



MashUp

Software Engineering Group

info@mashup-unipd.it

Informazioni Documento

Nome documento	<i>Piano di Qualifica</i>
Versione	<i>v0.0.1</i>
Data redazione	2014-12-14
Redattori	Ceccon Lorenzo Faccin Nicola
Verificatori	Santacatterina Luca
Approvazione	Cognome Nome
Lista distribuzione	<i>MashUp</i> <i>Prof. Tullio Vardanega</i> <i>Prof. Riccardo Cardin</i> <i>Dott. David Santucci - Zing Srl</i>
Uso	Esterno

Sommario

Documento che descrive le attività di verifica e validazione adottate dal gruppo MashUp per il progetto BDSMApp.

Diario Revisioni

Modifica	Autore & Ruolo	Data	Versione
<i>Inizio stesura del documento</i>	Ceccon Lorenzo <i>Redattore</i>	2014-12-14	v0.0.1
<i>Terminata stesura sezione introduzione</i>	Ceccon Lorenzo <i>Redattore</i>	2014-12-14	v0.0.2

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Visione generale della strategia di verifica	2
2.1	Definizione degli obiettivi	2
2.2	Organizzazione	2
2.3	Pianificazione strategica e temporale	2
2.4	Responsabilità	2
2.5	Risorse necessarie	2
2.6	Misure e metriche	3
2.6.1	Metriche per i processi	3
2.6.2	Metriche per i documenti	3
2.6.3	Metriche per il software	3
3	Gestione amministrativa della revisione	4
3.1	Gestione delle anomalie e delle discrepanze	4
3.2	Procedure di controllo di qualità di processo	4
3.3	Procedure di controllo di qualità di processo	5
4	Resoconto delle attività di verifica	6
5	Pianificazione ed esecuzione del collaudo	6

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di definire la strategia e descrivere le modalità di verifica e validazione che il gruppo MashUp intende adottare per lo sviluppo del progetto al fine di raggiungere gli obiettivi qualitativi prefissati. Per perseguire questi obiettivi è necessaria una costante attività di verifica in modo da permettere rilevare e risolvere eventuali anomalie.

1.2 Scopo del prodotto

Lo scopo del progetto denominato BDSMAApp è di sviluppare un'applicazione cloud che permetta il monitoraggio dei big data nei social network al fine di offrire un sistema che consenta all'utente finale di accedere ai contenuti dei social network in maniera più fruibile possibile. L'obiettivo principale è, quindi, quello di creare una applicazione composta da un'interfaccia web che permetta di consultare e interrogare i dati e da dei servizi REST interrogabili.

1.3 Glossario

Al fine di permettere una migliore comprensione del documento, i termini tecnici e gli acronimi utilizzati sono riportati nel documento glossario1.0.0 che contiene una descrizione dettagliata di tutti i termini utilizzati. In questo documento, i termini tecnici presente nel glossario saranno riportati in corsivo e avranno una 'g' corsiva come pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di progetto:** *NormediProgetto1.0.0*
- **Capitolato d'appalto C1:** *BDSMAApp: Big Data Social Monitoring App*
<http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C1.pdf>
- **Standard ISO/IEC 9126:** http://en.wikipedia.org/wiki/ISO/IEC_9126
- **Standard ISO/IEC 15504:** http://en.wikipedia.org/wiki/ISO/IEC_15504

1.4.2 Informativi

- **Piano di progetto:** *PianodiProgetto1.0.0*
- **Slides di Ingegneria del Software modulo A:** <http://www.math.unipd.it/~tullio/IS-1/2014/>
- **SWEBOK 2004:** *Chapter 11 - Software Quality* <http://www.computer.org/portal/web/swebok/html/ch11>
- **Software Engineering 9th Edition Ian Sommerville:** *Chapters 8, 24, 26*

2 Visione generale della strategia di verifica

2.1 Definizione degli obiettivi

Per garantire la qualità di processo necessaria per ottenere la qualità di prodotto che si intende raggiungere, è stato scelto di basarsi su due modelli:

- **SPICE:** *definito nello standard ISO/IEC 15504 è il modello di riferimento per dare un giudizio di maturità e individuare azioni migliorative.*
- **PDCA:** *serve per promuovere il miglioramento continuo dei processi e per ottimizzare l'utilizzo di risorse.*

Per massimizzare l'efficacia della qualità del prodotto si è deciso di puntare sullo standard ISO/IEC 9126 che definisce le metriche per la sua misurazione.

2.2 Organizzazione

2.3 Pianificazione strategica e temporale

L'attività di verifica necessaria, per il miglioramento della qualità dei processi e del prodotto, deve essere sistematica ed organizzata. Ciò permetterà l'individuazione e la correzione degli errori il prima possibile evitando la propagazione di questi ultimi in larga scala.

