



Piano Di Qualità

Informazioni sul documento

Titolo documento	Piano Di Qualità
Versione attuale	v1.0.0
Data versione attuale	2011/12/16
Data creazione	2011/12/11
Redazione	Luca Lorenzini Giacomo Lorigiola
Revisione	Andrea Zirona
Approvazione	Antonio Pretto
Stato documento	Formale
Uso	Esterno
Distribuito da	SevenFold
Destinato a	SevenFold

Sommario

Il presente documento descrive la strategia di verifica e validazione seguita nello sviluppo del progetto.

Diario delle modifiche

Versione	Data	Autore	Modifiche
v1.0.0	2011/12/16	Antonio Pretto	Approvazione e rilascio prima versione
v0.3.1	2011/12/15	Luca Lorenzini	Migliore definizione dei punti 3.1, 3.2 e 3.3
v0.3.0	2011/12/15	Giacomo Lorigiola	Aggiunto 3.2
v0.2.1	2011/12/14	Luca Lorenzini	Modificato punto 3.1.1
v0.2.0	2011/12/12	Luca Lorenzini	Aggiunto punto 2.3.3
v0.1.0	2011/12/11	Luca Lorenzini	Creazione documento

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Normativi	4
2	Visione generale della strategia di verifica	5
2.1	Organizzazione, pianificazione strategica e temporale, responsabilità	5
2.2	Risorse necessarie, risorse disponibili	5
2.3	Strumenti, tecniche, metodi	5
2.3.1	Strumenti	5
2.3.2	Tecniche	6
2.3.3	Metodi	6
3	Gestione amministrativa della revisione	8
3.1	Comunicazione e risoluzione di anomalie	8
3.2	Trattamento discrepanze	8
3.3	Procedure di controllo di qualità di processo	9
4	Resoconto delle attività di verifica	10
4.1	Tracciamento componenti - requisiti	10
4.2	Dettaglio delle verifiche tramite analisi	10
4.3	Dettaglio delle verifiche tramite prove (test)	10
5	Pianificazione ed esecuzione del collaudo	11
5.1	Specifiche della campagna di validazione (collaudo incluso)	11

Capitolo 1

Introduzione

1.1 Scopo del documento

Con questo documento si intende garantire e gestire la qualità del prodotto software definendo la strategia generale di verifica e di validazione messa in atto per il progetto PMAC^g. La prima versione del documento avrà valore contrattuale per la gara d'appalto, poi verrà costantemente aggiornato per coprire ogni necessità nello sviluppo del progetto.

1.2 Scopo del prodotto

Il sistema software PMAC si pone come obiettivo la realizzazione di una piattaforma innovativa per l'apprendimento comportamentale nell'ambito della sicurezza del lavoro, che utilizzi le tecniche della gamification^g per incentivare il coinvolgimento e la partecipazione degli utenti e per scardinare l'instaurarsi di abitudini errate.

1.3 Glossario

Per evitare ridondanze tutti i termini e gli acronimi presenti nel seguente documento che necessitano di definizione, saranno seguiti da una "g" ad apice (E.g. User^g) alla loro prima occorrenza e saranno riportati in un documento esterno denominato *Glossario.pdf*. Tale documento accompagna e completa il presente e consiste in un listato ordinato di termini e acronimi con le rispettive spiegazioni.

1.4 Riferimenti

1.4.1 Normativi

- Capitolato d'appalto:
<http://www.math.unipd.it/~tullio/IS-1/2011/Progetto/C3.pdf>
- Norme generali di progetto:
vedi documento fornito in allegato *NormeDiProgetto.pdf*

Capitolo 2

Visione generale della strategia di verifica

Di seguito sono specificati i metodi di verifica dei documenti che saranno adottati per tutto l'ambito di progetto.

2.1 Organizzazione, pianificazione strategica e temporale, responsabilità

La realizzazione del progetto da parte dei membri del gruppo di lavoro sarà sviluppata secondo una divisione dei ruoli basata sulle responsabilità.

La figura del verificatore ha il ruolo di verificare ogni documento, codice e decisione analitica presa dalle rispettive figure addette. La responsabilità del verificatore è quella di respingere o accettare ogni nuovo documento o modifica di esso. Per un esame in dettaglio dei ruoli si veda il documento *Organigramma.pdf*.

Da ora in poi, per l'intera durata del documento con il termine file si intende un qualsiasi testo, sia descrittivo che di codice.

2.2 Risorse necessarie, risorse disponibili

Le risorse per la parte di verifica sono divisibili in categorie:

- **Necessarie:** il verificatore deve aver accesso a tutti i file, sia in versione "stabile" che in versione "finito", inoltre deve poter notificare in breve tempo la conferma o la non conferma di un file.
- **Disponibili:** il verificatore avrà a sua disposizione del software per notificare l'esito della verifica e per comunicare i dettagli di tale procedura, per maggiori dettagli vedere punto 2.3.1.

