di definire e pianificare le procedure che il team QuiXoft adotterà per garantire la qualità del prodotto denominato "SIGEOL". Requisito fondamentale per garantire un adeguato livello qualitativo è il rispetto delle specifiche definite in fase di analisi del progetto.

Particolare attenzione verrà posta anche nella definizione delle attività di verifica e validazione del materiale prodotto e, col procedere dello sviluppo, nella descrizione degli ambienti di prova e di collaudo.

1.2 Scopo del prodotto

Il prodotto "SIGEOL" si prefigge di automatizzare la generazione, la gestione, l'ottimizzazione e la consultazione degli orari di lezione.

Per ulteriori informazioni riguardanti scopi e funzioni del prodotto si prega di fare riferimento al documento ANALISI DEI REQUISITI.

1.3 Glossario

Le definizioni dei termini specialistici usati nella stesura di questo e di tutti gli altri documenti possono essere trovate nel documento GLOSSARIO al fine di eliminare ogni ambiguità e di facilitare la comprensione dei temi trattati. Ogni termine la cui definizione è disponibile all'interno del glossario verrà marcato con una sottolineatura.

1.4 Riferimenti

- Capitolato d'appalto reperibile all'indirizzo:
<http://www.math.unipd.it/tullio/IS-1/2008/Progetti/SIGEOL.html>
- documento ANALISI DEI REQUISITI alla sua ultima versione
- NORME DI PROGETTO
- standard ISO/IEC 9126:2001

2 Visione generale

2.1 Organizzazione

Le attività finalizzate a garantire la qualità del prodotto "SIGEOL" inizieranno in concomitanza con l'inizio della fase di analisi del progetto, per garantire immediatamente l'assenza di errori o incongruenze.

Il controllo della qualità continuerà quindi per tutta la durata del progetto, sino alla consegna ed all'accettazione del prodotto finito. Durante tale

periodo di tempo sarà compito dei membri del team QuiXoft aggiornare il presente documento con i risultati aggiornati dei test e delle verifiche a cui il progetto verrà sottoposto. Al momento della consegna del prodotto, sarà compito del committente valutarne le conclusioni ed esaminare se i risultati rispetteranno appieno i requisiti stilati durante l'iniziale fase di analisi.

Nelle fasi successive all'accettazione del prodotto la gestione qualità non sarà più compito del team QuiXoft: per ulteriori informazioni sulla fase di manutenzione si prega di consultare il documento PIANO DI PROGETTO.

Qualsiasi risultato prodotto dal team QuiXoft dovrà essere sottoposto a verifica, sia esso un documento ufficiale, un file di codice sorgente o un qualsiasi altro risultato del lavoro del team.

Vista la natura del team di sviluppo, in cui tutti i componenti occuperanno a rotazione tutti i ruoli necessari, particolare attenzione sarà posta nel garantire l'assenza di conflitti di interesse: nessun Verificatore dovrà quindi trovarsi nella posizione di valutare un documento o un file sorgente prodotto da lui stesso in una precedente fase del progetto.

2.2 Obiettivi

Il lavoro del team, dalla fase di consegna del capitolato all'accettazione del prodotto finale, dovrà essere svolto nell'ottica di ottenere un risultato che soddisfi le esigenze del committente. Sarà responsabilità di tutti i componenti ottenere un prodotto finale che soddisfi pienamente tutti i requisiti esposti.

L'obiettivo di ogni componente del team QuiXoft è di completare le proprie mansioni nel miglior modo possibile, di rispettare le NORME DI PROGETTO, di restare fedele a quanto prestabilito nel PIANO DI PROGETTO per quanto riguarda le ore massime di impegno personale ed il tetto massimo di spesa per portare a compimento l'intero progetto.

Ogni attività del team deve essere svolta puntando all'obiettivo finale di realizzare un prodotto di qualità, finalizzato al soddisfacimento:

- **del committente**, realizzando un prodotto che soddisfi pienamente i requisiti, che sia funzionale, usabile e pratico, che sia affidabile, efficiente e sicuro;
- **del team stesso**, realizzando un progetto qualitativamente alto che dia la prospettiva di sviluppi futuri, che possa essere riusato facilmente, che sia facilmente manutenibile, che sia portabile.

