

404NotFound

Premi: better than Prezi.



Piano di Qualifica

Informazioni sul documento

Versione	1.0
Redazione	Camborata Marco Rettore Andrea
Verifica	Manuto Monica Cossu Mattia
Responsabile	Vegro Federico
Uso	Esterno
Ultima modifica	22 Gennaio 2015
Lista di distribuzione	404NotFound

Descrizione

Documento riguardante le strategie di verifica adottati dal gruppo 404NotFound e gli obiettivi qualitativi del il progetto Premi.

Registro delle modifiche

Versione	Autore	Data	Descrizione
1.0	Vegro Federico	22-01-2015	Approvazione documento
0.???	Cossu Mattia	22-01-2015	Verifica finale documento
0.???	Manuto Monica	20-01-2015	Verifica prima stesura documento
0.1	Camborata Marco	05-01-2015	Stesura scheletro documento

Tabella 1: Storico versioni del 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
1.4.2	Informativi	4
2	Visione Generale della Strategia di Verifica	5
2.1	Organizzazione	5
2.1.1	Analisi dei requisiti	5
2.1.2	Progettazione	6
2.1.3	Realizzazione	6
2.1.4	Validazione	6
2.2	Pianificazione Strategica e Temporale	6
2.3	Responsabilità	7
2.4	Strumenti, Tecniche e Metodi	7
2.4.1	Strumenti	7
2.4.2	Tecniche di Analisi	8
2.4.3	Misure e Metriche	10
3	Risorse	11
3.1	Risorse Necessarie	11
3.1.1	Risorse Umane	11
3.1.2	Risorse Software	11
3.1.3	Risorse Hardware	11
3.2	Risorse Disponibili	11
3.2.1	Risorse Software	11
3.2.2	Risorse Hardware	12
4	Obiettivi di Qualità	13
4.1	Qualità dei Processi	13
4.2	Qualità del Prodotto	13
4.2.1	Funzionalità	13
4.2.2	Affidabilità	13
4.2.3	Efficienza	13
4.2.4	Usabilità	13
4.2.5	Manutenibilità	13
4.2.6	Portabilità	13
5	Gestione amministrativa della revisione	14
6	Resoconto dell Attività di Verifica	15
7	Pianificazione ed Esecuzione del Collaudo	16

Elenco delle tabelle

1	Storico versioni del documento.	1
---	---	---

Elenco delle figure

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di illustrare le strategie adottate per implementare i processi di verifica e validazione del lavoro svolto da 404NotFound per assicurare la qualità del progetto Premi e dei processi coinvolti nel suo sviluppo. Per raggiungere gli obiettivi di qualità è necessario un processo di verifica continua sulle attività svolte, per questo motivo il presente documento potrà essere aggiornato in seguito a scelte progettuali del gruppo e/o variazione dei requisiti da parte del Proponente.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di un software di presentazione di slide non basato sul modello di PowerPoint_G, sviluppato in tecnologia HTML5_G e che funzioni sia su desktop che su dispositivo mobile. Il software dovrà permettere la creazione da parte dell'autore e la successiva presentazione del lavoro, fornendo effetti grafici di supporto allo storytelling e alla creazione di mappe mentali.

1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio e ai termini utilizzati nei documenti formali tutti i termini e gli acronimi presenti nel seguente documento che necessitano di definizione saranno seguiti da una "G" in pedice e saranno riportati in un documento esterno denominato Glossario_v1.0.pdf. Tale documento accompagna e completa il presente e consiste in un listato ordinato di termini e acronimi con le rispettive definizioni e spiegazioni.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** NormeDiProgetto_v1.0.pdf
- **Capitolato d'appalto C4:** Premi: Software di presentazione better than Prezi
<http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>

1.4.2 Informativi

- piano di progetto

2 Visione Generale della Strategia di Verifica

2.1 Organizzazione

Per di garantire la qualità del prodotto in tutte le sue fasi di realizzazione, accertandone la conformità rispetto a quanto emerso durante la fase di Analisi dei Requisiti (vedi allegato *AnalisiDeiRequisiti_v1:0:pdf*), si intende svolgere una costante attività di verifica trasversale a tutte le fasi di sviluppo del progetto.