2.3 Stumenti, tecniche, metodi

2.3.1 Strumenti

Il verificatore deve disporre di alcuni servizi per compiere il suo ruolo in maniera efficiente, quali:

- **Apollo⁹:** utilizzato dal verificatore per segnare come "finito", "accettato", "non accettato" e "stabile" i file presi in analisi, come nelle voci 3. e 5. del punto 2.3.2 e per chiedere l'intervento di un ulteriore verificatore come nella voce 6. del punto 2.3.2.

In generale verrà usato per la comunicazione e la gestione di tutte le anomalie e le discrepanze.

- GitHub^g: utilizzato per segnalare un errore minore, ad esempio un semplice errore di programmazione come nella voce 4. (c) del punto 2.3.2.

2.3.2 Tecniche

Di seguito verrà descritto il comportamento del verificatore nella fase di verifica.

1. Il verificatore si prende carico della verifica di un file che è stato segnato come "finito" dal suo creatore, che quindi è pronto per essere verificato.
2. Il verificatore analizza il file preso in carico e ne giudica vari aspetti:
 - (a) Progettuale: concerne la fase progettuale che sta alle spalle della creazione dei file e si giudica la conformità alle finalità del progetto.
 - (b) Contenuto: il contenuto deve essere inerente alle scelte prese dai progettisti/analisti.
 - (c) Codice: il codice deve essere corretto sia semanticamente che sintatticamente.
 - (d) Formale: la forma dei documenti deve essere inerente all'ambito di sviluppo.
3. Il verificatore accetta il file se e solo se tutti le voci del punto precedente hanno esito positivo, un file dopo essere stato accettato è in stato "stabile" e rimarrà in questo stato fino a nuove modifiche che necessitano di una nuova verifica.
4. Il verificatore, in caso trovi delle anomalie in una delle voci del punto 2, deve segnare come "non accettato" il file e deve fornire delle motivazioni in base all'entità dell'errore che, in accordo con il punto 2, può essere:
 - (a) Progettuale: il verificatore deve indicare il motivo per il quale il progetto del file non è inerente alle finalità del progetto, specificando in dettaglio la sezione contenete l'anomalia.
 - (b) Contenuto: il verificatore deve indicare il motivo per il quale il contenuto del file non è conforme alle scelte prese dai progettisti ed analisti, specificando in dettaglio la sezione contenete l'anomalia.
 - (c) Codice: il verificatore deve indicare l'entità dell'errore nel codice, specificando in dettaglio la sua posizione.
 - (d) Formale: il verificatore deve motivare la non accettazione, specificando in dettaglio la posizione della parte la qui forma non è adatta all'ambito di sviluppo.
5. L'autore del file non accettato, o di un altro membro che lo sostituisce, deve attuare il "restart" del file e seguendo le istruzioni del verificatore sistemare gli errori riscontrati.
6. In caso l'autore del file che è stato segnato in "restart" non riesca ad individuare l'errore o non ne condivide l'entità, può richiedere un'ulteriore revisione del file con la collaborazione di un verificatore terzo.

2.3.3 Metodi

Ticketing

La procedura di assegnazione della correzione degli errori è denominata ticketing.

Un ticket è un'assegnazione, come può essere quella eseguita da parte del verificatore a favore di un programmatore o progettista, che deve seguire le norme descritte al capitolo 4.2 del documento *NormeDiProgetto.pdf*

La creazione di un ticket può essere eseguita da ogni membro del gruppo di lavoro e deve rispettare la struttura appena descritta. In caso un ticket sia creato in modo errato si deve procedere contattando il creatore del ticket chiedendo delucidazioni per evitare di perdere l'utilità dello stesso.

In generale la segnalazione di un errore non deve contenere indicazioni sull'entità della correzione, ad eccezione di anomalie la cui spiegazione risulterebbe particolarmente complicata se priva di essa. La procedura di creazione dei ticket prevede che per più errori trovati in una unica sezione di verifica vengano aperti più ticket, per avere un resoconto e uno storico più dettagliato.

Capitolo 3

Gestione amministrativa della revisione

3.1 Comunicazione e risoluzione di anomalie

Un'anomalia si presenta quando in seguito ad una revisione si riscontrano degli errori di programmazione o di stesura dei documenti, che non riguardano la progettazione, ma sono nati nelle fasi successive.

Nel caso si trovi un'anomalia si deve comunicare al redattore del documento che il compito portato a termine presenta un'anomalia, specificando la locazione, l'entità e la motivazione della segnalazione.

Le varie tipologie di anomalie, in accordo con il capitolo 2.3.2, sono:

1. Anomalie di contenuto:

Un'anomalia di contenuto è un'anomalia che riguarda un file, o una parte di esso, che non è inerente alle decisioni progettuali prese dall'analista/progettista che ha progettato il documento.

2. Anomalie di codice:

Un'anomalia di codice prevede un errore logico o sintattico commesso nel codice.

3. Anomalia formale:

Un'anomalia formale è un'anomalia che riguarda la forma dei file, che quindi non è coerente all'ambito di sviluppo del progetto.