2.3 Caratteristiche del prodotto

Il risultato dal lavoro del team QuiXoft dovrà il più possibile aderire ai parametri di qualità descritti nello standard ISO/IEC 9126:2001. Le princi-



2 VISIONE GENERALE

pali caratteristiche che il prodotto dovrà garantire per essere considerato di qualità sia dal committente sia dal team stesso saranno:

- **Funzionalità:**
 - Utilità: fare esattamente quello per cui è stato progettato
 - Accuratezza: funzionare rispettando i requisiti
 - Interoperabilità: interagire facilmente con l'esterno
 - Conformità: conformarsi a norme e standard
 - Sicurezza: impedire accessi non autorizzati
- **Affidabilità:**
 - Maturità: garantire una bassa frequenza di fallimenti dovuti a errori
 - Tolleranza ai guasti: mantenere le prestazioni prefissate in caso di errori
 - Ripristinabilità: ristabilire i dati perduti in caso di errori
- **Usabilità:**
 - Comprensibilità: poco sforzo necessario per comprenderne l'uso
 - Apprendimento: facilitare la comprensione dell'utente
 - Operabilità: agevolare l'utente nel controllo del software
- **Efficienza:**
 - Rispetto dei tempi di risposta e di esecuzione
 - Rispetto delle risorse utilizzate
- **Manutenibilità:**
 - Analizzabilità: facilitare la diagnostica degli errori
 - Modificabilità: agevolare la manutenzione futura
 - Stabilità: rischi controllati in caso di modifica
 - Verificabilità: validazione semplificata dopo una manutenzione
- **Portabilità:**
 - Adattabilità: adattamento ad ambienti diversi senza modifiche aggiuntive
 - Installabilità: permette l'installazione in uno specifico ambiente
 - Conformità: aderire a standard, norme e convenzioni di portabilità
 - Sostituibilità: sostituire elementi dell'ambiente esterno

Nel caso di sviluppo di prodotti che possiedono un'interfaccia via pagina web, sarà altresì considerata caratteristica importante la validazione del codice tramite gli strumenti messi a disposizione dal consorzio W3C. Per maggiori informazioni si può visitare il sito <http://www.w3.org>.

2.4 Responsabilità

Ogni componente del team QuiXoft è tenuto a svolgere il lavoro a lui assegnato con precisione, puntando a renderlo il più possibile esente da errori. Primo requisito per tenere alto il livello di qualità dei lavori svolti è quindi di

leggere con attenzione le NORME DI PROGETTO, e di applicarne alla lettera il contenuto.

Nonostante ciò, è ovviamente sempre presente una percentuale di incertezza sulla perfezione del lavoro svolto, ed è proprio compito del presente documento illustrare le procedure da seguire in caso di errori, imprecisioni o incongruenze, al fine di ottenere il maggior livello qualitativo possibile.

Le figure chiave per la gestione della qualità all'interno del team sono:

- **Verificatore**: è colui che effettivamente controlla ed esegue i test sul materiale prodotto dai membri del team. Ogni documento, file o prodotto passa necessariamente attraverso le sue verifiche, e senza il suo responso positivo nulla può essere dichiarato ufficiale. Se durante una fase del progetto vi fosse la necessità di avere più di un verificatore, sarà compito del Responsabile eleggere un rappresentante che coordini le loro attività interne;
- **Responsabile**: suo compito è di convalidare le modifiche proposte dai Verificatori e di controllarne l'operato;
- tutti gli altri ruoli previsti all'interno del team (Analista, Progettista, Programmatore, Amministratore) sono tenuti a sottoporre il proprio lavoro al Verificatore per le fasi di controllo e test. E' loro responsabilità anche apportare le modifiche che il Verificatore avrà segnalato essere necessarie e che il Responsabile avrà convalidato.

2.5 Pianificazione

Come detto in precedenza, il controllo qualità parte già dall'inizio del progetto: anche le attività di analisi dei requisiti e di studio di fattibilità necessitano della verifica da parte di qualcuno estraneo alla redazione, per evitare errori o incongruenze. Tali attività di controllo del lavoro svolto continuano fino alla consegna del prodotto finale.