Per poter effettuare un corretto processo di verifica si è scelto di effettuare le dovute operazioni di controllo ogni volta che il prodotto in esame avrà maturato sostanziali modifiche rispetto alla sua precedente versione. Per quanto riguarda la documentazione questa maturazione si rispecchia nel variare dell'indice di versione dei documenti stessi (vedi documento interno *NormeDiProgetto_v1.0.pdf*, sezione 6.6) e una fase di verifica finale è necessaria affinché un qualsiasi documento possa passare alla fase di approvazione da parte del *Responsabile del Progetto*. E' auspicabile che siano svolte verifiche sui documenti non solo prima dell'approvazione ma anche in fasi intermedie nelle quali il documento può non essere ancora stato completato. Ogni svolgimento di una fase di verifica globale sarà riportata nell'apposito registro delle modifiche. Per assicurare il massimo livello di controllo, tuttavia, un primo controllo sommario sui nuovi contenuti viene svolto dal *Verificatore* ad ogni modifica del documento (per approfondimento vedi documento interno *NormeDiProgetto_v.1.0.pdf* sezione 5.4)

Si è scelto e adottato il metodo "Broken Window Theory" secondo il quale, non appena un errore viene rilevato, questo andrà segnalato e corretto il prima possibile onde evitarne la propagazione.

Il ciclo di vita scelto per lo sviluppo del progetto è un ciclo di vita incrementale (vedi documento allegato *PianoDiProgetto_v1.0*) e di conseguenza le operazioni di verifica verranno realizzate in modo tale da intervenire in maniera coerente nelle varie fasi del progetto come illustrato di seguito:

2.1.1 Analisi dei requisiti

Tutta la documentazione relativa alla RR, una volta completata, entrerà nella dedicata fase di revisione. Di seguito i parametri di controllo:

- La presenza di eventuali errori lessico/grammaticali e la generale correttezza dei contenuti esposti. Nel dettaglio, il controllo ortografico verrà effettuato con gli strumenti messi a disposizione da *TexMaker_G*, mentre il controllo lessicale, grammaticale e sintattico da un'accurata rilettura del testo.
- Il controllo dei contenuti con l'obiettivo di verificare la copertura delle richieste del proponente e questo tramite un'accurata rilettura e confronto con il capitolato d'appalto.
- Corrispondenza tra ogni requisito e caso d'uso corrispondente.
- Verifica dei contenuti grafici e tabellari e conformità dei documenti alle *Norme di Progetto* stabilite.

Se durante la verifica saranno state rilevate irregolarità queste verranno segnalate tramite un apposito ticket dal verificatore e corrette dal redattore.

2.1.2 Progettazione

Il processo di verifica in fase di Progettazione consisterà nel verificare che tutti i requisiti descritti durante la fase di Analisi dei Requisiti siano tracciabili nei componenti individuati e viceversa che ogni componente soddisfi o sia associato ad almeno un requisito. Qualora dalla verifica sorgano incongruenze o mancanze, queste verranno segnalate tramite ticket e successivamente risolte.

2.1.3 Realizzazione

La verifica in questa fase verrà effettuata da parte dei programmatori stessi utilizzando appositi e specifici strumenti di verifica automatizzata del codice. La presenza di errori verrà segnalata da un apposito ticket che verrà preso in carico dai programmatori e chiuso una volta risolto il problema.

2.1.4 Validazione

Il team 404NotFound si impegna a garantire il corretto funzionamento del prodotto Premi e a fornire al collaudo una versione funzionante e possibilmente completa del prodotto. Nel caso in cui vengano riscontrati malfunzionamenti o discrepanze tra le caratteristiche del prodotto e le richieste del cliente sarà cura del fornitore eliminare tali difetti, interamente a proprio carico.

2.2 Pianificazione Strategica e Temporale

Avendo l'obiettivo di rispettare le scadenze fissate nel Piano di Progetto v1.0, è necessario che l'attività di verifica della documentazione e del codice sia sistematica e ben organizzata. Ogni fase di redazione dei documenti e di codifica deve essere preceduta da una fase di studio preliminare per eliminare all'origine possibili imprecisioni di natura concettuale e/o tecnica.

