

Norme di Progetto

Contatti: swateng.team@gmail.com

Versione: 0.6



Registro delle Modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.6	24-11-2023	Aggiunta sezione "Analisi requisiti"	Riccardo Costantin	Amministratore
0.5	17-11-2023	Aggiunta specifica inserimento termini nel Glossario	Matteo Rango	Amministratore
0.4	14-11-2023	Aggiunta sezione "Processi Primari"; Chiarito workflow documentazione e codice	Matteo Rango	Amministratore
0.3	13-11-2023	Aggiunte sezioni "Oggetti Esterni al Repository" e "Tracciamento del Tempo Speso"; modificata organizzazione logica	Matteo Rango	Amministratore
0.2	08-11-2023	Aggiunta sezione "Processi di Supporto" e "Processi Organizzativi"	Matteo Rango	Amministratore
0.1	07-11-2023	Aggiunta sezione "Introduzione"	Matteo Rango	Amministratore

Indice

1 Introduzione	4
1.1 Glossario	4
1.2 Scopo del Documento	4
2 Processi Primari	5
2.1 Sviluppo	5
2.1.1 Codifica e Verifica	5
2.1.2 Analisi dei Requisiti	5
2.1.2.1 Identificazione caso d'uso	6
2.1.2.2 Struttura dei casi d'uso	6
2.1.2.3 Requisiti	6
2.1.2.4 Identificazione Requisiti	7
3 Processi di Supporto	8
3.1 Documentazione	8
3.1.1 Nomenclatura	8
3.1.2 Versionamento	8
3.1.3 Creazione e Modifica di un Documento	8
3.2 Gestione della Configurazione	9
3.2.1 Issue Tracking System	9
3.2.2 Strumento di Condivisione	9
3.2.3 Tracciamento del Tempo Speso	10
3.3 Verifica	10
3.3.1 Elementi Interni al Repository	10
3.3.2 Elementi Esterni al Repository	10
4 Processi Organizzativi	10
4.1 Gestione Organizzativa	10
4.1.1 Decisioni	10

1 Introduzione

1.1 Glossario

Al fine di evitare possibili ambiguità relative al linguaggio utilizzato nei documenti, viene fornito il *Glossario*, nel quale sono presenti tutte le definizioni di termini aventi uno specifico significato che vuole essere disambiguato. Tali termini, sono scritti in *corsivo* e marcati con una _G a pedice. Un'attività che comprende l'inserimento di un termine di glossario può considerarsi conclusa, nella sua interezza, solo nel momento in cui il termine viene scritto e spiegato nel *Glossario*.

1.2 Scopo del Documento

Il presente documento ha come scopo la definizione delle *best practices*_G e del *way of working*_G che ogni componente del *SWAT Engineering* ha l'obbligo di rispettare durante l'intero svolgimento del progetto. In questo modo si prova a garantire un metodo di lavoro omogeneo, verificabile e migliorabile nel tempo.

2 Processi Primari

2.1 Sviluppo

2.1.1 Codifica e Verifica

Come sistema di controllo di versione si utilizza *Git* all'interno del servizio online *Github*.

All'interno dei repository si utilizza il *Rebase workflow*_G come metodo di gestione, con piccoli accorgimenti, segnalati nelle apposite sezioni.

In particolare si nota la differenza tra il repository di documentazione, chiamato *Docs* e quello di progetto (codice) chiamato *InnovaCity*: per entrambi si utilizza il *Rebase workflow*, con l'utilizzo dei *Feature branch* per separare logicamente il lavoro da svolgere. Nel primo repository però, i *Feature branch* si derivano direttamente dal *main*, mentre nel secondo si derivano dal branch *dev*. Questo impone quindi che, prima di andare ad effettuare la chiusura di una *Pull request*, si vada ad effettuare un *rebase* del branch di derivazione, per rendere il nostro branch di sviluppo aggiornato rispetto alla base.

I *Feature branch* vengono aperti a partire dalle issue create nell' *Issue Tracking System* (vedi Sezione 3.2.1). Si procede poi ad associare una *Pull request*, a una o più issue collegate tra loro, per effettuare la verifica.

Nel caso del repository *InnovaCity* il branch *main* viene utilizzato per la pubblicazione di cambiamenti *major*, cioè quando si è implementato un numero di funzionalità significativo all'avanzamento del progetto. In quel caso è il responsabile che esegue l'approvazione finale.

