SweeneyThreads

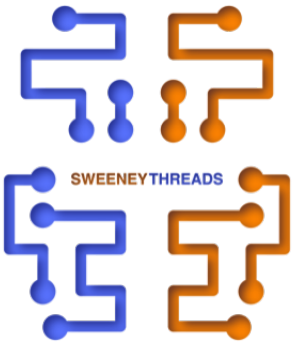
Actorbase

a NoSQL DB based on the Actor model

Glossario

Redattori:
Biggeri Mattia
Nicoletti Luca

Approvazione:
Bonato Paolo
Verifica:
Bonato Paolo



Versione 3.0.0

16 maggio 2016

Indice

Diario delle modifiche

Versione	Data	Autore	Descrizione
3.0.0	2016-05-16	*Responsabile* Bonato Paolo	Approvazione documento
2.0.1	2016-05-13	*Progettista* Biggeri Mattia	Aggiunto indice
2.0.0	2016-04-11	*Responsabile* Maino Elia	Documento approvato
1.4.0	2016-04-11	*Verificatore* Bonato Paolo	Verificato il documento
1.3.4	2016-04-11	*Responsabile* Maino Elia	Aggiunta di voci relative agli attori definiti nel capitolato e termini di progettazione.
1.3.3	2016-04-10	*Progettista* Nicoletti Luca	Separata tabella del diario della modifiche in file esterno.
1.3.2	2016-03-09	*Progettista* Nicoletti Luca	Inseriti i termini Stub e Driver. Corretta la definizione di Attore.
1.3.1	2016-02-23	*Responsabile* Nicoletti Luca	Rivista l'intera tabella del diario delle modifiche secondo le indicazioni
1.3.0	2016-02-19	*Responsabile* Nicoletti Luca	Prime modifiche post consegna RR: tolto il comando per la creazione di paragrafi da noi introdotto, sostituiti tutti i "paragrafi" con sezioni; cambiata classe del documento da "report" ad "article" come consigliato nella correzione.
1.2.0	2016-01-18	*Responsabile* Padovan Tommasin	Documento approvato, pronto alla consegna
1.1.0	2016-01-18	*Verificatore* Tommasin Davide	Verificato il documento
1.0.3	2016-01-17	*Analista* Maino Elia	Inserimento voci
1.0.2	2016-01-16	*Amministratore* Nicoletti Luca	Riordinato il documento, tolto indice
1.0.1	2016-01-16	*Amministratore* Nicoletti Luca	Sistemati elenchi puntati interni
1.0.0	2016-01-16	*Analista* Maino Elia	Stesura scheletro documento e Introduzione

Tabella 1: Diario delle modifiche

1 A

- **Actor:** Actor si riferisce agli attori all'interno del progetto Actorbase, non al ruolo dell'utente nell'interazione con il sistema. Un Actor può essere StoreKeeper, Ninja, StoreFinder, Manager.
- **Akka:** Libreria per il linguaggio Scala che definisce un sistema di attori estendibili con nuove caratteristiche, permette di gestire concorrenza e distribuzione in modo trasparente.
- **Ambiente di lavoro:** Quanto serve ai processi di produzione, è fatto di persone, ruoli e procedure, infrastrutture.
- **Application Programming Interface:** Una API esprimeun comonente software in termini delle sue operazioni, dei suoi input e dei suoi output. Permette di cambiare l'implementazione del componente senza compromettere l'interfaccia.
- **API:** Vedi Application Programming Interface.
- **Approccio sistematico:** Sapere perché si fa una cosa, approcciare un problema conoscendo i passi da svolgere. Essere sistematici porta sia all'efficacia che all'efficienza.
- **Approccio disciplinato:** Approccio basato su delle regole fissate in partenza da rispettare in ogni situazione al fine di rendere possibile l'organizzazione di un lavoro di gruppo e la pianificazione di tempi e costi del lavoro.
- **Approccio quantificabile:** Ottenere un sistema in cui si riesca a dare un costo affidabile alle operazioni da svolgere. La disciplina e la sistematicità rendono possibile la quantificazione del lavoro.
- **Architettura SW:** Insieme di regole che danno una struttura al software. Deve avere le seguenti qualità:
  - Sufficienza
  - Comprensibilità
  - Modularità
  - Robustezza
  - Flessibilità
  - Riusabilità
  - Efficienza
  - Affidabiltà
  - Disponibilità
  - Security
  - Safety
  - Semplicità
  - Incapsulazione
  - Coesione
  - Basso accoppiamento
- **Attore:** ruolo dell'utente nell'interazione con il sitema, svolgono il caso d'uso per raggiungere l'obiettivo, sono un buon mezzo di individuazione dei casi d'uso, non includono dettagli implementativi, sui modi di interazione. Vanno identificati secondo un processo.