Il processo di verifica viene strutturato in tre fasi:

1. **Pre-Verifica:** Si tratta della pianificazione e la preparazione delle attività di verifica. Consiste nella scelta delle persone che si occuperanno di questa attività e nella distribuzione dei documenti o componenti software da controllare.
2. **Verifica effettiva:** I *Verificatori* lavorano indipendentemente per trovare errori, omissioni e scostamenti rispetto agli standard, durante questa fase, un autore del documento o componente software attende il responso del *Verificatore*. Deve stillato un elenco delle azioni correttive da intraprendere.
3. **Post-Verifica:** Dopo che le correzioni sono state apportate al componente in esame il *Verificatore* usando come checklist l'elenco delle correzioni da lui redatto nella fase precedente, potrà constatare l'avvenuta correzione.

Durante le attività di verifica è inevitabile che gli errori commessi dagli individui vengano esposti a tutto il gruppo. E' quindi molto importante che si incoraggi nel team una mentalità per la quale la segnalazione degli errori non diventi motivo per screditare il lavoro di un singolo, ma occasione di crescita per la persona e per l'intero gruppo di lavoro.

2.3 Responsabilità

La responsabilità dell'attività di verifica viene affidata ai seguenti ruoli:

- **Responsabile di Progetto:** Macroscopicamente ha il compito di controllare che l'evoluzione del progetto rispetti le tempistiche prefissate, è garante della qualità dei processi interni e della conformità dei prodotti a quanto pianificato e progettato ponendosi come garante nei confronti del *Committente*. In particolare in questo contesto ha il compito di assicurarsi che le attività di verifica vengano svolte sistematicamente e non vi siano conflitti di interesse tra redattori e verificatori. Egli è l'unico a poter decidere l'approvazione di un documento e a sancirne la distribuzione.
- **Verificatore:** Ha il compito di coordinare e definire le attività volte alla verifica del materiale prodotto, sia esso software, documenti o materiale d'altro genere. La responsabilità del verificatore è quella di respingere o validare ogni nuovo documento o modifica di esso e di segnalare formalmente gli errori riscontrati.

2.4 Strumenti, Tecniche e Metodi

2.4.1 Strumenti

Per lo svolgimento del processo di verifica faremo uso dei seguenti strumenti:

- **Correttore automatico di TeXMaker:** come segnalato nelle Norme di Progetto v1.0 per la scrittura di documenti si è scelto di utilizzare l'ambiente grafico TeXMaker. Tale strumento integra i dizionari di OpenOffice.org e segnala i potenziali errori ortografici presenti nel testo;
- **404TrackerDB:** Strumento software realizzato dal gruppo 404NotFound che contiene ed associa:
 - Requisiti individuati durante l'analisi;
 - Fonti di requisiti individuate, inclusi anche i casi d'uso.

Permette inoltre di esportare automaticamente:

- Codice \LaTeX per la descrizione dei casi d'uso;
- Tabella in \LaTeX per il tracciamento fonti-requisiti.
- Strumenti W3C_G (www.w3.org) per la validazione:
 - validatore HTML5_G (<http://validator.w3.org>)
 - validatore CSS_G (<http://jigsaw.w3.org/css-validator/>)

- Strumenti per debugging_G HTML_G, CSS_G e JavaScript_G messi a disposizione dai vari browser_G:
 - **Chrome Developer Tools** (<https://developers.google.com/chrome-developer-tools>)
 - **Firebug** (<http://getfirebug.com/>)
- **JSLint** Ambiente di test (<http://www.jshint.org>): tool per la validazione di codice JavaScript_G;
- **JUnit** (<http://www.junit.org>): semplice framework per eseguire test ripetibili;
- **BrowserStack** (<http://www.browserstack.com/>): per eseguire il test comparato sui vari browser_G;
- **WebStorm** (<https://www.jetbrains.com/webstorm/>): IDE JavaScript scelto come ambiente di sviluppo.

