



Winning Software Solution

winningsoftwaresolution@gmail.com

ShopChain

SyncLab

Norme di progetto

Informazioni

<i>Redattori</i>	Elia Scandaletti
	Raffaele Oliviero
	Giovanni Cocco
	Alberto Nicoletti
<i>Versione</i>	2.0.0
<i>Uso</i>	interno

Descrizione

Questo documento contiene le procedure, gli strumenti e i criteri di qualità che verranno usati nel progetto.

Versione	Data	Persona	Attività	Descrizione
2.0.0	20/03/2022	Alberto Nicoletti	Redazione	Rifacimento documento e stesura 2.1
0.0.9	16/02/2022	Giovanni Cocco	Redazione	Stesura di 6.2
0.0.8	08/02/2022	Raffaele Oliviero	Redazione	Stesura di 5.1.7, 5.1.7 e 5.1.7
0.0.7	04/02/2022	Giovanni Cocco	Redazione	Stesure di 7.1
0.0.6	16/01/2022	Elia Scandaletti	Redazione	Adeguamento di 5.2 a decisioni collettive
0.0.5	09/01/2022	Raffaele Oliviero	Redazione	Stesura 5.1.7 e 5.2
0.0.4	09/01/2022	Raffaele Oliviero	Redazione	Ristrutturazione documento
0.0.3	22/12/2021	Elia Scandaletti	Redazione	Stesura 5.1.7
0.0.2	07/12/2021	Elia Scandaletti	Redazione	Inserimento norme da verbale del 01/12/2021
0.0.1	14/11/2021	Elia Scandaletti	Redazione	Stesura 5.3.1
0.0.0	13/11/2021	Elia Scandaletti	Redazione	Stesura 5.1

Contents

1	Introduzione	3
1.1	Scopo del documento	3
2	Processi Primari	3
2.1	Fornitura	3
2.2	Sviluppo	3
3	Processi di supporto	3
3.1	Documentazione	3
3.1.1	Strumenti Utilizzati	3
3.1.2	Struttura	3
3.1.3	Convenzioni utilizzate	4
3.2	Versionamento	4
3.3	Verifica	4
3.3.1	Regole Generali	4
3.3.2	Metodi di Verifica	5
3.4	Change Management	5
4	Processi organizzativi	5
4.1	Gestione dei processi	5
4.2	Miglioramento dei processi	5
5	norme vecchie	6
5.1	Documentazione	6
5.1.1	Strumenti utilizzati	6
5.1.2	Ciclo di vita	6
5.1.3	Versionamento	6
5.1.4	Organizzazione dei file	6
5.1.5	Struttura dei documenti	7
5.1.6	Convenzioni utilizzate	8
5.1.7	Documenti	8
5.2	Verifica e Validazione	10
5.3	Configurazione	10
5.3.1	Repository GitHub	10
6	(vecchio)Workflow di lavoro	11
6.1	Versionamento	11
6.2	Sviluppo Software	11
7	(vecchio)Tracciamento dei progressi	11
7.1	Revisioni di avanzamento interne	12

1 Introduzione

1.1 Scopo del documento

Lo scopo del presente documento è definire procedure, strumenti e criteri di qualità al fine di stabilire un Way of Working. Ogni membro del gruppo sarà tenuto a rispettare le indicazioni qui presentate.

2 Processi Primari

2.1 Fornitura

2.2 Sviluppo

3 Processi di supporto

3.1 Documentazione

3.1.1 Strumenti Utilizzati

Per la stesura dei documenti verrà utilizzato il linguaggio L^AT_EX. Ogni documento dovrà includere il file `docs/template/style.tex` che contiene indicazioni di stile. Ogni documento userà per la prima pagina il template `docs/template/front_page.tex`. Eventuali risorse che dovessero servire per il template, saranno nella stessa cartella.