2 B

- **Baseline:** E' l'insieme di Configuration Item consolidato ad una specifica Milestone. Rappresenta cioè un punto dello sviluppo verificato, approvato e garantito.É una configurazione a cui si può tornare senza perdite in caso di fallimento.
- **Best practice:** Prassi (modo di fare) che per esperienza e per studio abbia mostrato di garantire i migliori risultati in circostanze note e specifiche.
- **Brainstorming:** Tecnica di analisi dei requisiti che prevede discussioni creative e paritetiche con o senza il committente. Solitamente necessita di un facilitatore, ovvero un partecipante neutrale che fa si che si rispettino le regole, ed un rapporteur, ovvero colui che tiene nota della discussione riportando solo le parti importanti producendo così le "miniature".

3 C

- **Casi d'uso:** Un caso d'uso é la descrizione di un'interazione tra uno o più attori esterni ed il sistema, al fine di effettuare una determinata operazione. I casi d'uso sono utili per identificare requisiti in quanto descrivono i modi in cui il prodotto verrà utilizzato.
- **Commit:** Un commit si effettua quando si copiano le modifiche eseguite sui file locali nella directory del repository (il software di controllo versione controlla quali file sono stati modificati dall'ultima sincronizzazione).
- **Configuration:** E' un sistema in un suo istante; cambia a seconda di alcune svolte importanti (milestone).
- **Configurazione:** Vedi Configuration.
- **Ciclo di vita del software:** Gli stati che il prodotto assume dal concepimento al ritiro.
- **Coesione:** Fa si che le attività di processo siano ben definite e correlate tra loro (e così anche i compiti al loro interno).
- **Ciclo PDCA:** Organizzazione dei processi basata sul principio del miglioramento continuo
  - **Pianificare (plan):** Definire attività, scadenze, responsabilità, risorse utili a raggiungere specifici obiettivi di miglioramento.
  - **Eseguire (do):** Eseguire le attività secondo i piani.
  - **Valutare (check):** Verificare l'esito delle azioni di miglioramento rispetto alle attese.
  - **Agire (act):** Applicare soluzioni correttive alle carenze rilevate.
- **CLI:** Vedi Command Line Interface.
- **Command Line Interface:** Interfaccia utente da riga di comando, l'utente interagisce con essa come se interagisse con un terminale: inserisce testo e preme invio per confermare l'input, riceve testo come risposta dal sistema.

4 D

- **Design pattern:** Soluzione progettuale ad un problema ricorrente.
- **Driver:** Il driver simula un'unità chiamante. È l'opposto dello Stub, che invece simula un'unità chiamata.

5 E

- **Efficacia:** È determinata dal grado di conformità del prodotto rispetto alle norme vigenti e agli obiettivi prefissati. Un prodotto software finale è da considerarsi tanto più efficace quanto più si avvicina agli obiettivi fissati inizialmente. Perseguire l'efficacia equivale a garantire la qualità del prodotto.
- **Efficienza:** È inversamente proporzionale alla quantità di risorse impiegate nell'esecuzione delle attività richieste. Perseguire l'efficienza equivale a contenere i costi e i tempi di produzione.
- **Elicitare:** Riferito a concetti o informazioni, ottenerli mediante domande o altri stimoli.