2.1.2 Analisi dei Requisiti

L'analisi dei requisiti, condotta dagli Analisti, rappresenta un'attività preliminare nello sviluppo di un sistema software. Questa attività mira a definire le funzionalità che il nuovo prodotto deve offrire. L'obiettivo principale è quello di identificare, in modo chiaro e completo, i requisiti che il software sviluppato deve soddisfare, al fine di rispondere adeguatamente alle esigenze degli utenti e della Proponente.

L'analisi si compone di varie fasi:

- Scopo del prodotto: si devono soddisfare le esigenze della Proponente;
- Definizione degli attori: entità o persone che interagiscono con il sistema_G
- Definizione dei casi d'uso: rappresentazione di uno scenario specifico che descrive come un attore interagisce con il sistema;

- Definizione di requisiti obbligatori, opzionali, di vincolo;
- Confronti interni ed esterni.

La collaborazione con la Proponente è essenziale per comprendere in modo accurato e approfondire dettagliatamente il processo di realizzazione del prodotto.

2.1.2.1 Identificazione caso d'uso

I casi d'uso sono identificati nel seguente modo:

UC[Numero].[Numero sottocaso] [Titolo]

legenda:

- **Numero:** numero del caso d'uso;
- **Numero sottocaso:** numero del sottocaso del caso d'uso generale se presente;
- **Titolo:** breve titolo che descrive il contesto del caso d'uso in questione.

2.1.2.2 Struttura dei casi d'uso

I casi d'uso sono strutturati nel seguente modo:

- Attore;
- Precondizioni;
- Postcondizioni;
- Scenario Principale;
- Scenari Secondari (ove necessario);
- Estensioni, se presenti;
- Specializzazioni, se presenti.

2.1.2.3 Requisiti

I requisiti trovati vengono classificati nei seguenti modi:

- **Funzionali**

un requisito funzionale specifica una funzionalità che il sistema deve essere in grado di svolgere;

- **Qualità**

Un requisito di qualità stabilisce gli standard e i criteri che il sistema deve soddisfare per garantire prestazioni, affidabilità, sicurezza e altri aspetti relativi alla qualità;

- **Vincolo**

Un requisito di vincolo è una restrizione o una condizione imposta al progetto.

2.1.2.4 Identificazione Requisiti

I requisiti trovati avranno un codice univoco con la seguente sintassi:

R[Importanza][Tipo][Numero]

legenda:

- **Importanza:**

- O -> se requisito obbligatorio;
- D -> se requisito desiderabile;
- P -> se requisito opzionale.

- **Tipo:**

- F -> se funzionale;
- Q -> se qualità;
- V -> se di vincolo.

- **Numero:** per ogni requisito aggiunto il numero viene incrementato.

3 Processi di Supporto

3.1 Documentazione

3.1.1 Nomenclatura

La consueta nomenclatura per i documenti si ottiene unendo, attraverso un underscore (`_`), il nome del file in *CamelCase* senza spazi (`NomeDelFile`) e la sua versione (`3.5`). Ad esempio `NormeDiProgetto_2.6.pdf`. Nel caso di documenti il cui nome contiene una data, essa si inserisce dopo il nome, ma prima della versione, sempre separandolo con gli underscore, nella forma `ggmmaa` senza separatori tra i singoli componenti della data: `gg` rappresenta il giorno, sempre scritto in due cifre, allo stesso modo `mm` rappresenta il mese, mentre l'anno è rappresentato da `aa`, corrispondente alle ultime due cifre dell'anno corrente.

3.1.2 Versionamento

Il versionamento avviene secondo il seguente formato **x.y**:

- **y** si incrementa una volta effettuata una modifica e la sua conseguente verifica;
- **x** si incrementa quando si effettua la modifica definitiva in vista di una verifica di avanzamento, questo comporta l'azzeramento di **y**.

Due modifiche, fatte in momenti diversi, differiscono l'una dall'altra solo se hanno scopi diversi. Ad esempio non è necessario incrementare la versione se viene fatta una modifica alla stessa sezione in due giorni differenti; anche se faccio una modifica, ed essa non viene approvata, non è necessario incrementare la versione con le nuove modifiche proposte dal/dai verificatore/i, dal momento che modifica e verifica "viaggiano" parallelamente.