3.1.2 Struttura

Ogni documento avrà:

- Una pagina di intestazione
- Una pagina di elenco delle versioni, dove ammesso
- Una pagina di indice, dove ammesso
- Il contenuto

Le sezioni avranno le seguenti caratteristiche:

Pagina di intestazione

La pagina di intestazione contiene:

- logo e nome del gruppo;
- nome del progetto e azienda cliente;
- contatto;
- titolo del documento;
- informazioni sul documento:
 - elenco dei redattori;
 - elenco dei revisori;
 - elenco dei responsabili;
 - versione, laddove prevista;
 - destinazione d'uso;
- breve descrizione.

La destinazione d'uso può essere “interno” o “esterno”.

Elenco delle versioni

L'elenco delle versioni è una tabella con le seguenti colonne:

- versione;
- data;
- persona;
- attività;
- descrizione.

Con persona si intende il membro del gruppo che svolge l'attività.

Le possibili attività su un documento sono “Redazione”, “Revisione” e “Approvazione”.

Le righe della tabella sono in ordine cronologico inverso.

Indice

L'indice è creato usando il comando `\tableofcontents`.

3.1.3 Convenzioni utilizzate

Riferimenti a file e cartelle

Per indicare il nome di un file o di una cartella si utilizza il `testo monospaziato`.

Il nome di una cartella termina sempre con `/`.

Stringhe e nomi

Le stringhe vengono scritte tra doppi apici “ ”.

Per indicare un parametro in un nome o una stringa si usa del testo tra parentesi angolari `< >`. Il parametro non può contenere spazi.

Per indicare una parte di nome o stringa opzionale si usa del testo tra parentesi quadre `[]`.

Riferimenti tra documenti

Per riferirsi a sezioni all'interno dello stesso documento si usa il comando `\ref`.

Per riferirsi a sezioni di un altro documento si usa il nome di quella sezione scritto in corsivo.

3.2 Versionamento

Per il versionamento si farà uso del Versionamento Semantico, secondo le seguenti linee guida, se non indicato diversamente successivamente.

Un qualsiasi scatto di versione può avvenire solo attraverso una previa verifica

3.3 Verifica

3.3.1 Regole Generali

- Il verificatore si occupa di controllare le Pull Request aperte dagli altri membri del gruppo
- Il verificatore non può aver contribuito ai prodotti che deve verificare.

- Il verificatore non può modificare i prodotti che verifica, ad eccezione di correzioni grammaticali.
- Il verificatore, qualora trovasse errori, è tenuto a commentare le PR, usando la funzione apposita di GitHub.

3.3.2 Metodi di Verifica

Il verificatore può far uso dei test automatici presenti nel *Piano di qualifica*. Per le sezioni di codice è consigliato un walkthrough del prodotto, similmente si dovrebbero chiedere a chi abbia scritto il codice eventuali delucidazioni su elementi poco chiari

3.4 Change Management

Nel caso si dovessero cambiare sezione del codice bisogna controllare che eventuali modifiche non vadano ad impattare altri elementi. Per evitare ciò bisogna fare uso del *Diagramma delle classi* presente nelle *Specifiche architetturali*.

4 Processi organizzativi

4.1 Gestione dei processi

4.2 Miglioramento dei processi

5 norme vecchie

5.1 Documentazione

5.1.1 Strumenti utilizzati

Per la stesura dei documenti verrà utilizzato il linguaggio L^AT_EX.

5.1.2 Ciclo di vita

Ogni documento passa per le seguenti fasi di vita:

- **Pianificazione:** il documento viene discusso e delineato per sommi capi sulla base delle necessità a cui deve rispondere;
- **Redazione:** un membro del gruppo, nel ruolo di redattore, scrive fisicamente il documento;
- **Revisione:** un membro del gruppo, nel ruolo di revisore, controlla che non siano presenti errori grammaticali e che sia aderente alle norme di progetto;
- **Approvazione:** un membro del gruppo, nel ruolo di responsabile, verifica che il contenuto del documento sia corretto e coerente con lo scopo dello stesso.

Non è possibile per un membro del gruppo coprire più ruoli nella gestione dello stesso documento.

