SweeneyThreads

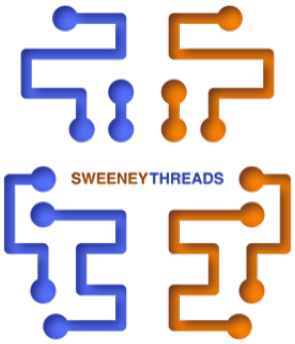
Actorbase

a NoSQL DB based on the Actor model

Piano di qualifica

Redattori:
Bonato Paolo
Bortolazzo Matteo
Maino Elia
Nicoletti Luca
Padovan Tommaso

Approvazione:
Padovan Tommaso
Verifica:
Biggeri Mattia
Tommasin Davide



Versione 1.0.1

19 gennaio 2016

Indice

Diario delle modifiche

Versione	Data	Autore	Descrizione
1.0.1	2016-01-17	*Amministratori* Nicoletti Luca + Tommaso Padovan	Visione generale della strategia di verifica e Gestione amministrativa della revisione
1.0.0	2016-01-17	*Amministratore* Nicoletti Luca	Scrittura scheletro logico del documento

Tabella 1: Diario delle modifiche

Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è di descrivere le scelte effettuate in merito alle strategie che il gruppo ha deciso di adottare per raggiungere obiettivi qualitativi e misurabili da applicare al proprio prodotto. Per soddisfare questi obiettivi sarà necessario attuare un processo di verifica continuo sulle attività svolte in modo da poter rilevare ed eventualmente correggere anomalie e incongruenze in modo tempestivo per evitare danni e sprechi di risorse.

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un DataBase NoSQL key-value basato sul modello ad Attori$_G$ con l'obiettivo di fornire una tecnologia adatta allo sviluppo di moderne applicazioni che richiedono brevissimi tempi di risposta e che elaborano enormi quantità di dati. Lo sviluppo porterà al rilascio del software sotto licenza MIT.

1.3 Glossario

Con lo scopo di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.0*. La prima occorrenza di ogni termine termine contenuto nel *Glossario* e presente in questo documento verrà marcato con una "G" maiuscola in pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di progetto:**
  Norme di progetto v1.1.1;
- **Capitolato d'appalto:**
  `http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1p.pdf`.

1.4.2 Formativi

- **Piano di progetto:**
  Piano di progetto v1.0.4;

- **Slide del corso:**
  `http://www.math.unipd.it/~tullio/IS-1/2015/`;
- **SWEBOK - Version 3:**
  `http://www.computer.org/web/swebok/v3`
- ...

Visione generale della strategia di verifica

2.1 Definizione obiettivi

In questa sezione verranno descritti gli obiettivi di qualità relativi al prodotto che il gruppo ha deciso di raggiungere e gli obiettivi relativi ai processi che saranno svolti per il completamento del progetto.

2.1.1 Qualità di processo

Per garantire la qualità del prodotto è necessario garantire anche quella dei processi necessari al suo completamento. A questo scopo si è deciso di adottare lo standard *ISO/IEC 15504* denominato *SPICE*.

Questo modello descrive come ogni processo debba essere controllato costantemente in maniera da rilevare possibili errori o debolezze e correggerli prima che essi si diffondano, facendo aumentare esponenzialmente il carico di lavoro. Affinché le singole valutazioni contribuiscano all'effettivo miglioramento dei processi devono essere sempre ripetibili, oggettivi e comparabili. *SPICE* definisce livelli di maturità del processo:

0 - Incompete

1 - Performed

2 - Managed

3 - Established

4 - Predictable

5 - Optimizing

Al fine di applicare correttamente questo modello è evidentemente indispensabile adottare il principio *PDCA* il quale definisce una metodologia di controllo dei processi durante il loro ciclo di vita che consente di migliorarne in modo continuativo la qualità.
Esso di compone di 4 fasi:

- **Plan:** definire dettagliatamente cosa deve essere realizzato rispetto agli obiettivi di miglioramento, e come questi controlli saranno effettuati;

- **Do:** fase di esecuzione delle attività pianificate;
- **Check:** vengono confrontati i dati in uscita dalla fase *Do* con quelli pianificati nella fase *Plan*, per intervenire in tempo e migliorare i risultati;
- **Act:** fase in cui si mette in pratica il miglioramento continuo dei processi utilizzando i risultati della verifica per modificare gli aspetti critici dei processi in esame.

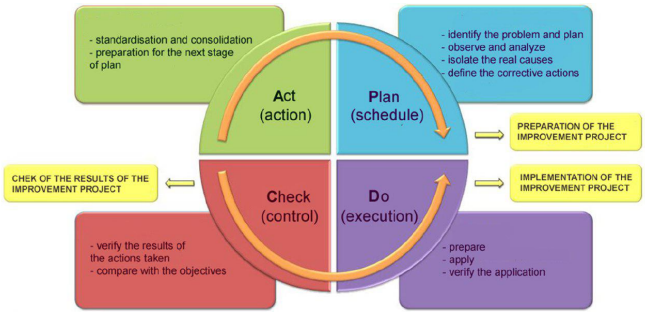


Figura 2.1: Fasi del principio PDCA.