Ciascuna attività che riguarda la documentazione o la codifica dovrà essere preceduta da uno studio preliminare che ci permetta di rendere chiaro la struttura degli stessi. Questo studio preventivo ci consentirà di ottenere un maggiore livello di qualità e una minore possibilità di fallimento.

Per quanto riguarda le tempistiche, l'obiettivo primario è quello di rispettare le scadenze forniteci del committente.

2.4 Responsabilità

Le responsabilità relative all'assegnazione degli incarichi appartengono al Responsabile di Progetto, mentre le responsabilità relative all'adeguamento dell'ambiente di lavoro per lo svolgimento di tutti i compiti necessari alla realizzazione del progetto appartengono all'Amministratore di Progetto.

2.5 Risorse necessarie

Le risorse necessarie alla verifica della qualità dei processi e del prodotto sono:

- **Risorse umane:** di Progetto controlla la qualità dei processi interni, l'Amministratore definisce le norme e i piani per le attività di verifica, il Programmatore esegue le prove di verifica e validazione del codice, il Verificatore esegue la verifica dei documenti e fornisce i risultati delle prove effettuate.
- **Risorse software:** Sono necessari strumenti per il tracciamento dei requisiti, per la stesura dei documenti in \LaTeX , per la creazione di diagrammi UML, per lo sviluppo del prodotto e per supporto e verifica del codice.
- **Risorse hardware:** Sono necessari computer per scrivere documentazioni e codice del prodotto e stabili dove poter lavorare al progetto.

2.6 Misure e metriche

Descrizione delle metriche e delle misure per rendere quantificabili e conseguentemente qualificabili i processi, i documenti e il software prodotto.

2.6.1 Metriche per i processi

L'organizzazione interna dei processi si basa sul principio PDCA, che è in grado di garantire un miglioramento continuo della qualità di tutti i processi e conseguentemente dei prodotti derivanti dai processi. I processi saranno pianificati dettagliatamente rispetto ai requisiti e alle risorse disponibili. Se durante il processo di verifica l'analisi evidenzia dei valori che si discostano, in modo peggiorativo, dai piani prefissati, questo denoterà la presenza di un problema che verrà risolto in modo correttivo sul processo o eventualmente sul piano iniziale dello stesso.

Le misurazioni sul processo consistono in:

- Tempo impiegato per essere completato
- Cicli iterativi interni al processo
- Risorse utilizzate e/o consumate durante il processo
- Attinenza ai piani stabiliti
- Soddisfazione dei requisiti richiesti

2.6.2 Metriche per i documenti

Indice Gulpease :questo indice, tarato specificatamente per la lingua italiana, ha anche il vantaggio di utilizzare la lunghezza delle parole in lettere e non delle sillabe, semplificandone il calcolo.

$$89 + \frac{300 * (\text{Numero delle frasi}) - 10 * (\text{Numero delle lettere})}{\text{Numero delle parole}}$$

100 indica la leggibilità più alta mentre 0 quella più bassa, sono presenti dei range così da poter quantificare meglio la complessità del documento in analisi:

- inferiori a 80 sono difficili da leggere per chi ha la licenza elementare;
- inferiori a 60 sono difficili da leggere per chi ha la licenza media;
- inferiori a 40 difficili da leggere per chi possiede un diploma superiore;

Range-ottimale[50-100], range-accettazione [40-100].

2.6.3 Metriche per il software

- **Complessità Ciclomantica** : è utilizzata per misurare la complessità di un metodo, attraverso il grafo di controllo di flusso che misura direttamente il numero di cammini linearmente indipendenti. I nodi di questo grafo rappresentano gruppi indivisibili di istruzioni e gli archi connettono due nodi solamente se le istruzioni di un nodo possono essere eseguite immediatamente dopo le istruzioni dell'altro nodo.

In questo progetto si cercherà di rispettare la raccomandazione di *McCabe*, che sviluppò tale teoria, ossia quella di non superare una complessità di 10.

Rispettando questo vincolo si aumentano le possibilità di riuso del codice, manutenibilità, coesione e correttezza di quest'ultimo. Il vincolo presentato sarà di tipo lasco, ossia potrà essere portato a valori maggiori nell'eventualità porti a notevoli benefici in termini di velocità di esecuzione.

Valore-ottimale <10 , valore-accettazione <15 .

- **Numero di metodi** : metrica utilizzata per calcolare una media delle occorrenze dei metodi per package; valori alti potrebbero indicare la necessità di scomporlo.

Range-ottimale $[3-8]$, range-accettazione $[3-10]$.

- **Numero di parametri** : metrica utilizzata per calcolare il numero di parametri formali di un metodo. Un valore basso e indice di maggior manutenibilità e astrazione del codice.