6 F

- **Fase:** Durata temporale entro uno stato di ciclo di vita o una transazione tra essi.

7 G

- **Gantt (diagramma di):** Diagramma per la dislocazione temporale delle attività. Per ogni attività indica durata temporale, sequenzialità/parallelismo con le altre. Permette di confrontare le stime con progressi reali.

8 I

- **Incremento:** Procedere per incrementi significa aggiungere (o togliere) a un impianto base, la caratteristica fondamentale dell'incremento è l'utilizzare un solo tipo di operazione, non si torna sui propri passi come per l'iterazione. Ha caratteristiche preferibili perché permette di stabilire una data di termine.
- **Ingegneria (Engineering):** L'applicazione di principi scientifici e matematici a fini pratici (civili, sociali, prodotti di consumo).
- **Ingegneria del software:** Lo studio e l'applicazione dell'ingegneria al design, allo sviluppo e al mantenimento del software. L'approccio utilizzato deve essere sistematico, disciplinato e quantificabile. Non è una branchia dell'informatica ma un interfacciarsi di diverse discipline dell'informatica, matematica, economia, ingegneria, psicologia e sociologia.
- **Ingegneria dei requisiti:** È parte integrante dell'ingegneria di sistema e corrisponde all'insieme di attività necessarie per il trattamento sistematico dei requisiti. Tali attività si suddividono in due insiemi principali:
  - Analisi
  - Specifica di verifica e validazione
- **ISO 8601:2004:** (Data elements and interchange formats - Information interchange - Representation of dates and times) E' lo standard di riferimento per la rappresentazione di date ed orari.
- **ISO/IEC 12207:** ISO 12207 è uno standard dell'ISO per la gestione del Ciclo di vita del software. Si propone di diventare lo standard di riferimento definendo tutte le attività svolte nel processo di sviluppo e mantenimento del software. Lo standard ISO 12207 stabilisce un processo di ciclo di vita del software, compreso processi ed attività relative alle specifiche ed alla configurazione di un sistema. Ad ogni processo corrisponde un insieme di risultati (outcome).
- **Iterazione:** Fare un passo indietro per fare un passo avanti, operare raffinamenti o rivisitazioni in maniera ciclica finché certe condizioni non vengono soddisfatte. L'iterazione ha caratteristiche molto pericolose perché è una operazione distruttiva (elimina quanto fatto e sovrascrive) e potenzialmente può durare all'infinito: non sa garantire un termine al processo di sviluppo, quindi è non quantificabile.

9 J

- **JVM:** La macchina virtuale Java, detta anche Java Virtual Machine o JVM, è il componente della piattaforma Java che esegue i programmi tradotti in bytecode dopo una prima compilazione.

10 L

- **LaTeX:** LaTeX è un linguaggio di markup usato per la preparazione di testi basato sul programma di composizione tipografica TEX.

11 M

- **Main:** Tipologia di attore che si occupa di fornire un punto di accesso primario al sistema di attori sottostante.
- **Manager:** Tipologia di attore che si occupa di gestire altri attori e i vincoli presenti su di essi.
- **Manutenibilità/Manutenzione:** Qualità fondamentale del sw che si presenta in diverse forme:
  - **Correttiva:** per correggere difetti eventualmente rilevati
  - **Adattativa:** per adattare il sistema alla variazione dei requisiti
  - **Evolutiva:** per aggiungere funzionalità al sistema