2.1.2 Qualità di prodotto

2.2 Organizzazione

2.3 Pianificazione strategica e temporale

Avendo lo scopo di rispettare le scadenze riportate nel Piano di progetto, è necessario che l'attività di verifica, sia del codice che della documentazione, sia sistematica e ben organizzata; in questo modo l'individuazione, e quindi la correzione degli errori avverrà il prima possibile, limitando la diffusione degli stessi.
Per cercare di ridurre il numero degli errori, e quindi semplificare l'attività di verifica, ogni fase di codifica o documentazione sarà preceduta da una fase di studio preliminare. Evitando le imprecisioni di natura concettuale si ridurranno le correzioni necessarie.

Di seguito vengono riportate le scadenze previste:

- **Revisione dei requisiti:** 2016-01-22;
- **Revisione di progettazione:** 2016-04-18;
- **Revisione di qualifica:** 2016-05-23;
- **Revisione di accettazione:** 2016-06-17.

2.4 Responsabilità

Il *Responsabile di progetto* ha il compito di:

- Accertarsi che le attività di verifica vengano svolte sistematicamente secondo quanto riportato nelle *Norme di progetto*;
- Accertarsi che vengano rispettati ruoli e competenze assegnate nel *Piano di progetto*;
- Verificare che non ci siano conflitti di interesse tra redattori e *Verificatori*;
- Aprire ed assegnare i ticket principali e le task-list;
- Approvare un documento e sancirne la distribuzione.

I *Verificatori* hanno il compito di:

- Effettuare la verifica dei documenti con strumenti e metodi proposti nel *Piano di Qualifica*;7
- Attenersi rigidamente a quanto sancito nelle *Norme di progetto*;
- Segnalare tempestivamente un errore, qualora riscontrato;
- Sottoporre i documenti all'approvazione del *Responsabile*, una volta giunti ad uno stadio finale.

2.5 Risorse

Per la realizzazione del progetto sono necessarie risorse sia umane che tecnologiche.

2.5.1 Risorse umane

Vengono descritte nel dettaglio nel *Piano di progetto* e sono:

- *Responsabile di progetto*;
- *Amministratore*;
- *Analista*;
- *Progettista*;
- *Programmatore*;
- *Verificatore*.

2.5.2 Risorse software

Sono necessari tutti i software utili

- alla gestione di documentazione in LaTeX;
- alla creazione di diagrammi UML;
- allo sviluppo di codice *Scala*;
- a semplificare ed automatizzare la verifica;

- a semplificare ed automatizzare la pianificazione e la documentazione della stessa;
- a semplificare ed automatizzare la comunicazione interna tra i membri del gruppo;
- a gestire test ed analisi sul codice.

2.5.3 Risorse hardware

- computer dotati di tutti i software descritti nel *Piano di qualifica* e nelle *Norme di progetto*;
- luoghi dove effettuare le riunioni del gruppo.

2.6 Strumenti

Le risorse software che si utilizzeranno durante il processo di verifica sono:

- ***TexMaker*:** un ambiente grafico Open-Source per LaTeXcross-platform, permette la compilazione rapida e la visualizzazione del PDF generato;
- ***Scalastyle*:** analizzatore statico che rileva potenziali problemi nel codice (`https://github.com/scalastyle/scalastyle`);
- ***Scapegoat*:** un altro analizzatore statico che si concentra maggiormente sugli standard di stile e di coding (`https://github.com/sksamuel/scapegoat`);
- ***CLOC (Count Lines Of Code)*:** misura alcune metriche riguardanti il codice sorgente in vari linguaggi, tra cui *Scala* (`cloc.sourceforge.net`);
- ***ScalaTest*:** framework per i test su Scala (`http://www.scalatest.org/`).

2.7 Analisi

2.7.1 Tecniche per l'analisi statica

L'analisi statica non richiede l'esecuzione del codice in oggetto, ed è quindi applicabile sia alla documentazione che al codice. Permette di individuare errori ed anomalie al più presto possibile, scongiurandone la diffusione.
Essa può essere svolta in due modi distinti.

Walkthrough

Si svolge effettuando una lettura critica a pettine. Questa tecnica viene utilizzata prevalentemente nelle prime fasi del progetto, in cui non si ha ne una adeguata esperienza, ne uno storico degli errori più comuni che permetta una indagine più mirata. I Verificatori, tramite questa tecnica, saranno in grado di stilare una lista di errori più frequenti, potendo così applicare successivamente la tecnica *Inspection*. Il *Walkthrough* è una tecnica onerosa e richiede l'intervento di più persone. Dopo una fase iniziale in cui i Verificatori leggono il documento ed individuano potenziali errori essi devono essere discussi in una riunione con altri componenti del gruppo per accertare che non siano dei falsi positivi.