Range-ottimale $[0-4]$, range-accettazione $[0-8]$.

- **Linee di codice per linee di commento** : metrica atta a migliorare la manutenibilità del codice attraverso il monitoraggio del rapporto tra questi valori.

Valore-ottimale <0.20 , valore-accettazione <0.35

- **Bugs for lines of code** : metrica per la misura dei bug trovati per un certo quantitativo di linee di codice. Questa metrica è utile in quanto all'aumentare dell'ampiezza del codice si aumenta la probabilità di nascondere degli errori. Presupponendo che nessuno del gruppo avrà conoscenze sufficienti dello stack tecnologico che si andrà ad utilizzare si partirà con un valore di accettazione alto per poi cercare di ridurlo in modo incrementale. L'obiettivo fissato è quello di raggiungere valori compresi tra 0 e 20. Difficoltà particolari verranno gestite dal responsabile di progetto.
- **Numero di livelli di annidamento** : metrica per misurare il livello di annidamento dei metodi. Un numero elevato comporta eccessiva complessità del codice e ne riduce il livello di astrazione.

Range-ottimale $[1-4]$, range-accettazione $[1-6]$.

3 Gestione amministrativa della revisione

3.1 Gestione delle anomalie e delle discrepanze

La fase di verifica porta alla ricerca di eventuali difetti, i quali possono essere errori logici oppure anomalie presenti nel codice. Il verificatore ha il compito di esaminare scrupolosamente il codice del prodotto e sottolineare le eventuali anomalie e problemi per essere risolti successivamente.

3.2 Procedure di controllo di qualità di processo

Il PDCA, noto anche come ciclo di Deming, è un metodo di gestione iterativo a quattro fasi per il controllo e il miglioramento continuo dei processi. Le quattro fasi che lo compongono sono:

- **Plan:** Stabilisce gli obiettivi e i processi necessari per ottenere risultati uguali a quelli attesi.
- **Do:** Fase composta dall'attuazione del piano, dall'esecuzione del processo e dalla creazione del prodotto. Termina con una raccolta dei dati sul risultato di quanto ottenuto.
- **Check:** Si confrontano i risultati ottenuti dalla fase precedente con i risultati attesi per verificare la presenza di differenze.
- **Act:** Si effettuano correzioni laddove sono presenti differenze tra i risultati ottenuti e previsti. Si determinano le cause e dove applicare le modifiche per ottenere un miglioramento del processo e del prodotto.

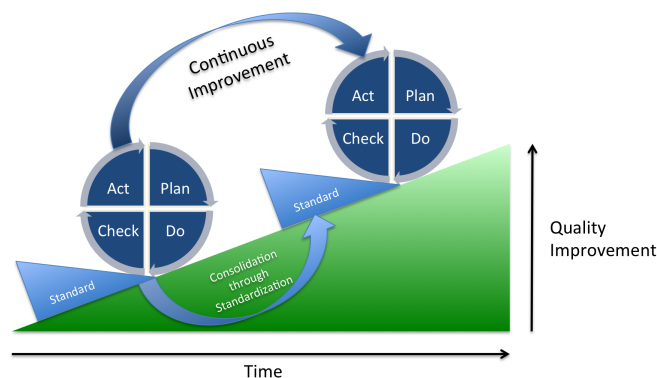


Figura 1: Ciclo di miglioramento della qualità PDCA

3.3 Procedure di controllo di qualità di processo

Per garantire la qualità del prodotto si è deciso di fare riferimento allo standard ISO/IEC 9126 che suddivide la norma tecnica in quattro parti:

- **Quality model:** La prima parte dello standard classifica il modello di qualità nelle seguenti sei caratteristiche:
 - Functionality
 - Reliability
 - Usability
 - Efficiency
 - Maintainability
 - Portability
- **External metrics:** Le metriche esterne applicabili al software, e quindi rilevabili tramite l'analisi dinamica, misurano il comportamento del prodotto sulla base dei test.
- **Internal metrics:** Le metriche interne, misurabili attraverso l'analisi statica, sono utili per prevedere il livello della qualità esterna ed in uso, poiché i suoi attributi interni influiscono su quelli esterni ed in uso. Permettono così di individuare anomalie prima che quest'ultime possano influenzare la qualità del prodotto finale.

- **Quality in use metrics:** La qualità in uso, raggiungibile solo dopo aver ottenuto la qualità interna ed esterna, fornisce metriche per misurare il grado di utilizzabilità del prodotto da parte dell'utente finale.

4 Resoconto delle attività di verifica

5 Pianificazione ed esecuzione del collaudo

Questa sezione sarà redatta quando il prodotto software sarà pronto per il collaudo.