Una volta che un documento è revisionato o approvato può comunque tornare in fase di stesura sulla base delle esigenze del momento.

5.1.3 Versionamento

Ogni documento è univocamente identificabile dal titolo e dalla versione. La versione è definita da tre numeri nel formato `<x>.<y>.<z>`.

La prima versione è la 0.0.0

Il numero `<z>` viene modificato ogni volta che viene effettuata una stesura del documento.

Il numero `<y>` viene modificato ogni volta che viene effettuata una revisione del documento.

Il numero `<x>` viene modificato ogni volta che viene effettuata un'approvazione del documento.

Se il numero `<x>` viene modificato, `<y>` e `<z>` tornano a 0.

Se il numero `<y>` viene modificato, `<z>` torna a 0.

I verbali sono esclusi dal sistema di versionamento poiché non possono essere modificati nel tempo.

5.1.4 Organizzazione dei file

Tutta la documentazione sarà contenuta all'interno della cartella `docs/`. La cartella è a sua volta divisa in due sottocartelle `docs/interni/` e `docs/esterni/`, che conterranno rispettivamente la documentazione interna al gruppo e quella da condividere con gli altri stakeholders.

Tutti i verbali saranno conservati nella sottocartella `interni/verbali/`.

Per ogni documento sarà presente una cartella che sarà chiamata `<nome>/`, dove `<nome>` è il nome del documento come specificato nella sezione 5.1.7.

Il file principale di ogni documento sarà all'interno della rispettiva cartella e sarà chiamato `<nome>.tex`, dove `<nome>` è il nome del file come definito nella sezione 5.1.7.

Eventuali altri file utilizzati nella stesura del documento saranno posizionati all'interno della stessa cartella.

5.1.5 Struttura dei documenti

Ogni documento avrà una pagina di intestazione e una pagina con l'elenco delle versioni, laddove il versionamento del documento è ammesso. Sarà inserita una pagina con l'indice, a meno di indicazione contraria nella sezione 5.1.7.

Pagina di intestazione

La pagina di intestazione contiene:

- logo e nome del gruppo;
- nome del progetto e azienda cliente;
- contatto;
- titolo del documento;
- informazioni sul documento:
 - elenco dei redattori;
 - elenco dei revisori;
 - elenco dei responsabili;
 - versione, laddove prevista;
 - destinazione d'uso;
- breve descrizione.

La destinazione d'uso può essere “interno” o “esterno”.

Ogni documento userà per la prima pagina il template `docs/template/front_page.tex`.

Eventuali risorse che dovessero servire per il template, saranno nella stessa cartella.

Elenco delle versioni

L'elenco delle versioni è una tabella con le seguenti colonne:

- versione;
- data;
- persona;
- attività;
- descrizione.

Con persona si intende il membro del gruppo che svolge l'attività.

Le possibili attività su un documento sono “Redazione”, “Revisione” e “Approvazione”.

Le righe della tabella sono in ordine cronologico inverso.

Stile

Ogni documento dovrà includere il file `docs/template/style.tex` che contiene indicazioni di stile.

5.1.6 Convenzioni utilizzate

Riferimenti a file e cartelle

Per indicare il nome di un file o di una cartella si utilizza il `testo monospaziato`.

Il nome di una cartella termina sempre con `/`.

Stringhe e nomi

Le stringhe vengono scritte tra doppi apici `" "`.

Per indicare un parametro in un nome o una stringa si usa del testo tra parentesi angolari `< >`. Il parametro non può contenere spazi.

Per indicare una parte di nome o stringa opzionale si usa del testo tra parentesi quadre `[]`.

Riferimenti tra documenti

Per riferirsi a sezioni all'interno dello stesso documento si usa il comando `\ref`.

Per riferirsi a sezioni di un altro documento si usa il nome di quella sezione scritto in corsivo.

5.1.7 Documenti

Norme di progetto

Scopo Lo scopo delle norme di progetto è definire procedure, strumenti e criteri di qualità al fine di stabilire un Way of Working. Ogni membro del gruppo sarà tenuto a rispettare le indicazioni lì presentate.