Inspection

È una tecnica molto meno onerosa. Consiste nel controllare alcune parti dei documenti che si sono rivelate maggiormente prone ad errori. Per ottenere questo risultato è necessario avere una lista di controllo che indichi quali sono le parti da controllare in maniera mirata. Essa viene stilata durante le fasi di *Walkthrough*. Un altro motivo per cui la *Inspection* è preferibile è il fatto che essa richiede l'intervento dei soli verificatori, che poi possono procedere alla correzione della maggior parte degli errori, oppure ad aprire un ticket riguardante quelli che non sono di immediata risoluzione.

Durante l'applicazione del *Walkthrough* ai documenti sono state riportate le tipologie di errori più frequenti, esse costituiscono quindi la lista di controllo per le verifiche ad *Inspection*:

- **Norme stilistiche:**
  - Nome del documento: non viene utilizzata la macro predisposta;
  - Versione del documento in prima pagina errata;
  - Immagini mancanti;
  - Spazi lasciati vuoti per aggiunte successive e non rimossi;
  - Mancanza di uniformità delle espressioni all'interno dello stesso documento;
  - Mancanze nella sezione dei riferimenti.
- **Italiano:**
  - Doppie;
  - Accenti.
- **LaTeX:**
  - mancanza dell'indice delle immagini e delle tabelle.
- **UML:**
  - incongruenze tra l'immagine contenente i diagrammi e la descrizione testuale della stessa;
  - errori nel testo delle immagini dovute a copia-incolla.

2.7.2 Tecniche per l'analisi dinamica

Questo tipo di analisi richiede una esecuzione di parte del programma, quindi ovviamente non applica ai documenti ma solo al codice. Il suo obiettivo è rilevare errori o difetti di implementazione mediante l'uso di test che devono essere necessariamente ripetibili: solo un test che produca lo stesso output partendo dallo stesso ambiente e lo stesso input può è capace di riscontrare problemi. L'attore che esegue un test deve definire a priori ed avere il pieno controllo su:

- **Ambiente:** insieme di hardware a software come sistema operativo e altri programmi o processi in esecuzione;

- **Specifiche:** definizione degli input e dei relativi output attesi, che sono ripetibili in quanto si postula di essere in un ambiente deterministico;
- **Procedure:** descrizione delle azioni compiute dall'attore (umano o computer che sia) per arrivare allo stato iniziale, far partire l'esecuzione, inserire gli input specificati e verificare che l'output sia uguale a quello atteso.

2.8 Misure e metriche

2.8.1 Metriche per i processi

2.8.2 Metriche per i documenti

2.8.3 Metriche per il software

2.9 Metodi

2.9.1 Analisi dei processi

2.9.2 Analisi dei documenti

Gestione amministrativa della revisione

3.1 Comunicazione e risoluzione di anomalie

Una anomalia è una violazione da parte di un documento, o unità di codice, di una o più delle seguenti condizioni:

- Conformità alla norme tipografiche o di codifica;
- Appartenenza al range di accettabilità per tutte le metriche descritte nella *Sezione 2*;
- Congruenza del prodotto con funzionalità indicate nell'*analisi dei requisiti*;
- Congruenza del codice con il design del prodotto.

Se un *Verificatore* dovesse trovare una anomalia egli è tenuto ad aprire un sotto-ticket all'interno della task-list a lui assegnata. Nel caso la risoluzione del ticket avesse la necessità di essere strutturato in sotto-attività sarà compito del *Responsabile* aprire una nuova task-list ed assegnarla alle figure coinvolte.

3.2 Procedure di controllo qualità per i processi

La qualità del processo viene garantita da:

- **Pianificazione:** i processi devono essere pianificati nel dettaglio, in maniera da determinare i punti e le tempistiche in cui effettuare controlli;
- **Controllo:** i controlli pianificati devono essere eseguiti in maniera oggettiva e neutrale, quindi con strumenti automatici ovunque possibile;
- **Miglioramento continuo:** è garantita dall'applicazione del principio *PDCA*.

3.3 Procedure di controllo qualità per il prodotto

La qualità del prodotto viene garantita da:

- **Comprensione ed analisi del dominio**;

- **Verifica:** determina che l'output di una fase sia consistente, completo e corretto. Deve essere eseguita costantemente per tutta la durata del progetto, ma cercando di essere minimamente invasiva;
- **Validazione:** conferma oggettivamente che il prodotto sia conforme alle aspettative;
- **Quality Assurance:** garantisce il raggiungimento degli obiettivi di qualità, in maniera preventiva. In questo modo si riduce drasticamente il ricorso a tecniche retrospettive, e con esse si riducono le iterazioni.

Resoconto delle attività di verifica

4.1 Riassunto delle attività di verifica

4.2 Tracciamento componenti - requisiti

4.3 Dettaglio delle verifiche tramite analisi

4.3.1 Processi

4.3.2 Documenti

4.4 Dettaglio dell'esito delle revisioni