Durante lo sviluppo del progetto saranno svolte delle revisioni per saggiarne lo stato di salute. In queste occasioni sarà possibile valutare il livello di qualità del lavoro svolto.

I processi di revisione previsti saranno di due tipi:

1. Revisioni formali:

- revisione dei requisiti (RR)
- revisione di accettazione (RA)

2. Revisione informali:

- revisione del progetto preliminare (RPP)
- revisione di qualifica (RQ)

Le revisioni formali verranno svolte in presenza del committente: uno scarso livello qualitativo del prodotto in questo caso comprometterebbe l'assegnazione (nel caso della RR) o l'approvazione (nel caso della RA) del prodotto, vanificando gli sforzi del team. Le revisioni formali, infatti, hanno effetto sanzionatorio.

Le revisioni informali invece verranno svolte in presenza del docente di riferimento, e avranno scopo puramente informativo: nel caso di qualità inadeguata, quest'ultima verrà solamente notificata e il team potrà recuperare gli eventuali errori fatti.

Per la pianificazione oraria delle attività di verifica durante le varie fasi del progetto e per le date delle revisioni si prega di consultare il documento PIANO DI PROGETTO.

2.6 Risorse

La gestione qualità dell'intero progetto impegna risorse umane come tecnologiche, ed è compito dell'Amministratore coordinarne l'utilizzo.

Una stima dei tempi che i Verificatori dedicheranno al controllo della qualità è consultabile sul PIANO DI PROGETTO. Nonostante ciò, non è da escludere che tali ipotesi di impegno siano inadeguate. Il livello di carico infatti varierà in base all'onerosità e alla tipologia del processo in esame, ed è arduo preventivare con precisione i tempi necessari per assicurare un corretto e preciso svolgimento delle attività di verifica.

Sarà compito dell'Amministratore gestire l'uso delle risorse umane e tecnologiche per garantire che siano sufficienti ed efficienti per consentire una verifica di buon livello.

2.7 Tecniche e metodi di verifica

La qualità dei processi è critica per la qualità del prodotto, quindi il team QuiXoft si impegna a garantire e migliorare la qualità dei propri processi.

Per far ciò il prodotto che si intende sviluppare, nonché ogni processo istanziato, saranno sottoposti a costante attività di verifica in ogni fase di lavoro descritta nel PIANO DI PROGETTO. Di seguito sono riportate le metodologie per attuare tale attività.

2.7.1 Verifica del codice prodotto

L'intero sviluppo del sistema "SIGEOL" sarà guidato dalla metodologia *Test-Driven Development* (d'ora in poi TDD).

La strategia di sviluppo TDD enfatizza in modo formidabile i test automatizzati per verificare la correttezza del codice. Invece di costruire una struttura di test alla fine dello sviluppo dell'applicazione, il TDD prevede che la scrittura del test sia la fase iniziale del processo. Quindi l'implementazione di una nuova funzionalità viene preceduta dalla definizione dei test

che ne stabiliscono la correttezza formale e semantica. Questo approccio garantisce che il codice prodotto risulti naturalmente modulare ed i moduli possano essere testati in isolamento gli uni dagli altri.

Il membri del team QuiXoft dovranno quindi, nella fase denominata *Progettazione*, focalizzare la loro attenzione sulla scelta di un linguaggio di programmazione, un framework ed un insieme di plugin che facilitino ed incoraggino l'uso di una tale strategia. Sarà poi compito degli addetti alla fase denominata *Realizzazione* aderire il più possibile a questa metodologia.

Le diverse tipologie di test sono le seguenti:

- **Test di unità:** ha lo scopo di verificare ogni unità software affinché soddisfi i requisiti previsti mediante l'impiego di un insieme di dati in ingresso;
- **Test di integrazione:** ha lo scopo di verificare la cooperazione di più unità software;
- **Test di sistema:** ha lo scopo di verificare il soddisfacimento dei requisiti di tutto il sistema, in relazione anche all'ambiente d'installazione.