Titolo Il titolo del documento è "Norme di progetto".

Nome del file Il file sarà chiamato `norme_di_progetto.tex`.

Verbali

Scopo Lo scopo dei verbali è tenere traccia del dialogo interno ed esterno al gruppo e delle eventuali decisioni prese.

Titolo Ogni verbale sarà titolato "Verbale del `<data>` [con `<esterni>`]", dove `<data>` indica la data dell'incontro nel formato europeo. In caso all'incontro siano presenti persone esterne al gruppo, verrà usata anche la dicitura "con `<esterni>`", dove `<esterni>` indica i partecipanti alla riunione separati da virgole. Gli esterni possono essere identificati collettivamente tramite il nome dell'azienda o ente che rappresentano, tramite nome e cognome o tramite titolo e cognome.

Esempi:

- "Verbale del 09/11/2021";
- "Verbale del 28/10/2021 con SyncLab";
- "Verbale del 25/12/2021 con prof. Cardin".

Nome dei file I file saranno chiamati `<data>_<tipo>.tex`, dove `<data>` è la data dell'incontro in formato `<yyyy>_<mm>_<dd>` e `<tipo>` è I se non sono presenti persone esterne al gruppo, E altrimenti.

Indice Non è presente una pagina con l'indice.

Struttura Ogni verbale deve riportare:

- data, ora e durata;
- luogo;
- partecipanti;
- ordine del giorno;
- riassunto dei contenuti;
- eventuali impegni assunti.

Stesura All'inizio di ogni riunione verrà nominato un partecipante che redigerà il verbale e uno che lo approverà.

Chi redige il verbale ha 24 ore per completare il lavoro. Chi approva il verbale ha 24 ore per approvarlo o segnalare eventuali correzioni da apportare.

Nel secondo caso, il redattore avrà ulteriori 24 ore per adeguarsi alle indicazioni.

Piano di Qualifica

Scopo Lo scopo del *Piano di Qualifica* è definire le metriche e i requisiti minimi di qualità per l'approvazione dei prodotti del progetto.

Titolo Il titolo del documento è "Piano di Qualifica".

Nome del file Il file sarà chiamato `piano_di_qualifica.tex`.

Piano di Progetto

Scopo Lo scopo del *Piano di Progetto* è definire gli obiettivi da raggiungere, stabilendo le tempistiche e chi se ne occupa e tiene traccia del loro progresso, valutando i costi sostenuti rispetto a quelli preventivati.

Titolo Il titolo del documento è "Piano di Progetto".

Nome del file Il file sarà chiamato `piano_di_progetto.tex`.

Analisi dei requisiti

Scopo Lo scopo dell'*Analisi dei requisiti* è raccogliere i risultati dell'attività di analisi dei requisiti. Esso contiene quindi la descrizione dei casi d'uso del prodotto software da sviluppare, ed i requisiti suddivisi per tipologia.

Titolo Il titolo del documento è "Analisi dei requisiti".

Nome del file Il file sarà chiamato `analisi_dei_requisiti.tex`.

Analisi delle tecnologie

Scopo Lo scopo dell'*Analisi delle tecnologie* è valutare le tecnologie disponibili alla realizzazione del progetto. Esso contiene quindi le metriche di valutazione e l'analisi delle tecnologie in base a tali metriche.

Titolo Il titolo del documento è "Analisi delle tecnologie".

Nome del file Il file sarà chiamato `analisi_delle_tecnologie.tex`.

5.2 Verifica e Validazione

Ogni prodotto per essere accettato nel branch `main` deve rispettare i requisiti minimi definiti dal *Piano di Qualifica*.

A tal fine, verrà predisposto un insieme di test per rendere automatica, in toto o in parte, la verifica e la validazione dei prodotti.

5.3 Configurazione

5.3.1 Repository GitHub

Per il versionamento del progetto si è scelto di utilizzare git su piattaforma GitHub.