  Più risulta semplice eseguire queste operazioni più si considera il prodotto software mantenibile.
- **Milestone:** Letteralmente pietre miliari, vengono tipicamente utilizzate nella pianificazione e gestione di progetti complessi al fine di importanti traguardi intermedi nello svolgimento del progetto stesso. Molto spesso sono rappresentate da eventi, cioè da attività con durata zero o di un giorno, e vengono evidenziate in maniera diversa dalle altre attività nell'ambito dei documenti di progetto.
- **Modello di ciclo di vita:** Descrive come i processi si relazionano tra loro nel tempo rispetto agli stati di ciclo di vita, è la base concettuale intorno alla quale pianificare, organizzare, eseguire e controllare lo svolgimento delle attività necessarie. Esistono diversi possibili cicli di vita, non diversi per numero e significato degli stati ma diversi per le transizioni tra essi e le relative regole di attivazione.
- **Modularità:** Fa si che i processi siano tra loro relazionati in modo chiaro e distinto.

12 N

- **NoSQL:** NoSQL è un movimento che promuove sistemi software dove la persistenza dei dati è caratterizzata dal fatto di non utilizzare il modello relazionale, di solito usato dai database tradizionali (RDBMS).
- **Ninja:** Tipologia di attore che si occupa di mantenere in memoria principale una copia di una porzione dei dati.

13 O

- **Opportunità:** Un'offerta relativa a un prodotto software non implica necessariamente un'opportunità (sw già esistente, progetto irrealizzabile, ...) dunque è compito del software engineer valutare la presenza di opportunità al momento di accettare o meno la commissione.

14 P

- **PERT (diagramma di):** Acronimo di Program Evaluation and Review Technique. È una tecnica di project management (un diagramma) nato per ridurre i tempi ed i costi per la progettazione. Con questa tecnica si tengono sotto controllo le attività di un progetto utilizzando una rappresentazione reticolare che aiuta ad individuare il cammino critico e a ragionare sulle scadenze di un progetto.
- **Problematiche essenziali:** Problematiche del software relative al suo sviluppo concettuale e dunque non eliminabili dal progresso tecnologico.
- **Problematiche accidentali:** Problematiche dello sviluppo software dovute all'inadeguatezza temporanea delle tecnologie di supporto, più legate agli strumenti che ai concetti, tali problemi possono essere risolti/eliminati grazie alle evoluzioni tecnologiche.
- **Prodotto software:** Il software vero e proprio associato eventualmente alla relativa documentazione.
- **Processo di ciclo di vita:** Un processo di ciclo di vita specifica le attività che vanno svolte per causare transizioni nel ciclo di vita di un prodotto sw.
- **Processo software (Way of working):** Il quadro metodologico, normativo e strategico delle attività di progetto per organizzare al meglio le attività necessarie all'interno di vincoli dati di tempo, di risorse e di obiettivi. Un processo è un insieme di attività correlate e coese che trasformano ingressi in uscite secondo regole fissate, consumando risorse nel farlo (Glossario ISO 9000).
- **Progetto software:** L'insieme delle attività di produzione di un software:
  - Pianificazione
  - Analisi dei requisiti
  - Progettazione
  - Realizzazione
  - Verifica e validazione
  - Manutenzione
- **Progetto didattico:** Sequenza di revisioni di avanzamento come principale modalità di interazione basata sulla logica di relazione cliente-fornitore. Ciascuna revisione di avanzamento richiede diversi adempimenti formali e valuta il raggiungimento di uno specifico stato di avanzamento.
- **Prototipo:** Serve per provare e scegliere soluzioni a lavoro in corso, possono essere di due tipi: "usa e getta" (iterazione) o "baseline" (fornire stati di incremento). L'utilizzo di prototipi comporta in ogni caso un costo che se supera il beneficio ottenibile ne rende improduttivo l'uso.

15 R

- **Requisito:** Una delle primissime attività da svolgere per quanto riguarda lo sviluppo è la definizione dei requisiti del prodotto, un sistema software sarà considerato efficace se e solo se si dimostrerà in grado di soddisfare i requisiti. Inoltre la definizione dei requisiti fornisce subito un'idea sulla realizzabilità del prodotto (l'opportunità è sostenibile?) e sui tempi di sviluppo.