Sarà compito del Verificatore accertarsi della corretta applicazione di tale metodologia ed inoltre sarà incaricato di effettuare un'analisi del codice per evitare un'eccessiva ridondanza od una complessità elevata sia in termini di manutenibilità che riguardo all'efficienza. A tal scopo si cercheranno di utilizzare strumenti di benchmark che facilitino la misurazione delle performance del sistema.

2.7.2 Verifica della documentazione

L'incaricato alla verifica di ogni documento dovrà accertarsi che gli argomenti trattati siano conformi a quanto presente nel *Sommario* e nella sezione *Scopo del documento* e che il contenuto di ogni sezione rispetti il titolo assegnato alla stessa.

Inoltre dovrà segnalare parti del documento che risultino poco chiare od ambigue, eliminare porzioni di testo ritenute superflue e privare il documento di ogni errore ortografico, grammaticale o dattilografico.

2.7.3 Ambienti di prova e di collaudo

Con il procedere del progetto, come anche specificato nell' ANALISI DEI REQUISITI, si renderà necessario organizzare degli ambienti di prova e di collaudo per il software prodotto, per rilevare errori o inconsistenze nei dati. Nei prossimi rilasci del presente documento ne verranno specificate le caratteristiche e le funzionalità.



3 Automazione delle attività di verifica

La procedura di verifica può risultare spesso tediosa e ripetitiva se svolta in modo completamente manuale, nonchè l'assenza di un'adeguata infrastruttura a supporto di tale attività può essere fonte di errori ed imprecisioni.

Per questo motivo il team QuiXoft utilizza una serie di applicativi che facilitino ed automatizzino il più possibile l'attività di verifica, rendendo disponibili i risultati quando questi saranno necessari.

3.1 Tickets e consultazione ore di lavoro

Quando un compito è assegnato ad un membro tramite un ticket, il Responsabile stimerà le ore di lavoro necessarie per lo svolgimento dell'incarico ed annoterà tale numero nell'apposito spazio presente nel ticket.

Il componente che ha accettato il ticket, una volta completata la propria mansione, registrerà le ore effettive di lavoro che potranno essere superiori alle aspettative del Responsabile. Tramite gli strumenti messi a disposizione dal sito <http://www.assembla.com>, è immediata la consultazione del carico di lavoro e quindi il Responsabile, od un suo delegato, potrà apportare, se lo ritiene necessario, le dovute modifiche al PIANO DI PROGETTO.

3.2 Validazione delle pagine web

L'adesione agli standard proposti dal consorzio W3C nella creazione di pagine web è un elemento fondamentale per la corretta visualizzazione delle stesse attraverso diversi tipi di browser. Per questo motivo il team QuiXoft utilizzerà gli strumenti messi a disposizione da tale consorzio per validare le pagine create.

Questo compito risulta spesso ripetitivo e, se svolto manualmente, produce un'eccessivo consumo di ore di lavoro. Per questo motivo verrà utilizzato un plugin che validerà ogni pagina creata in modo automatico e renderà disponibili immediatamente i risultati di tale validazione per poter eseguire le adeguate correzioni.

3.3 TDD e Mocks objects

Come precedentemente dichiarato, l'intero sviluppo del progetto sarà guidato dalla metodologia Test-Driven Development (confronta sezione 2.7.1). Ogni componente del team, nell'arco di tutta la durata del progetto, dovrà ricoprire ogni ruolo (per maggiori dettagli consultare il documento PIANO DI PROGETTO), quindi ogni membro ha discrete competenze riguardo la stesura del codice. Per questo motivo il Verificatore di turno (o i Verificatori se ne sono previsti più di uno) avrà il compito di sviluppare i test anticipatamente alla stesura del codice. In questo modo il Programmatore avrà il solo compito di implementare le funzionalità che sono richieste dai test.



4 GESTIONE AMMINISTRATIVA

Per facilitare la scrittura di test verrà utilizzato un plugin che permetta la creazione di Mocks object, letteralmente “oggetti di pezza”, in altre parole un oggetto che simula il comportamento di uno reale, permettendo di isolare una componente del sistema dalle altre per poterla testare al meglio. Un mock object non ha solamente metodi per simulare un oggetto reale, ma ha un’implementazione dei metodi propri di tale oggetto nella quale viene verificato il comportamento di chi usa l’oggetto, per esempio verificando che i metodi siano richiamati in un certo ordine o con certi parametri.