La repository <https://github.com/iota97/WinningSoftwareSolution> conterrà tutti i file del progetto.

Comandi utili

Alcuni comandi utili sono:

- **Sincronizzazione con repository remota:** `git pull`;
- **Creazione di un nuovo branch:** `git branch <nome_branch>`;
- **Passaggio a un branch specificata:** `git checkout <nome_branch>`;
- **Aggiunta delle modifiche alla stage area:** `git add .`;
- **Creazione del commit con le modifiche:** `git commit`;
- **Push sul remote di un nuovo branch:** `git push --set-upstream origin <nome_branch>`;
- **Push su un branch già esistente:** `git push`.

6 (vecchio)Workflow di lavoro

6.1 Versionamento

Ogni modifica alla repository dovrà seguire i seguenti passaggi:

- modifica del codice su repository locale;
- modifica su un branch collettivo;
- pull request tramite piattaforma web GitHub per merge sul branch `main`;
- review della pull request;
- merge sul branch `main`.

Un branch collettivo è un branch su cui operano uno o più membri del gruppo e ha come scopo l'elaborazione di un prodotto o di un processo di progetto. Il branch si chiamerà come il prodotto o il processo relativo.

Le modifiche su un branch collettivo avverranno tramite merge di un branch personale o con commit diretto. I membri del gruppo che lavorano sul branch concordano una modalità.

È auspicabile che i branch personali non compaiano nella repository remota.

Prima di effettuare un push alla repository remota o un merge è necessario fare uno squash dei commit superflui.

Non appena chi lavora su un branch collettivo ritenga che ci sia materiale pronto per una revisione, apre una pull request su GitHub.

La review dovrà essere fatta dal verificatore. Il verificatore potrà accettare la pull request.

Una pull request può essere accettata solo se tutti i prodotti coinvolti superano i requisiti minimi definiti nel *Piano di Qualifica*.

Quando il verificatore accetta un pull request, dovrà eliminare il branch relativo a quest'ultima.

Non è consentito fare il caricamento diretto di file tramite interfaccia web o desktop.

Non è consentito forzare i push sulla repository remota.

6.2 Sviluppo Software

Per lo sviluppo del codice valgono le seguenti regole:

- chi modifica il SW nella stessa PR modifica la documentazione: diagramma delle classi, di sequenza e lista metodi;
- eseguire i test automatici prima di creare la PR (no regression);
- se si inserisce codice assicurarsi che sia coperto da test;
- non lasciare nulla di implicito, sia per visibilità che per tipi.

7 (vecchio)Tracciamento dei progressi

Per tracciare l'avanzamento del progetto, oltre alla dashboard nel *Piano di Progetto*, verrà utilizzato il sistema di issues e di milestone di GitHub.

Il *Piano di Progetto* definisce quali sono le milestone.

Per ogni milestone verranno create delle issues, in accordo con il *Piano di Progetto*.

Gli issues sono obiettivi intermedi, necessari a raggiungere la milestone corrispondente.

Ogni issue è legato a una milestone.

Un issue può essere diviso in più parti utilizzando una task list.

Ogni sotto-task può essere trasformata in un issue. Quest'ultimo dovrà essere collegato alla stessa milestone dell'issue padre.

7.1 Revisioni di avanzamento interne

Al fine di verificare periodicamente l'effettivo avanzamento delle varie task e meglio coordinare il gruppo è previsto di trovarsi sul canale Discord del gruppo settimanalmente.

Questo incontro avverrà ogni **Mercoledì alle 15.00**.

In caso un componente non possa presentarsi dovrà presentare un resoconto scritto del suo stato di avanzamento ed eventuali criticità riscontrate.

Questo resoconto va consegnato **prima del meeting** in modo che sia possibile discutere con i presenti anche della parte di lavoro svolta dagli assenti. Gli assenti rimarranno aggiornati sullo stato di avanzamento consultando i verbali redatti in occasione di ogni incontro.