3.1.3 Creazione e Modifica di un Documento

Ogni documento segue le fasi del seguente *workflow_G*:

1. Si crea un branch per lo sviluppo del documento nell'apposita repository *Docs* e si mette in uso.
2. Si copia dall'apposita repository *Templates* il template relativo al file che si deve redigere, e lo si inserisce nella cartella appropriata.
3. Si redige il documento o una sua sezione. Nel caso di documenti nuovi, in cui è necessario un elevato parallelismo di lavoro, è possibile usare *Google Drive* per la prima stesura e successivamente caricare il documento all'interno del branch.
4. Nel file `changelog.typ` si aggiunge una riga **in coda**, secondo il seguente formato: `<versione>,<data-modifica>,<descrizione-modifica>,<nome-`

autore>, <ruolo-autore>; la versione segue le regole descritte nella Sezione 3.1.2.

5. Si esegue la commit sul branch creato.
6. Si apre una *pull request* dal branch appena creato verso il branch *main*: se il documento non è pronto per la verifica, ma ha bisogno di ulteriori modifiche, si apre la *pull request* in modalità *draft*, altrimenti in modalità normale, spostando la issue nell'apposita corsia *Ready to Review*.
7. Per ulteriori modifiche richieste dal/dai verificatore/i si ripetono i punti, **in ordine**, dal punto 3 al punto 5.
8. Si elimina, **solo quando la pull request viene chiusa o risolta**, il branch creato.

La modifica di un documento avviene allo stesso modo, saltando il passo 2. Ogni cambiamento di stato è accompagnato dal conseguente movimento della issue, associata allo sviluppo, attraverso le diverse corsie dell'issue tracking system.

3.2 Gestione della Configurazione

3.2.1 Issue Tracking System

Come ITS si utilizza *Github* che, attraverso le funzioni di *Project*, *Issues* e *Pull requests*, garantisce una struttura all'organizzazione di progetto.

Le *Corsie di Stato_G* descrivono lo stato attuale delle attività, all'interno del *Project* nell'*ITS* sono presenti:

Backlog attività individuate da svolgere.

Ready attività individuate per il completamento durante il prossimo *sprint_G*.

In Progress attività che sono in corso d'opera da parte dei redattori.

Ready to Review attività svolte che sono pronte per essere verificate.

In Review attività in corso di verifica da parte dei verificatori.

Done attività le cui modifiche sono state verificate e accettate.

3.2.2 Strumento di Condivisione

Per la condivisione veloce o la creazione di bozze si utilizza *Google Drive*. Uno dei suoi principali casi d'uso consiste nella collaborazione in tempo reale nella stesura di sezioni testuali ampie, da inserire successivamente nella documentazione (questo risulta particolarmente utile nel momento in cui il documento è alla sua prima stesura). Viene inoltre utilizzato come sistema per l'immagazzinamento di conoscenze acquisite durante lo svolgimento del progetto.

3.2.3 Tracciamento del Tempo Speso

Al fine di tracciare il tempo speso nel corso del progetto, nei diversi ruoli, si userà uno spreadsheet appositamente creato, disponibile all'interno di *Google Drive* dove, a fine giornata, ogni membro del team andrà ad inserire le proprie ore **produttive** svolte quel giorno, secondo la sua miglior stima del rapporto tra ore di orologio e ore produttive. Si inserisce *una sola* riga per ogni giornata e nella descrizione si andranno ad inserire dei brevi titoli rappresentativi delle attività svolte.

3.3 Verifica

3.3.1 Elementi Interni al Repository

La verifica del documento avviene tramite apposito metodo nell'ITS, attraverso la *Pull request*, indicando i punti in cui si richiede la modifica, il motivo della richiesta e una proposta se necessario. Anche in questo caso i verificatori si occupano di spostare la issue di riferimento nelle corsie appropriate, chiudendola se la verifica è terminata con successo, o spostandola nuovamente nella corsia *In Progress* in caso vengano richieste altre modifiche.

3.3.2 Elementi Esterni al Repository

Potrebbero esservi delle issue aperte all'interno dell'ITS che non hanno un corrispondente documento o prodotto in generale, all'interno del repository, ma che fungono come attività di gestione. Per queste, il ciclo di vita segue il normale flusso attraverso i diversi stati elencati nella Sezione 3.2.1. La verifica viene effettuata attraverso i commenti della issue stessa, che avranno la seguente forma:

- caso *richiesta cambiamenti*:

[REV]

- richiesta 1;
- richiesta 2;

- caso *approvazione*:

[REV] done

4 Processi Organizzativi

4.1 Gestione Organizzativa

4.1.1 Decisioni

Per poter prendere una qualsiasi decisione è necessario vi siano due condizioni:

1. Si deve raggiungere il $quorum_G$ di quattro persone su sei;
2. La decisione deve essere verbalizzata e motivata.