2.4.2 Tecniche di Analisi

Anamalisi Statica:

Consiste nell'analisi della documentazione e dei prodotti software senza effettuare l'esecuzione. Viene svolta mediante due tecniche complementari:

- **Walkthrough:** È una tecnica che viene utilizzata soprattutto nelle prime fasi del progetto, quando ancora non è stata maturata una adeguata esperienza da parte dei membri del gruppo, che permetta di attuare una verifica più mirata e precisa. Consiste nella rilettura completa e metodica da parte dell'autore stesso o da parte del *Verificatore* allo scopo di trovare errori. Con l'utilizzo di questa tecnica, il Verificatore sarà in grado di stilare una lista di controllo con gli errori più frequenti in modo da favorire il miglioramento di tale attività nelle fasi future. Questa è un'attività onerosa e collaborativa che richiede l'intervento di più persone per essere efficace ed efficiente. Segue una fase di discussione con la finalità di esaminare i difetti riscontrati e di proporre le dovute correzioni. L'ultima fase consiste nel correggere gli errori rilevati.
- **Inspection:** Questa tecnica consiste nell'analisi di alcune parti del documento o del codice alla ricerca di errori solo in parti ritenute critiche in base all'esperienza derivata dalle revisioni precedenti. La lista di controllo o checklist, che deve essere seguita per svolgere efficacemente questo processo, deve essere redatta anticipatamente ed è frutto del lavoro svolto dai verificatori con la tecnica di walkthrough. L'Inspection è da preferire al Walkthrough, poichè non necessita della lettura integrale dei documenti in oggetto, ma richiede un sufficiente livello di dettaglio nella lista di controllo.

La lista di controllo risultante è la seguente: thgdfddbfavabrwnbafvgrhjrnsvbfewgrt-jryrnbvfjetynbgdbfvrjyngbdfv

Metodi di Analisi Statica:

- **Analisi del flusso di controllo:** si controlla che il codice sia correttamente strutturato e che segua il flusso aspettato, che non vi siano parti del programma che possano non terminare e che non esistano porzioni di codice non raggiungibile.
- **Analisi del flusso dei dati:** si accerta che il software non acceda mai a variabili non inizializzate o non modifichi più volte di seguito una variabile senza leggerne il valore tra una modifica e l'altra.
- **Analisi del flusso di informazione:** si verifica che le uniche dipendenze tra gli input e gli output di ogni unità di codice o di più unità siano quelle previste in fase di progettazione.

Analisi Dinamica:

Consiste nella verifica dei componenti del software o del sistema in generale e richiede l'esecuzione del programma per eseguire il test. Perchè tale attività sia utile e generi risultati attendibili è necessario che i test effettuati siano ripetibili: cioè dati uno stesso input e uno stesso ambiente di esecuzione deve fornire gli stessi risultati quando vengono effettuate più prove. Questi risultati saranno utili solo se porteranno alla luce errori permettendo di correggerli, nel caso non vengano riscontrate anomalie, ciò non costituisce una prova dell'assenza di errori.

Metodi di Analisi Dinamica:

- **Test di unità:** Viene verificata ogni unità software che deve soddisfare i requisiti per essa richiesti ed è necessario testare tutte le possibili esecuzioni del codice che lo compone. Per unità si intende la più piccola quantità di software che è utile verificare singolarmente e che viene prodotta da un singolo programmatore.
- **Test di integrazione:** I moduli che hanno superato il test di unità possono venire integrati tra di loro. Il test di integrazione ha lo scopo di individuare errori residui nella realizzazione dei moduli e problemi nell'integrazione con altre componenti fornite da terze parti che non si conoscono a fondo.
- **Test di sistema e collaudo:** Serve a verificare il completo soddisfacimento dei requisiti software stabiliti in fase di Analisi. Consiste nella validazione del prodotto software nel momento in cui vengono aggiunti tutti i componenti. Il test potrà riguardare, in una fase iniziale, solamente alcune delle componenti del prodotto finale, per poi interessare il sistema nella sua interezza.
- **Test di regressione:** Consiste nell'eseguire nuovamente i test di unità e di integrazione su componenti software alle quali sono stati apportati cambiamenti. Serve a controllare che le modifiche apportate non provochino malfunzionamenti alla componente stessa o ad altre che dipendono da essa.
- **Test di accettazione:** È il test di collaudo del prodotto software che viene eseguito in presenza del Committente. Se questa fase finale di test viene superata positivamente si può procedere al rilascio ufficiale del prodotto sviluppato.