Per maggiori informazioni è consigliata la consultazione del sito <http://flexmock.rubyforge.org>.

3.4 Metriche sulla copertura dei test sul codice

Durante l’implementazione da parte del Programmatore delle funzionalità richieste dai test, è possibile che venga aggiunto del codice che non verrà mai testato. A questo scopo sarà utilizzato un plugin che, in modo completamente automatico, esplori il codice sorgente segnalando quelle porzioni che non vengono verificate da nessun test.

Il plugin renderà disponibili le porzioni di codice non testato, nonché la percentuale di codice coperta da test. Sarà quindi auspicabile che tale percentuale raggiunga il 100% prima della consegna del prodotto.

Per maggiori informazioni è consigliata la consultazione del sito <http://rubyforge.org/projects/rcov/>.

4 Gestione amministrativa

4.1 Comunicazioni

Come spiegato nel documento NORME DI PROGETTO, ad ogni prodotto del lavoro del team è assegnato un ticket, consultabile sul sito di appoggio <http://www.assembla.com>. Per prodotto si intende ovviamente qualsiasi risultato del lavoro dei componenti del team QuiXoft (documentazione, codice sorgente, moduli o programmi completi, ecc...).

Al momento del completamento di un ticket accettato dovrà esservi associato un altro ticket che ne richieda la verifica. Il nuovo ticket verrà assegnato all’unico Verificatore nel caso ce ne sia solo uno, oppure al rappresentante eletto nel caso ce ne siano più d’uno. Sarà compito di quest’ultimo completare la verifica richiesta o inoltrare il ticket a qualche suo sottoposto.

Con questa procedura si assicura che ogni compito svolto verrà poi controllato dal Verificatore designato.



4.2 Risoluzione anomalie

Nel caso il Verificatore noti qualche anomalia nelle fasi di controllo e verifica, dovrà inserire i risultati all'interno del relativo ticket e chiuderlo segnandolo come completato.

Sarà cura del Responsabile poi controllare le modifiche proposte dalla fase di verifica, e nel caso di approvazione creare un nuovo ticket all'autore del prodotto testato per fargli apportare i dovuti cambiamenti.

La procedura appena descritta dovrà venir svolta finchè il Verificatore non notificherà più errori nei suoi test.

4.3 Trattamento delle discrepanze

La procedura per il trattamento delle discrepanze è formalmente equivalente a quella appena elencata per la risoluzione delle anomalie.

L'unica differenza risiede nel fatto che il Verificatore che la nota dovrà inserire, nel resoconto del ticket che conclude la sua fase di verifica, anche il prodotto da cui la discrepanza è nata.

Se delle discrepanze dovessero presentarsi frequentemente tra i documenti prodotti e le NORME DI PROGETTO, il Verificatore sarà tenuto a notificarlo all'Amministratore, il quale, in collaborazione con il Responsabile, procederà ai dovuti provvedimenti.

Non è infatti tollerabile che del tempo venga sprecato per correggere e notificare errori che potrebbero essere facilmente evitati leggendo e seguendo con più attenzione le NORME DI PROGETTO.



4 GESTIONE AMMINISTRATIVA

Diario delle modifiche

DATA	VERSIONE	MODIFICA
<i>21-01-2009</i>	1.1.0	Aggiunta la sezione Automazione delle attività di verifica
<i>09-12-2008</i>	1.0.0	Approvazione del Responsabile e passaggio di stato in “formale”
<i>08-12-2008</i>	0.5.1	Correzioni varie
<i>07-12-2008</i>	0.5.0	Completamento stesura della sottosezione Tecniche e metodi di verifica
<i>06-12-2008</i>	0.4.1	Varie correzioni grammaticali
<i>05-12-2008</i>	0.4.0	Aggiunta delle sezione Gestione Amministrativa
<i>05-12-2008</i>	0.3.0	Stesura della sottosezione Tecniche e metodi di verifica
<i>04-12-2008</i>	0.2.0	Aggiunta delle sezione Visione Generale
<i>03-12-2008</i>	0.1.0	Prima stesura dell'indice del documento