  1. **Condizione (capability):** necessaria a un utente per risolvere un problema o raggiungere un obiettivo [lato bisogno]
  2. **Condizione (capability):** che deve essere soddisfatta o posseduta da un sistema per adempiere a un obbligo (contratto, standard, specifica, documento formale) [lato soluzione]

  Spesso un requisito lato utente si trasforma in molti requisiti lato soluzione.
- **Riuso:** Molte volte nell'ingegneria del software risulta decisamente produttiva la tecnica del riuso a patto che lo si faccia in maniera sistematica e consapevole. Adattare componenti preesistenti alle proprie esigenze se fatto con criterio può far risparmiare al team molte risorse, aumentando l'efficienza.
- **Ruolo:** Funzione aziendale assegnata a progetto.

16 S

- **Scala:** Linguaggio di programmazione ad oggetti e funzionale utilizzato dal gruppo.
- **Servizio:** Mezzo per fornire valore all'utente, agevolando il raggiungimento dei suoi obiettivi, sollevandolo da costi e rischi.
- **Scala:** Scala (da Scalable Language) è un linguaggio di programmazione di tipo general-purpose multi-paradigma studiato per integrare le caratteristiche e funzionalità dei linguaggi orientati agli oggetti e dei linguaggi funzionali. La compilazione di codice sorgente Scala produce Java bytecode per l'esecuzione su una JVM.
- **Software engineer:** Realizza parte di un sistema complesso con la consapevolezza che potrà essere usato, completato e modificato da altri. Deve guardare e comprendere il quadro generale nel quale il sistema cui contribuisce si colloca, deve operare compromessi intelligenti e lungimiranti tra visioni e spinte contrapposte.
- **Standard di processo:** Sono fondamentalmente di due tipi: Standard come modello di azione e Standard come modello di valutazione. I primi definiscono e impongono/propongono delle procedure da seguire, i secondi cercano di identificare una "best practice" da usare come linea di valutazione dell'operato.
- **Stakeholder (People):** L'insieme di persone a vario titolo coinvolte nel ciclo di vita del SW con influenza sul prodotto.
  - Business management
  - Project management
  - Team di sviluppo
  - Clienti
  - Utenti Finali
- **Storefinder:** Tipologia di attore che si occupa di indicizzare gli attori che gestiscono i dati.
- **Storekeeper:** Tipologia di attore che si occupa di mantenere in memoria principale una porzione dei dati.
- **Stub:** porzione di codice utilizzata in sostituzione di altre funzionalità software. Uno stub può simulare il comportamento di codice esistente e temporaneo sostituto di codice ancora da sviluppare.

17 T

- **Task:** Compito organizzativo ottenuto dalla frantumazione di un processo. Generalmente è delle dimensioni adeguate per essere svolto da una solo persona ed è parallelizzabile.

18 U

- **UML:** Famiglia di notazioni grafiche che si basano su un singolo meta-modello e servono a supportare la descrizione e il progetto dei sistemi software
- **UTF-8:** (Unicode Transformation Format, 8 bit) è una codifica dei caratteri Unicode in sequenze di lunghezza variabile di byte, creata da Rob Pike e Ken Thompson. UTF-8 usa gruppi di byte per rappresentare i caratteri Unicode.

19 V

- **Validazione:** Serve ad accertare che il prodotto realizzato corrisponda alle attese (Did I built the right system?) ed è rivolta ai prodotti finali.
- **Verifica:** Serve ad accertare che l'esecuzione delle attività di processo non abbia introdotto errori. Si confronta ciò che si sta facendo con le regole che si devono rispettare. (Am I building the system right?)

20 W

- **WBS:** Vedi Work Breakdown Structrure.
- **Work Breakdown Structure:** Identifica l'elenco di tutte le attività di un progetto. Le WBS sono usate dal Project manager come supporto per le attività di cui è responsabile.