Nel caso si trovi una qualunque delle anomalie descritte sopra, questa deve essere segnalata attraverso Apollo con l'apertura di un apposito ticket.

In generale, in seguito alla segnalazione di un'anomalia i programmatori dovranno implementare le modifiche richieste dal verificatore ed infine i verificatori inizieranno la verifica di tutti i documenti modificati e gli eventuali nuovi documenti.

3.2 Trattamento discrepanze

Sarà considerata discrepanza un discostamento del prodotto dalle attese del cliente e dello stesso gruppo SevenFold in base ai requisiti specificati nell'*AnalisiDeiRequisiti.pdf*, quindi il prodotto risultante non rispecchia le aspettative del cliente. Solitamente una discrepanza è attribuibile ad un errore di analisi o di progettazione.

Qualora durante il ciclo di vita di sviluppo del prodotto si verificassero delle discrepanze, queste saranno segnalate attraverso l'apertura di un apposito ticket in Apollo che sarà assegnato al responsabile del progetto.

Il ticket di notifica di una discrepanza sarà indicato con titolo *"ChangeRequest"* e dovranno

essere specificate le origini e le motivazioni del discostamento del prodotto dalle attese riportando una dettagliata descrizione della stessa. Di norma tali discrepanze verranno riscontrate dai verificatori durante la loro normale attività di verifica, ma potranno essere sollevate anche da qualsiasi figura del gruppo nel momento in cui questa si accorga della presenza di discrepanze all'interno del prodotto.

Successivamente il responsabile dovrà analizzare l'entità della discrepanza e assegnare a questa un livello di gravità valutato sulla base dell'importanza del discostamento dalle attese, e sulla base dei costi e dei tempi necessari per risolvere la discrepanza.

Verrà quindi definito un piano correttivo della discrepanza che analizzerà come procedere nella risoluzione della stessa.

Nel caso in cui dal piano correttivo si ricavi che la gravità della discrepanza sia così grave a tal punto da poter compromettere la realizzazione del prodotto o i costi e tempi risultassero inaccettabili per il cliente, la discrepanza sarà risolta soltanto parzialmente o non sarà risolta. In alternativa verrà attuato il rispettivo piano correttivo, al termine del quale verrà convocato un verificatore che dovrà testare la reale risoluzione dell'assonanza, altrimenti la suddetta procedura si ripeterà ricorsivamente.

A discrepanza risolta verrà effettuato un test di regressione per verificare la perfetta compatibilità delle modifiche apportate con il restante prodotto. A test concluso, verrà chiuso il ticket di notifica di discrepanza.

3.3 Procedure di controllo di qualità di processo

Per assicurare la qualità del processo si effettua un controllo su tutti i membri del gruppo di progetto e il loro prodotto, si vede nella figura di Amministratore la persona che deve gestire l'efficienza del lavoro.

Nel caso si riscontri una falla dovuta ad uno qualsiasi dei membri del team, si provvede a segnalare all'Amministratore, che in tempi rapidi la elabora e provvede una segnalazione al diretto interessato. Una falla è un comportamento non efficiente o inadeguato per il lavoro di gruppo e la realizzazione del progetto.

Tutte le segnalazioni sono tracciate tramite software, in generale Apollo, mentre per casi specifici si può prevedere l'intervento umano dell'Amministratore. La procedura che garantisce la qualità del processo è implementata grazie all'adozione del ciclo di vita incrementale, che permette di progettare, implementare e rivedere ogni modifica apportata al progetto e ogni nuovo componente prima di procedere con la fase successiva.

Capitolo 4

Resoconto delle attività di verifica

4.1 Tracciamento componenti - requisiti

Ogni attività di verifica, quindi ogni verifica compiuta da un verificatore e ogni indicazione scritta a riguardo, sono eseguite tramite software (come descritto nel punto 2.3.2) che permette il tracciamento e il relativo storico.

4.2 Dettaglio delle verifiche tramite analisi

Il tracciamento delle modifiche, come descritto al punto precedente, permette all'Amministratore di tenere sotto controllo tutto l'avanzamento del progetto. Tramite analisi delle verifiche si può verificare l'avanzamento del progetto, le tempistiche e la qualità del lavoro. Per facilitare il lavoro dell'Amministratore è previsto che possa prendere contatto diretto con gli altri membri del gruppo di lavoro per chiarimenti e notifiche seguendo una procedura che sarà decisa di volta in volta in base alle necessità.

4.3 Dettaglio delle verifiche tramite prove (test)

Il dettaglio delle verifiche tramite prove sarà specificato quando si avranno sufficienti elementi per stabilire un adatto criterio per eseguire i test.

Capitolo 5

Pianificazione ed esecuzione del collaudo

Il collaudo per essere significativo deve seguire una procedura standard così da poter eseguire un'analisi sui risultati.

5.1 Specifica della campagna di validazione (collaudo incluso)

La validazione tramite collaudo sarà specificata in dettaglio quando si avranno sufficienti elementi per stabilire un adatto criterio di giudizio.