2.4.3 Misure e Metriche

Le misure e le metriche che il team adotterà si ispireranno alle indicazioni dello standard ISO/IEC-14598. Tale norma descrive il processo di valutazione della qualità del software. Vengono di seguito descritte le metriche sulle quali 404NotFound intende basarsi nei processi di verifica, sia in fase di progettazione che di codifica, che potranno essere integrate e stabilite con maggiore precisione durante l'avanzamento del progetto.

- **Complessità ciclomatica:** La complessità ciclomatica di un programma è il numero di cammini linearmente indipendenti attraverso il codice sorgente. Per esempio, se il codice sorgente non contiene IF o FOR, allora il livello di complessità sarà 1, poiché esiste un solo cammino, se è presente un IF la complessità diventa di 2. Un programma si può quindi rappresentare come un albero nel quale i nodi sono i blocchi del programma e gli archi sono il passaggio del controllo da un blocco all'altro. La complessità è quindi definita come:

$$C = c - n + 2p$$

- **Misure nella progettazione:**
 - profondità di annidamento dei costrutti condizionali
 - numero di classi e coesione tra di esse
 - complessità di flusso fan in e fan out
- **Misure sul codice:**
 - lunghezza del codice
 - lunghezza degli identificatori
 - numero di parametri
 - numero di campi dati per classe

3 Risorse

La gestione della qualità prevede l'utilizzo di alcune risorse suddivisibili in categorie.

3.1 Risorse Necessarie

3.1.1 Risorse Umane

I ruoli necessari per garantire un'adeguata qualità sono i seguenti:

- **Responsabile del Progetto:** E' responsabile nei confronti del committente della corretta realizzazione del prodotto;
- **Verificatore:** Coordina e svolge le attività di verifica vere e proprie;
- **Programmatore:** Esegue attività di debugging sul codice.

Per una descrizione dettagliata delle figure elencate e di tutti gli altri ruoli specifici si rimanda al documento allegato *PianoDiProgetto_v1.0.pdf*.

3.1.2 Risorse Software

Durante la fase di realizzazione del progetto saranno necessari:

- Software per la gestione di documenti in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
- Piattaforma di testing sui vari browser_G dell'applicazione da sviluppare;
- Piattaforma di versionamento per la creazione e la gestione di ticket_G.
- Software per la creazione dei diagrammi in UML_G;
- Ambiente per lo sviluppo del codice nel linguaggio di programmazione scelto;
- Strumenti di validazione del codice prodotto.

3.1.3 Risorse Hardware

- Computer dotati di tutti gli strumenti software descritti nel Piano di Qualifica e nelle Norme di Progetto.
- Luogo fisico in cui incontrarsi per lo sviluppo del progetto, possibilmente con una connessione ad Internet.

3.2 Risorse Disponibili

3.2.1 Risorse Software

Vengono di seguito elencate le risorse software disponibili. Per una descrizione più dettagliata si rimanda alla sottosezione Strumenti 2.4.1 del presente documento.

- TeXMaker per l'editing dei documenti in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
- BrowserStack per il testing sui vari browser_G;

- GitHub per il versionamento e la gestione dei ticket_G;
- Astah per i diagrammi UML_G;
- WebStorm come ambiente di sviluppo.
- Strumenti di validazione online del W3C_G.

3.2.2 Risorse Hardware

- Computer personali (portatili o fissi) dei membri del gruppo;
- Computer messi a disposizione nei laboratori informatici del Dipartimento di Matematica Pura ed Applicata dell'Università di Padova;
- Aule studio del Dipartimento di Matematica Pura ed Applicata dell'Università di Padova.

4 Obiettivi di Qualità

4.1 Qualità dei Processi

4.2 Qualità del Prodotto

4.2.1 Funzionalità

4.2.2 Affidabilità

4.2.3 Efficienza

4.2.4 Usabilità

4.2.5 Manutenibilità

4.2.6 Portabilità

5 Gestione amministrativa della revisione

6 Resoconto dell Attività di Verifica

7 Pianificazione ed Esecuzione del Collaudo